UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**DESIGN AND EVALUATION OF
NOVEL GESTURE-CENTRIC INTERACTION MODELS
IN SMART SPACES**

TESIS DOCTORAL

XIAN WANG

Ingeniero en Informática

2016

Departamento de Señales, Sistemas y Radiocomunicaciones

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE MADRID

**DESIGN AND EVALUATION OF
NOVEL GESTURE-CENTRIC INTERACTION MODELS
IN SMART SPACES**

TESIS DOCTORAL

Autor:

XIAN WANG

Ingeniero en Informática

Directora:

ANA MARÍA BERNARDOS BARBOLLA

Doctora Ingeniera de Telecomunicación

2016

Department: Señales, Sistemas y Radiocomunicaciones
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid (UPM)

Ph.D. Thesis: Design and Evaluation of Novel Gesture-Centric Interaction Models
in Smart Spaces

Author: Xian Wang

Advisor: Ana María Bernardos Barbolla

Year: 2016

Committee named by the Rector of Universidad Politécnica de Madrid, on the 14th of
July, 2016.

President:

Member:

Member:

Member:

Secretary:

After the defense of the Ph.D. Thesis on the of 201 .., at
the E.T.S.I. de Telecomunicación, the committee agrees to grant the following qualification:

.....
President **Members**

Secretary

To my family

Acknowledgments

I would like to thank my advisor, Ana María Bernardos, for supporting me over the years, guiding me through the whole master and doctoral study, and for the patience and generous help. Other professors in the Data Processing and Simulation Group (José Ramón Casar, Juan Besada, Javier Portillo and Gonzalo de Miguel) have also been very supportive and gave me valuable advices. Also to Minna Isomursu, Fulvio Corno and José Ramón Casar, who have so generously accepted as the reviewers of the thesis, their constructive comments and suggestions help to improve the thesis.

I would like to thank my many friends and colleagues at UPM, with whom it has been a pleasure to work over the years. Special thanks to Iñigo Marquínez for the Kinect library, the data collection tool with Kinect, the voice-based application and many others, to Luca Bergesio for the home automation system and many other technical help, to Eduardo Metola for the data collection tool with smartphone and the indoor localization module, to Paula Tarrío for the discussions on the gesture recognition approaches, to Iván Corredor for his platform system, to Héctor Cuevas for his user study that provided so many inspirations to the interface design of the system in the thesis, to Enrique García for the idea of gesture input manner with smartphone and the implementation of the gesture recognition approach in Android, to Noelia Rubio for the implementation of the gesture recognition approach in C++, as well as to all the volunteers who participated my many user studies and data collection experiments, without whom it would be impossible to finish the thesis.

I would like to thank Florian Alt for accepting me as an intern at the Media Informatics Group of the University of Munich, and for providing me the opportunity to work on their project and introducing the research in his group. At that time I learned a lot more about interface design. I would also like to thank the generous financial support from China Scholarship Council and the Data Processing and Simulation Group.

I would like to thank my friends in Madrid for providing a welcome distraction from the doctoral study and I would like to thank my family for supporting me, listening to my joys and concerns, giving me suggestions and letting me pursue my dream for so long and so far away from home.

Resumen

Las tecnologías de espacios inteligentes están emergiendo para transformar nuestras vidas. Uno de los mayores retos para acelerar la adopción de estas tecnologías y de sus servicios asociados está en proveer interfaces de usuario intuitivas y naturales. En esta dirección, la interacción basada en gestos parece tener un gran potencial para proporcionar interfaces atractivas y eficientes. Por ello, esta tesis explora y propone nuevos modelos de interacción basada en gestos para espacios inteligentes.

La tesis se desarrolla en torno a una propuesta de un sistema de interacción basado en gestos, pensado para controlar recursos inteligentes, que se analiza siguiendo el modelo de interacción instrumental. El sistema se basa en una gramática gestual la cual es una metáfora de las frases imperativas del lenguaje natural; el objetivo de esta gramática es facilitar la identificación de las entidades interactivas sobre las que se quiere realizar una acción, y las operaciones a realizar sobre ellas, siguiendo una estructura jerárquica sujeto-verbo o sujeto-verbo-objeto. El sistema que interpreta la gramática tiene también en cuenta la ubicación del usuario, para permitir la portabilidad de la interacción gestual y su interpretación contextual. Además de la implementación del sistema, que sigue una arquitectura cliente-servidor, en el trabajo se discuten los problemas de diseño de las interfaces de usuario utilizadas para diseñar soluciones de interacción basadas en tres tecnologías diferentes: reloj inteligente, teléfono y dispositivo Kinect. Estas tecnologías implementan y comparan tres tipos de interfaces, a saber: vestibles (“wearable”), tangibles (en su versión “graspable”) y “no instrumentadas”.

En la tesis se propone asimismo un método de reconocimiento de gestos multitecnología con entrenamiento previo, que es un elemento crítico del sistema de interacción. Este método permite satisfacer los requisitos de los escenarios de aplicación: la configuración de vocabularios de gestos personalizados, un funcionamiento consistente y satisfactorio con una pequeña cantidad de muestras de entrenamiento, la compatibilidad con diferentes tipos de datos de gestos y la facilidad de implementación para diferentes sensores. La evaluación *offline* previa a la implementación del sistema en tiempo real, realizada sobre un *dataset* de 30 gestos procedentes de 11 usuarios, muestra que el método propuesto funciona correctamente tanto con datos de posición como de aceleración, proporcionando en ambos casos una alta tasa de reconocimiento. Las pruebas, que cubren otros aspectos como la expresividad de diferentes tipos de letras o gestos, han permitido también averiguar que tres repeticiones de entrenamiento por gesto son suficientes para converger hacia una tasa de reconocimiento correcto

satisfactoria.

Con el objetivo de proporcionar una solución integral para configurar servicios personalizados en espacios inteligentes, además del reconocimiento de gestos para interacción, se explora en la tesis la identificación biométrica basada en gestos. Ésta puede resultar más integrada en un flujo de interacción basado en gestos que las soluciones que utilizan otros atributos biométricos físicos (e.g. huellas digitales). Las pruebas *offline* realizadas sobre el *dataset* de Kinect reflejan que la identificación a partir de pares de gestos concatenados mejora considerablemente la tasa de identificación correcta, cuando se compara ésta con la implementación que utiliza un único gesto.

En la tesis se documenta también un completo estudio de usuarios, realizado en un entorno realista, que se ha llevado a cabo para evaluar el diseño de las interfaces y el método de reconocimiento. Las pruebas han sido diseñadas siguiendo una estrategia *within-group* (todos los participantes han ejecutado las mismas tareas) y en ellas han participado doce individuos. El estudio se ha diseñado en cuatro fases: entrenamiento (para hacer que el sistema funcionase correctamente para cada usuario), realización de pruebas de precisión, ejecución de trayectorias de interacción (en las que los participantes han tenido que ejecutar seis acciones de la vida diaria con gestos) y utilización de reconocimiento de voz para control del entorno (para comparación de ambas técnicas). Para las tres tecnologías, se han recogido tanto métricas de errores y tiempos de ejecución como evaluaciones subjetivas del vocabulario de gestos, gramática, esfuerzo físico, interfaz de usuario, entrenamiento y preferencias. A partir de los datos obtenidos se puede observar que el reconocimiento de gestos en tiempo real tiene una precisión moderada, más reducida que la conseguida en las pruebas *offline*. Es efectivo en términos de tiempo, ya que no introduce un retardo significativo en la duración total de la interacción. El tiempo de realización de tareas con gestos es comparable con el necesario para ejecutar la interacción con voz, aunque los participantes han cometido más errores con gestos que con voz para una misma tarea. La mayor parte de los participantes han señalado que la gramática propuesta es fácil de aprender y podría ser efectiva para controlar los recursos inteligentes.

Considerando el análisis de los resultados del estudio de usuarios, se han extraído también una serie de recomendaciones de diseño que están relacionadas con los modos de realizar los gestos, el reconocimiento, la gramática, la interfaz de usuario o la mejora de la eficiencia del método, con el objetivo de facilitar el diseño de otros sistemas de interacción basados en gestos. La evaluación obtenida del estudio de usuarios puede asimismo ser utilizada para comparar modelos de interacción posteriores.

Abstract

Technologies of smart spaces are emerging to transform our lives. One of the main challenges to accelerate the adoption of these technologies and of their associated services lies in providing intuitive and natural user interfaces. In this direction, the interaction based on gestures seems to have a great potential to provide attractive and efficient interfaces. Thus, this thesis explores and proposes new models of interaction based on gestures for smart spaces.

The thesis develops around a proposal of an interaction system based on gestures, which is conceived to control smart resources, and is analyzed with the instrumental interaction model. The system is based on a gestural grammar that is a metaphor of imperative phases of natural languages; the objective of this grammar is to facilitate the identification of interactive entities on which users want to realize an action, and the operations to realize on them, following one hierarchical structure subject-verb or subject-verb-object. The system which interprets the grammar also considers location of users to allow portability of the gestural interaction and its contextual interpretation. In addition to the implementation of the system, which has a client-server architecture, this work discusses problems of design of user interfaces utilized for designing solutions of interaction based on three different technologies: the smartwatch, the smartphone and the Kinect device. These technologies implement and compare three types of interfaces, namely: wearable, tangible and natural.

In the thesis a gesture recognition approach, which is a critical element of the interaction system, is proposed with previous training and based on multiple enabling technologies. This approach allows meeting the requirements of application scenarios: configuration of personalized gesture vocabulary, performance that is consistent and satisfactory with a small amount of training samples, compatibility with different types of gesture data, and facility of implementation for different sensors. Before the implementation of real time system, an offline evaluation carried out on a dataset of 30 gestures collected from 11 users, shows that the proposed approach works with both position data and acceleration data, providing high recognition rate in both cases. The tests, which cover other aspects such as expressivity of different types of letters or directional movements, have allowed ascertaining that three repetitions of gesture training are sufficient in order to converge to a satisfactory rate of recognition.

To provide an integral solution to configure personalized services in smart spaces, aside

from the gesture recognition for interaction, biometric identification based on gestures is explored in the thesis. The identification can be more integrated in an interaction workflow based on gestures than solutions that utilize other physical biometric attributes (e.g. digital fingerprints). The offline tests carried out on the dataset of Kinect reveal that the rate of correct identification with concatenated gesture pairs improves considerably, compared with an implementation that utilizes single gestures.

A complete user study, which is carried out in a real environment to evaluate the design of the interfaces and the recognition approach, is documented in the thesis as well. The tests have been designed with a within-group strategy (all participants perform same tasks) and twelve individuals have participated. The study has been designed in four phases: training to make the system work correctly for each user, a gesture recognition test, a user journey test in which the participants have six daily life tasks with gestures, and utilization of speech recognition to control the environment for a comparison with gesture-based interaction. For the three technologies, the user study collects metrics of errors and time performances, as well as subjective evaluations of a gesture vocabulary, the grammar, physical effort, user interface, training and preferences. From the obtained data it can be observed that the gesture recognition in real environment has a moderate precision, much lower than that obtained in the offline tests. It is effective in terms of time complexity, since it does not introduce significant delay in the total duration of the interaction. The task time with gestures is comparable with the one needed for interaction with voice, although the participants have committed more errors with gestures than with voice for the same task. The majority of the participants have indicated that the proposed grammar is easy to learn and could be effective to control the smart resources.

Based on the analysis of the results of the user study, a series of design recommendations are extracted as well, which are related with gesture performance procedure, the gesture recognition, the grammar, user interface, and enhancement of gesture input efficiency, aiming at facilitating the design of other interaction systems based on gestures. The evaluation obtained from the user study could be utilized likewise as a benchmark for the following design iterations.

Contents

Contents	xv
List of Figures	xix
List of Tables	xxi
1 Introduction	1
1.1 Background	1
1.1.1 Human factors as a challenge for smart spaces technology	2
1.1.2 Gestures as interaction enablers	2
1.2 Research problems	4
1.3 Contributions	6
1.4 Document structure	15
2 State of the art	19
2.1 Gesture-based interaction in smart environments	19
2.1.1 Application scenarios	20
2.1.2 Sensors	21
2.1.3 Gesture set design	24
2.1.4 Gesture taxonomy	27
2.1.5 Device selection	27
2.1.6 Usability of gesture-based interaction systems	28
2.1.7 Social acceptability	28
2.2 Gesture recognition techniques	29
2.2.1 Hand tracking	29
2.2.2 Gesture spotting	30
2.2.3 Feature extraction	30
2.2.4 Classification	31
2.2.5 Efficient training	33
2.2.6 Criteria to choose a recognition approach	33
2.3 Gesture-based identification approaches	33
2.4 Conclusion	35

3	GENIE: A Gesture-ENabled Interaction method for smart Environments	39
3.1	The design space	39
3.2	A gesture-based grammar to command the smart environment	40
3.3	Multi-device GENIE architecture and implementation details	42
3.3.1	Architecture	42
3.3.2	Interaction workflow	45
3.3.3	The end-user interface	46
3.4	Conclusion	54
4	A strategy for gesture-based recognition and identification	59
4.1	Challenges and design objectives	59
4.2	A classification algorithm to enable gesture recognition and identification . .	61
4.2.1	Similarity measurement based on Dynamic Time Warping algorithm .	61
4.2.2	Recognition and identification approach with the similarity measuring engine	62
4.3	Off-line evaluation of the gesture recognition approach	65
4.3.1	Testing scenario	65
4.3.2	Results	71
4.4	Off-line evaluation of the gesture-based identification approach	82
4.4.1	Testing scenario	82
4.4.2	Results	82
4.5	Conclusion	85
5	User study	89
5.1	Research questions	89
5.2	The experimental design	90
5.2.1	Design	90
5.2.2	Participants and apparatus	90
5.2.3	Procedure	91
5.3	Results	96
5.3.1	The collected data and the analysis procedure	96
5.3.2	Evaluation of training and gesture recognition performance	98
5.3.3	Evaluation of the grammar use for smart space control	110
5.3.4	Analysis of technology preferences: technology comparison, usability, user experience	118
5.4	Design recommendations	125
5.5	Conclusion	129
6	Conclusion	135
6.1	Discussion	137
6.2	Limitations	138
6.3	Future work	138

Appendix A	Offline evaluation of the gesture recognition approach	143
A.1	Confusion matrices	143
A.2	Handwriting of letters	144
Appendix B	Brief introduction to the statistical methods used in data analysis	147
B.1	Repeated measures ANOVA	147
B.2	Paired-samples t-test	148
B.3	Friedman test	148
Appendix C	Questionnaire of the user study	151
C.1	User profile	151
C.2	Training section	152
C.2.1	Kinect	152
C.2.2	Phone	153
C.2.3	Watch	154
C.2.4	Summary	155
C.3	User journey section	156
C.3.1	Gesture preference	156
C.3.2	Kinect	157
C.3.3	Phone	161
C.3.4	Watch	165
C.3.5	Summary	169
C.4	Voice-based services section	175
C.4.1	System Usability Scale	175
C.4.2	Summary	176
Bibliography		177

List of Figures

1.1	Outline of the research	17
2.1	The pipeline of gesture recognition	29
3.1	The interface design described with the instrumental interaction model	40
3.2	The architecture of the interaction system	43
3.3	The interaction workflow chart	45
3.4	Illustration of the gesture input procedure of the three technologies. Each column shows the gesture input manner and snapshot of users doing gestures. Columns from left to right: the Kinect, the smartphone and the smartwatch	48
3.5	Illustration of the area of the sweet spot	49
3.6	The time series of performing gestures with the smartwatch	51
3.7	The virtual button to redo one training repetition	53
3.8	Finite state machine where the semaphore shows the status of gesture sequence composition	53
3.9	Illustration of the interface of the Kinect-based application	55
3.10	Illustration of the interface of the Phone-based application	56
3.11	Illustration of the interface of the Watch-based application	57
4.1	Architecture of the gesture recognition module	63
4.2	Illustration of directional/geometric gestures	66
4.3	Captured signal of the gesture ‘w’	68
4.4	Illustration of the Type I local continuity constraint	71
4.5	Frequency-normalized histograms of intra- and inter- class DTW distances	72
4.6	Recognition accuracy of users on the two platforms	74
4.7	Mean and standard deviation of accuracy of the gesture groups on the two platforms	75
4.8	The recognition accuracy with respect to the three gesture groups (for. = forward, back. = backward, c. c. = clockwise circle, cc. c. = counter-clockwise circle, s. l. = spiral to the left, s. r. = spiral to the right)	77
4.9	The evolution of the mean and the standard deviation of recognition accuracy with the number of training samples (directional=directional movements, small=small letters, capital=capital letters)	78

4.10	The illustration of gesture clusters. Different shapes represent different gestures. Shaded nodes are selected as templates.	79
4.11	The evolution of the mean and the standard deviation of recognition accuracy with the number of existing users	81
5.1	Views of the three zones in which the user study are carried out: Hall area, Meeting room and Relax room	93
5.2	Average recognition rate with the three technologies. Error bars show one standard deviation	99
5.3	Recognition rates of different gesture types with the three technologies in two postures. Error bars show one standard deviation	100
5.4	The distribution of gesture performance errors with the Kinect. (UGI - unintended gesture input, IGB - imprecise gesture boundaries, FSG - fail to start gesture, IHT - instable hand tracking, CP - confusion by participant)	103
5.5	Gesture performance error comparison among the three technologies in the training and recognition test (UGI - unintended gesture input, FSG - fail to start gesture, IHT - instable hand tracking)	104
5.6	The mean training time with the three technologies. Capped error bars represent a 95% confidence interval	105
5.7	Average duration (ms) of each gesture with the Kinect, for all users. The error bars are ± 1 average absolute deviation	106
5.8	Average duration (ms) of all gestures with the Kinect, the Phone and the Watch for each user. The error bars are ± 1 average absolute deviation	108
5.9	Time analysis on the phases in the gesture performance-recognition workflow. K = Kinect, P = Phone, W = Watch, Po1 = standing, Po2 = seated, Phase 1-4 are gesture performance, preprocessing, recognition, wireless transmission, respectively	109
5.10	The average time of tasks without errors	112
5.11	Agreement for each task in the questionnaire, sorted in descending order	115
5.12	Average ratings of gesture acceptability by location and audiences. Error bars show one standard deviation	116
5.13	Results of the six Likert scale questions (Scale 1 - perspective to gestures, Scale 2 - learnability of the grammar, Scale 3 - effectiveness of the grammar, Scale 4 - performance of multi-gesture, Scale 5 - the location-aware feature, Scale 6 - the confirmation step)	118
5.14	Comparison of average time of task without errors	119
5.15	Box plots of the results of the Likert scale ratings on the training section	120
5.16	The SUS scores of the four technologies	123
5.17	Results of the UEQ, with +3 the most positive and -3 the most negative	124
5.18	Technology preference questions in Likert scale format (G = Gesture, V = Voice, K = Kinect, P = Phone, W = Watch)	125
A.1	Survey of handwriting of letters	146

List of Tables

3.1	Input and feedback	54
4.1	A primitive vocabulary for experiments	66
4.2	The comparison of time cost on the Kinect and the Phone (ms)	75
4.3	Identification accuracy with single gestures	83
4.4	Identification with concatenated gestures, system-guided	84
4.5	Identification with concatenated gestures, user-centric	85
5.1	The user test sequence	91
5.2	The user journey tasks (C.C. = clockwise circle)	95
5.3	Tasks with the voice-based application	95
5.5	Confusion matrix with the Kinect (C. C. = clockwise circle, recog. = recognized)	101
5.6	Confusion matrix with the Phone	101
5.7	Confusion matrix with the Watch	101
5.8	Gesture duration statistics	106
5.9	Time and proportions of the phases in gesture input workflow	107
5.10	Task errors (“TI” means “task instance”)	111
5.11	TIs with grammar and vocabulary errors distributed by tasks and technologies .	111
5.12	p values for pair-wise comparison of locations. p values have been adjusted using Bonferroni’s correction and $p < 0.05$ is significant (shown with figures in bold) .	115
5.13	p values for pair-wise comparison of audiences. p values have been adjusted using Bonferroni’s correction and $p < 0.05$ is significant (shown with figures in bold) .	117
A.1	Confusion matrix about detection of directional movements based on the Kinect	143
A.2	Confusion matrix about detection of small letters based on the Kinect	144
A.3	Confusion matrix about detection of capital letters based on the Kinect	144
A.4	Confusion matrix about detection of directional movements based on the Phone	145
A.5	Confusion matrix about detection of small letters based on the Phone	145
A.6	Confusion matrix about detection of capital letters based on the Phone	146

Chapter 1

Introduction

Smart space technologies promise to improve our daily life and provide more and more opportunities for people to interact with the physical and digital world in an enriched way. Many of the challenges that are currently open in this area of knowledge and economic activity are related to how to improve the human-computer interaction workflow. In brief, this research work aims at exploring how to design a novel interaction method to facilitate operating in a smart space, in particular using gesture-based intermediation. This Chapter discusses the motivation, the research scope, and the contributions of the work.

1.1 Background

A smart space is defined as “a highly integrated computing and sensory environment that effectively reasons about the physical and user context to transparently act on human desires” [66]. The concept brings together four components in ubiquitous computing [102]: devices that support intuitive user interaction and user mobility, wireless communications, networks of sensors and actuators, and reasoning mechanisms to implement an environment that understands and reacts to human desires. The concept of smart space is applicable in very different settings, ranging from small scale scenarios such as smart offices to metro-wide scenarios like smart cities [68]. This work is focused on small to medium scale environments.

The reviewed literature emphasizes the application of smart space’s technologies on for example, enhancing energy efficiency [77], improving healthcare [2] or simply improving our daily routines [89].

The smart spaces, equipped with smart objects with the capabilities to sense, actuate and be commanded, can provide their inhabitants with more and more opportunities of control and automation of lighting and appliances, heating and ventilation, entertainment, etc. As described in [76], the goal of a smart home is that the ambient environment can be controlled and adapted to provide an improved living experience for the inhabitants by monitoring their activities along with how they interact with domestic devices.

1.1.1 Human factors as a challenge for smart spaces technology

The deployment of smart space technologies is impacting on personal lives and societies. However, its spreading also faces many challenges and open research questions. Apart from the technical issues mentioned in [35], like system architecture design, service enhancement, coexistence with the ubiquitous Web and markets of devices and sensors, Suresh et al. [89] identify the barriers in cost, investment, privacy and legal issues that hinder the widespread adoption of smart space/home technology. The concept of human factor is defined by the International Ergonomics Association (IEA) as “the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance”. Some publications investigate the role that the human factors play in the adoption of smart home applications. For example, Costanza et al. [23] and Paetz et al. [77] reveal the socio-technical challenges when integrating new technologies into everyday routines. Through the studies, the authors found very different behaviors among the participants, thus they argued that it is necessary to study the configuration of the trade-offs between convenience and utility so that the users will accept new technologies in their daily lives. There are other works stressing the importance of human factors. The work [89] regards the learnability of the technology might be an obstacle to its adoption. Based on the revision of the user studies on healthy smart homes, Kim et al. [45] listed several research issues, including that i) the systems should be developed based on the needs of the end users instead of on technological capabilities, that ii) it is essential that intuitive user interfaces are designed, that iii) the end users’ feedback about how the system performances in actual use is essential, that iv) user experience should be considered for a successful technology interaction and adoption of health smart systems, and that v) the consideration of the usability of the system is essential to increase the acceptability of the system. Cenedese et al. [18] also emphasize the importance to consider the man-in-the-loop issue and to pose the centrality of human behavior in terms of context awareness and interface.

1.1.2 Gestures as interaction enablers

We see from the above that the learnability, usability and user experience are crucial to the adoption of the smart home technology. An intuitive, natural and efficient user interface would help address the challenges of technology adoption. Facilitating natural human-computer interaction (HCI), i.e. interacting with computers as natural as an interaction between humans, will benefit the users in a smart space where the technology becomes so embedded into our daily lives and the applications require more and more interactions for efficient usage.

Gestures have long been considered a mode upon which more natural, creative and intuitive interaction models can potentially be built [80]. Gestures can be defined as “expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head,

face or body with the intent of: 1) conveying meaningful information or 2) interacting with the environment” [70]. We use gestures in our daily lives and according to Cadoz [22], the three functional roles of human gestures are: semiotic - to communicate meaningful information (65% of human communication consists of non-verbal gesture based communication [33]), ergotic - to manipulate the environment, and epistemic - to discover the environment through tactile experience. Kendon [44] defined five kinds of gestures:

- Gesticulation. Spontaneous movements of the hands and arms that accompany speech
- Language-like gestures. Gesticulation that is integrated into a spoken utterance, replacing a particular spoken word or phrase
- Pantomimes. Gestures that depict objects or actions, with or without accompanying speech
- Emblems. Familiar gestures such as “V for victory”, “thumb up”, and assorted rude gestures (these are often culturally specific)
- Sign language. Linguistic systems, such as American Sign Language, which are well defined

In the order from gesticulation to sign language as in the above list, the association with speech declines, language properties increase, spontaneity decreases and social regulation increases. Emblematic gestures and gestural languages, which are less spontaneous and natural but carry clearer semantic meanings, may be more appropriate for command-and-control interaction, as argued in [92]. Karam [42] reported that in the literature, hand has been widely used in comparison to other parts of the body for gesturing due to its natural form of medium for communication. The applications of hand gesture-based interaction are summarized in [80], ranging from visualization, robotics and virtual reality to ubiquitous computing. In other reviewed works, the gesture-based interaction has also been adopted in operating rooms, entertainment, smart driving, etc. Detailed literature review on the application scenarios is presented in Chapter 2.1.1.

There has been a strong commercial interest in enabling technologies of gesture-based interaction. Leap Motion [61], first released a few years ago, is a small USB peripheral device that allows touchless hand and finger input and MYO armband [73], worn on the wrist, senses electrical activity in the forearm along with inertial and magnetic measurements to recognize gestures. Most recently, Google showcased a new chip from its Project Soli [78], which is a sensing technology that detects touchless gesture interaction with miniature radar. A concept smartwatch with the chip had the capability to track small movements like waving of fingers by radar and worked completely based on gestures, enabling a new way of touchless interaction. The chip is small enough to be embedded in wearables, phones, computers, cars and IoT (Internet of Things) devices. In addition, Microsoft revealed a pre-touch sensing technology [36] for mobile interaction with hover gestures. The sensors on the sides of a device

CHAPTER 1. INTRODUCTION

detect whether the hand is approaching to the screen, how the phone is held by grip, etc. The floating touch features anticipate and sense interactions, and provide contextual features. For example, an app pops up control options automatically when the hand approaches the screen when it is playing a video. In the virtual reality area, the company of eyeSight Mobile Technology showed off gesture controls for a smartphone-based virtual reality headset [30], aiming at facilitating truly immersive virtual experience.

Depending on the application scenario, the gesture-based interfaces may have several advantages compared with the traditional ones and can be used by those with loss of vision, motor skills and mobility. A panel in a wall gathering e.g. the commands to manage a smart home, requires the users to approach to the panel to use, which is not convenient to those with difficulty of mobility. Remote control might have small buttons difficult to push and read. Touch screen interfaces can have similar limitations as those of the remote control. Speech recognition might face several difficulties in home automation environment, as discussed in [88]. The noise from the constant talking between the inhabitants or the background noises will force the users to speak very loudly and the control with vocal commands may interrupt a normal conversation in some social scenarios.

In addition, the development of sensor and tracking technology, the proliferation of computing devices (smartphones, wearables, etc.) that are available to the mass market and computationally powerful, the popularization of wireless communication infrastructures as well as the research on algorithms of pattern recognition and machine learning increase the possibilities of spreading of gesture-based interaction for smart environment applications.

1.2 Research problems

Although there are many types of gestures according to their definition, this work utilizes 3D in-air motion hand gestures for interaction. This work deals with “gesture-centric interaction”, which means gesture is the central and main input manner in the interaction. Another similar term is “gesture-based interaction”, which means that interaction is based on gestures and that gesture is the foundation for the interaction. Both of the terms “gesture-centric interaction” and “gesture-based interaction” indicate that the gesture is the primary input manner and there might be other auxiliary input modes. These two terms have the same meaning in this article and they are different from “gesture-only interaction”. The proposed gesture-based interaction models (model here means a particular design of a system) have the 3D in-air motion hand gestures as primary input, complemented by other input formats such as touchscreen gestures, the keyboard, the mouse, etc. Gesture-based user interface refers to this kind of input manner in the rest of the thesis.

Suppose that we have a smart home infrastructure comprising a wireless network, some smart objects and sensors, and a coordinator, which delivers diverse services (e.g. home automation, healthcare, etc.). In addition, the enabling technologies of gesture-based interaction, such as Kinects, smartphones or smartwatches, are widely available and have the capability to track the gesture movements. And in the smart home environment there are

users who in their daily lives communicate a lot with gestures. This work then focuses on design, implementation and evaluation of gesture-based interaction models that bridges the barrier between users and computers based on the enabling technologies, and facilitates the inhabitants to make better and more comfortable use of the technology around them.

Aiming at a real application scenario, i.e. managing smart resources in smart spaces with gestures, the main research problems in this work include:

- Deployment of an efficient interaction method via gestures to work over multiple smart resources available in the smart space. In order to select a target device to control, some works and commercial offers rely on a single-gesture schema [79], where the target device of a gestural command (one gesture) is implicitly indicated. Other alternatives propose the use of RFID tags [8] and visual marks [56], natural pointing [17] [28] or speech [15]. Mapping every gesture to each action of each device can lead to a large gesture set that is difficult to remember even if there are only a few target devices, and make some gestures less intuitive. Tags and marks need extra deployment and readers. The strategy with pointing might suffer from recalibration if the smart objects are relocated. In addition, the indication of devices by pointing and body orientation can be vague if the devices are placed too crowded. Speech is not suitable in some scenarios [88]. In this work, to overcome the above limitations, a gestural grammar is proposed in Chapter 3 to effectively organize target device selection and command specification.
- Design of a gesture recognition strategy that is sufficiently accurate, works in real-time, requires little training and is implementable for different devices. The recognition of gestures can be challenging. Unlike speech, there is no standardized vocabulary of gestures. The performance of the same gesture varies among different users and among the repetitions of the same user. The essentially same gesture performance can result in different gesture data samples, if the reference coordinate system is based on the sensors. Like speech signal, the gesture data is a continuous stream and segmentation would be necessary. The challenges and the design of gesture recognition approaches will be discussed in detail in Chapter 4.
- Definition of a user interface that satisfactorily guides the users when they input gestures and interact with the smart environment. Automatic gesture segmentation is considered difficult with the current technology [92] [70]. This work adopts the strategy of manually marking the start and end of the gesture performance by users via careful design of user interface. On the other hand, the enabling technologies of gesture-based interaction, e.g. the Kinect, the smartphone and the smartwatch, provide heterogeneous resources for input and output modes. The user interface design for gesture-based interaction will be explored in Chapter 3.
- Evaluation on the interaction method and the performance of the implemented interaction models. The proposed interaction method and interaction models need rigorous

CHAPTER 1. INTRODUCTION

evaluation to assess the design, to elicit design recommendations on gesture-based interaction system and to shed lights for the next design iterations.

1.3 Contributions

Several gesture-based interaction systems in the literature and this work are summarized in Table 1.1.

Table 1.1: Comparison of previous gesture-based interaction systems and this work

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
Rahman et al. [79] (2009)	Smart home	IR cam- era + glove with IR emitter	Letters and ge- ometric symbols	“Push-to- gesture”, single gesture command	Gesture drawing shown on the screen	37.25% in the first phase	Usability assess- ment	-
Kühnel et al. [58] (2011)	Smart home	Smart- phone	Elicited gesture set (di- rectional motions and tilt)	GUI to se- lect device + single- gesture command	Touch- based but- ton for “push-to- gesture”	90.76% (offline)	Gesture recog- nition test	On rec- ognizer and in- teraction mode
Kühnel et al. [57] (2010)	Smart home	Voice (wizard), smart- phone and mix- ture	Same as Kühnel et al. [58] (2011)	Same as Kühnel et al. [58] (2011)	Vibration feedback, GUI interface	51.33% (online), with 5 training repeti- tions	Recog- nition rate, task success, modality choice	Ques- tionnaire selection for mul- timodal interfaces
Do et al. [28] (2006)	Smart home	Color cameras	6 one- and 4 two-di- mensional motions	Pointing to device + single gesture	Vocal and visual feedback	96.5% on- line test, 97.1% of- fline test	20 tests per func- tion, success rate 96.5%	-
Continued on next page								

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
Kollee et al. [50] (2014)	Smart office spaces	Head-mounted devices with depth camera	Grasp gestures	Select display or items by grasping, open hand to release	Visual feedback on Google Glass	-	Modality usability, pick and drop task time, error count, SUS	-
Kela et al. [43] (2006)	Smart Design Studio	SoapBox equipped with accelerometer	Up, down, left and right, tilting and rotation, push and pull	“Push-to-gesture”; straight motions, tilting to move 3d model, push and pull for zooming	Visual feedback	Offline test - 81.2% (1 training) and 98.9% (12 trainings)	Usefulness of gesture modality compared to other modalities	Gesture design, recognition, interaction modality

Continued on next page

Continued on next page

Table 1.1 – continued from previous page

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
Bigdelou et al. [13] (2012)	Comput- erized medical system in operating room	Wireless inertial sensors placed on the arms	8 ges- tures in the user study	Voice or handheld switch to activate recog- nition; categorical gesture to select tools and spatio- temporal gesture for setting parameter	Visual feedback (medical image viewer)	Processing speed, sensitiv- ity to sensor place- ment, training dataset size (data from one person)	Usability test	-
Bailly et al. [6] (2011)	Interactive televi- sions	Wiimote	Pitch, yaw, roll and pad buttons	Pad but- tons and press-tilt- release for rotation to select menus	Eyes-free in expert mode and visual feedback in novice mode	90%+ for selection among 7 items with roll gesture	Selection accuracy, time, subjective prefer- ences	Gestural set for remote control, 5 novel in- teraction manners
Continued on next page								

Table 1.1 – continued from previous page

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
Dingler et al. [27] (2015)	Display interac- tion	Touch screen, Leap- Motion sensor, Kinect	Elicited gesture set for multiple, physical inter- action zones	-	-	-	Pointing task com- pletion dif- time, ges- ture set elicitation	Map functions to dif- ferent areas; use proxemics as input modality
Bailly et al. [7] (2012)	Interaction for mobile devices, accessi- bility and demon- strative scenarios	Depth camera	Triangle, radial and finger- count; combina- tions of the three	Pinch as registra- tion pose for 3d ra- dius menus and finger- count gestures	Shoeborne indica- tors, audio channel and visual annexing	Offline test, all are higher than 94%	Social accept- ability, physical, mental demand, user pref- erence, recogni- tion	-
Continued on next page								

Table 1.1 – continued from previous page

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
Walter et al. [95] (2013)	Interactive public displays	Depth sensor	Teapot gesture	Initial gesture as registra- tion, three strategies to reveal initial gesture	Visual	-	Mid-air gesture revelation strategies	Strategy of rev- elation gesture, initial gesture enabling interac- tions
Chat- topad- hyay et al. [19] (2014)	Interaction with large displays	Kinect	Simple di- rectional move- ments	Cross ROI to trigger menu; directional motions to select menus by crossing	Visual (color chang- ing and zooming of menus)	-	Effect of trigger lo- cation on selection; compar- ison of TCM and linear menu	-
Continued on next page								

Table 1.1 – continued from previous page

Ref.	Appli- cation scenario	Enabling technolo- gies	Gesture set	Interaction schema	Interface	Gesture recogni- tion	Evaluation	Design recom- menda- tions
This work	Smart spaces	Kinect, smart- phone and smart- watch	Letters and ge- ometric motions	Gestural grammar; gesture segmen- tation: threshold- based, push-to- gesture, reserved gesture	Visual, haptic	Offline: 98.64% (Kinect), 93.63% (smart- phone); online: 76.5% (Kinect), 76% (smart- phone), 71.1% (smart- watch)	Recogni- tion rate, time anal- ysis, task errors and time, UI eval- uation, social accept- ability, physical exertion, prefer- ence on tech- nologies, usability and UX	Gesture input pro- cedure, gesture recog- nition, gram- mar, user interface

The contributions of the thesis are listed below:

1. **Proposal of a novel gestural grammar for interaction with smart resources.**

The smart space is filled with different resources, each of which provides several commonly used operations. A gestural grammar, which is a metaphor of the imperative sentence in the natural language (i.e. “object A do action B” or “object A do action B over object C”), is proposed and applied to the smart space scenario to organize the resource entities and operations into a hierarchical structure, facilitating the reusability of gestures representing operations. The gesture vocabulary under this gestural grammar is more intuitive, compact and efficient than the one with single-gesture schema, which implicitly associates each gesture with each command of each device, such as in [79].

2. **Exploration of gesture recognition and gesture-based identification approaches for an interaction system.**

- a) **Design, implementation and validation of a multi-device gesture recognition approach.** Targeting at a real application scenario, the gesture recognition approach is designed to support a personalized gesture vocabulary, light in gesture training, compatible with different types of sensor input data (e.g. coming from inertial sensors or video cameras) in addition to being accurate and real-time. Most of the works about gesture recognition approaches in the review literature are focused on gesture data from optical [107, 31, 96] or inertial sensor [52, 18, 37], or the fusion of these two [21, 65, 20]. This work, however, explores a unified solution to the gesture recognition problem based on both types of sensors, widening the base of compatible gesture input devices in terms of sensing devices. The collected gesture dataset can be reused to test other recognition algorithms. In addition, the reviewed literature only tests the proposed recognition approaches offline (cross validation on common or local collected dataset), except [43, 79, 28]. [43] just discussed some problems in online (real application scenario) test while [79, 28] only reported the accuracy of their online test. This work will discuss in detail recognition accuracy, errors in gesture input procedure, and time analysis of gesture input workflow in a real application scenario.
- b) **A strategy to optimize user-dependant training mechanisms.** Based on an offline evaluation of the proposed gesture recognition approach, a gesture recognition strategy that first works in user-independent mode and then improves the performance with more availability of training samples, is proposed for Kinect-based applications to essentially eliminate the need of training by new users. Most of the reviewed recognition approaches [21, 54, 110] are tested with more than one training samples. Some works [96, 105, 52] are dedicated to approaches with only one training sample. Others ones [20, 3] report impressive accuracy with user-independent approaches, which do not require training from new users. The

proposal of training optimization in this work combines user-independent and user-dependent manners while removing the need of training.

- c) **Exploration of the feasibility of using concatenated 3D in-air gestures as a biometric input in smart spaces.** Gestures as behavioral attributes for identification are less invasive than the biometric physical ones. Previous works identify users with in-air single-gesture of digits [111], implicit pick-up or shaking [32], and signatures [69]. In this work, user identification based on the Kinect with gesture pairs is proposed, showing in evaluation a significant improvement on accuracy.
3. **Proposal of a system architecture and interface design for gesture-based interaction models (systems).**
- a) **Design and implementation of an architecture enabling the deployment of gesture-aware services in smart spaces.** A client-server architecture allows to easily scale the smart resources integrated into the interaction system, apart from facilitating the incorporation of other devices suitable to be used in gesture recognition (e.g. Leap Motion, Ring ZERO, etc.). These devices can be easily incorporated by implementing the interface between the client and the server and a user-end interface. Along with the ability to personalize the gesture set and the mapping between commands and gestures, the design has the potential to easily port to other smart spaces and facilitate the deployment of services that use the proposed grammar or single gestures. The architecture counts on a grammar compiler that facilitates this customization. It is specially designed for an application scenario of multiuser with heterogeneous gesture-input devices, typically found in smart spaces. Few reviewed works address this concern (a similar work [43] has a system providing multimodal interaction for design studio, including gestures).
 - b) **Interface design with three enabling technologies.** The issues to incorporate three enabling technologies to a gesture-based interaction system are explored. The technologies include the Kinect, the smartphone and the smartwatch, covering natural, graspable and wearable user interfaces, and explicit (position data) and implicit (inertial data) tracking technologies. Most of the previous works employ a single enabling technology, such as the smartphone [58], inertial sensors [13], color cameras [28], etc. There are also multimodal interaction systems in smart design studio [43] and smart office [50] environment including gestural interaction on a single technology. The display interaction in [27] combines technologies with different sensing ranges to cover multiple interaction zones. This work, however, is an attempt to develop one system enabling multiple independent gestural modalities for interaction in smart spaces. Not only should the gesture recognition approach adapt to these three technologies, but also the user interface related issues like gesture input procedure, control components and

feedback for real application scenario. The user interface is designed given the constraints of the resources of each platform; the design decisions to address the constraints will be discussed in detail in Chapter 3.

4. Evaluation and design recommendations.

- a) **Evaluation on the designed interaction models in a quasi-real environment.** Some previous works make only objective or only subjective evaluations on their developed systems [79, 58, 28]. A user study in this work, however, has been carried out to exhaustively evaluate the interaction models with both objective and subjective metrics including the performance in terms of effectiveness and efficiency, and subjective ratings on the grammar, user interface, social acceptability, physical exertion, training, preference on technologies, usability, user experience, etc. Metrics like gesture input errors, time and subject rating of training, time analysis on workflow of gesture input, or comparison between enabling technologies are rarely measured in the literature. The results can serve as a benchmark for the following design iterations.
- b) **Definition of a set of design recommendations for gesture-based interaction systems.** Reflections and suggestions on the gesture set design, gesture recognition and interaction design are proposed in previous works like [58, 43]. In this work, design recommendations regarding the gesture input procedure, the gesture recognition, the grammar, user interface, as well as directions of improvement are discussed as conclusions of the user study.

1.4 Document structure

The structure of the thesis is as follows. After reviewing the literature on the topics of gesture-based interaction systems, techniques of gesture recognition and gesture-based identification in Chapter 2, the design, architecture and implementation details of the gesture-based interaction system in the work are described in Chapter 3. As an important enabling technique, the strategy for the gesture recognition along with the feasibility analysis of using in-air gestures as biometric input for identification is designed and evaluated in Chapter 4, given the requirements of the interaction system. Within the research process, it was found essential to carry out a user study to evaluate the interaction system in a quasi-real environment, since on one hand it is an interface designed to be used in a daily basis and on the other hand, the gesture-based interface is relatively new and the human factors still need to be carefully examined. Key features like acceptability, learnability, usability, user experience and objective performance are studied in the Chapter 5 and some design recommendations are summarized finally. The Chapter 6 concludes the work and discusses future work directions.

Figure 1.1 shows an outline as well as process of the research. The numbers in italics between parentheses are sections of the chapter where the corresponding topics will be discussed. Although the investigation on gesture recognition and user identification is placed

CHAPTER 1. INTRODUCTION

in Chapter 4, this part was first carried out in the doctoral study. Taking consideration of requirements of the gesture-based interaction system, Dynamic Time Warping (DTW)-based approaches were used to recognize gestures and identify users. Off-line evaluation on the gesture recognition approach was conducted using data from the Kinect and the smartphone, which explicitly track hand with position from optical sensor and implicitly with acceleration by inertial sensing, respectively. So was off-line evaluation on gesture-based user identification, but only with data from the Kinect since user identification is normally not necessary for personal devices like the smartphone or the smartwatch.

Results of off-line evaluation on gesture recognition not only show feasibility to implement the recognition approach in a real interaction system, but also provide indications on integration of possible enabling technologies, choice of gesture vocabularies, necessary number of training samples, etc. After an analysis of design space, GENIE (Gesture-ENabled Interaction method for smart Environment), a gesture-based interaction system was designed incorporating the idea of a gestural grammar that aims at organizing resource entities and operations of smart space in a hierarchical structure. The system was implemented on a client-server architecture and a specific user interface for each of the three enabling technologies including the Kinect, the smartphone and the smartwatch.

To evaluate the acceptability of the proposed gestural grammar, the performance of the recognition approach and the interaction experience in a quasi-real environment, a user study was carried out, delivering a set of metrics to investigate the research questions. The analysis of the study's results leads to several design recommendations for gesture-based interaction systems, which may guide improvements on gesture recognition, user identification and system design in the following iterations, as indicated by the arrows in dashed lines in Figure 1.1.

The doctoral work is built on a review of the existing literature on approaches of gesture recognition and gesture-based user identification, and research on gesture-based interaction systems. Help and support from colleagues in the Data Processing and Simulation Group at UPM made the work possible and extended. They include development of a Kinect library for simplified retrieval of data, gesture data collection tools, a voice-based application, the home automation system, the indoor location module, coding of the DTW-based recognition approach in complementary languages (C++, Java), to name a few, which are used in the off-line evaluation, the development of the interaction system and the user study. In addition, previous works in the Group, in particular, the one by Cuevas et al. [24] provided a basis for the author to improve the user interface design.

Figure 1.1: Outline of the research

Chapter 2

State of the art

Gesture-based interaction has found its way in many application scenarios and promises a natural human-computer interaction along with other alternative or complementary methods like speech, touch or gaze-enabled interaction. In certain application scenarios, gesture-based interaction may be more intuitive and meet the particular needs of specific user interfaces, outperforming other interaction manners. The advances in other technology fields benefits the development and research in gesture-based interaction. For example, the availability of more powerful sensors and the prevalence of tangible and wearable devices make the gesture-based interaction increasingly closer to people's normal life. Intense research has been carried out on the problems encountered in the development of the particularities of gesture-based interaction systems, such as the design of the gesture set, the recognition approaches, the use of gestures as an identification mechanism, the social acceptability and the usability of gesture-based interaction systems. In this Chapter, the reviewed related works are organized into three topics: the gesture-based interaction systems particularly in smart environments, gesture recognition techniques and gesture-based identification approaches.

2.1 Gesture-based interaction in smart environments

A gesture-based interaction system, from the implementation point of view, is composed of two well-separated parts, which will be here referred as 'frontend' and 'backend'. The frontend will need a sensor to capture the movement of hands and a gesture-recognition module to tell the backend what gesture the user has performed. The user's feedback, i.e. the output in forms of visual, audio or haptic, will inform about the results and status of the system. The backend will then trigger operations according to the recognized gesture and a predefined control logic. The allowed operations will depend on the application scenarios.

Surrounding the topic of gesture-based interaction, the reviewed literature is organized into seven groups:

- Application scenarios. An overview of the current application areas of 3D motion hand gesture-based interaction, including the smart space scenario

CHAPTER 2. STATE OF THE ART

- Sensors. The different types of sensors that capture hand movements and how they are utilized in the research community and commercial products
- Gesture set design. The development of the methodology to determine a gesture vocabulary, which is nonstandard and is the media to interact with the system
- Gesture taxonomy. The developed taxonomy of motion gestures by the previous works can make the design space clear for designers
- Device selection. A review of the strategies to specify the target device among the many options to interact with within the framework of gesture-based interaction
- Usability of gesture-based interaction systems. Many works examined whether the users perceive the gesture-based interaction useful
- Social acceptability. Performing gestures can make users feel embarrassed in some usage contexts. The discoveries and the research methods on this topic are reviewed

2.1.1 Application scenarios

Gesture-based interaction has been applied to a wide range of applications scenarios including e.g. operating rooms, entertainment, smart driving, large interactive displays or smart objects control in home environment.

In some cases, gesture-based interaction can be more suitable than other alternatives. For instance, various computerized medical systems are essential and utilized in operating rooms. Surgeons face challenges when trying to e.g. visualize volumetric patient data because the use of classical mouse-driven or touch-based user is not available in the main operating site. The alternative way in which less-skilled assistants provide help to physically control the computerized systems leads to inefficiency due to the additional amount of verbal communication. To offer a solution to this problem, Bigdelou et al. [13] have developed a touchless user interface that gives surgeons direct control over the systems by gestures so surgeons can dynamically associate the gestures with arbitrary intra-operative computerized systems and dynamically change the assignment of gestures to target systems based on the current workflow stages.

Gestures can also be musical expressions. With the music installation by Mayer et al. [67], user stands inside a large dome creating melodies by continuous gestures in mid-air using an invisible two-dimensional melodic table of notes. The music is visualized in “Ring Theme” where notes are visualized by rings of specific color and width according to the pitch and length, and “Hayate Theme” which uses turbulence particle systems to visualize the generated musical expressions by changing particle’s color and number according to the note and length of tones.

Smart driving is another application in which gesture-based interaction may bring benefits. The car’s infotainment system, the instructions cluster, the navigation system and

the smartphone may distract driver's attention from driving. To relieve this problem, Exploride [29], which is an upcoming commercial product of head-up display, integrates gesture-recognition technology and shows system information on a transparent display. Up, down, left and right gestures and voice enable to command Exploride hands-free.

Walter et al. [95] argue that distant interaction through mid-air gestures has several advantages for public display interaction. There will not be hygienic problems introduced by touch-based interaction. It is not necessary to come close to the screen to interact. Furthermore, the mid-air gestures can make aware the interactivity of public displays and may favor performative interaction like expressive and highly visible gestures.

To provide seamless transition in different proximity zones in display interaction, the work by Dingler et al. [27] propose a system that combines technology-transparent interaction spaces across 4 interaction zones: touch, fine-grained, general and coarse gestures. The interaction modality is adjusted based on the proximity of users to the screen.

Mouse and keyboards are inconvenient to interact with large displays. Chattopadhyay et al. [19] develop touchless circular menus to make command selection by simple directional hand movements.

There has been plenty of work on gesture-based interaction in smart environments. Bartolini et al. [8] propose REGALS (Reconfigurable Gesture based Actuator and Low Range Smartifier) that uses a smart object as an interaction device. To address the obstacle to interoperability and mutual understanding of the smart devices and spaces involved, they adopt ontologies to digitally represent physical objects and keep track of the semantic relations between the objects in the environment. With their prototype, a tangible device, the digital representation of smart objects can be created by reading an RFID tag and doing gestures. To control a digital device, the user selects the device by reading its tag and then performs the gesture corresponding to the command to execute. Thanks to the ontology, the mapping between gestures and actions for a controlled device can be dynamically configured. In the work by Rahman et al. [79], the recognized gestures are used to toggle a lamp switch, change the light intensity and play/pause a movie. Kühnel et al. [57] developed a smartphone application by which users push touchscreen buttons that represent devices to control while moving the phone to do gestures. The 7 gestures including 4 swings, tilt, moving down and up, and pointing forward, could control TV, radio, lamps and blinds. Equipped with three ceiling-mounted zoom color cameras with pan/tilt modules, the Advanced Soft Remote Control System [28] detects users' pointing for device selection and recognizes 10 geometric gestures to control various home appliances such as TV, curtains and electric lights. Users in [50] select displays or items on displays by grasping and release them by opening hand in a smart office space using head-mounted devices (HMDs) with a external depth camera.

2.1.2 Sensors

The advent of powerful, small-sized and cheap sensors has made vibrant the development and research of gesture-base applications. Vision and motion sensors are the primary sensors adopted by almost all the revised works.

CHAPTER 2. STATE OF THE ART

Dating back to the year 2000, Starner et al. [88] developed a wearable device for control of home automation systems via gestures that were sensed with an infrared computer vision system in a pendant. As the authors argued, the wearable nature made the device suitable for those with loss of vision, motor skills and mobility, and maintained users' privacy. The interaction system supported control gestures and user-defined gestures, whose recognition results were continuous and discrete respectively.

In [51], users can operate a mobile robot using hand gesture based commands. To overcome the illumination challenges in outdoor environments, an IEEE-1394a Stereo Vision Camera System processes captured images to depth maps, which are the input of gesture detection module. Several static hand gestures command the mobile robot to move, stop and follow the user.

In [79], motion-path based gestures that are different symbols in the air are recognized to interact with the intelligent environment. The integrated services include toggling a lamp switch, changing the light intensity and playing/pausing a movie. Users perform gestures wearing hand gloves with an infrared emitter attached. The hand motion data is captured by an IR camera that tracks the infrared emitter.

After the release of infrared-based depth sensors like the Microsoft Kinect [46] (released in 2010), Leap Motion [61] (released in 2013), there has been a wide adoption of these depth sensors because they are low cost and powerful, being able to track movements with high precision. The Leap Motion controller uses optical sensors and infrared light. The controller tracks hands, fingers and tools in its field of view with high precision and the effective range extends from approximately 25 to 600 millimeters above the device. The Kinect sensor has a color sensor, an IR depth sensor and a microphone array. With its SDK, the sensor can, for example, track moving people using skeletal tracking and determine the distance of an object using depth data. The depth space range of the original Kinect sensor is 0.4m to 3m in Near Range and 0.8m to 4m in Default Range. The RealSense camera [81] by Intel, another depth sensing technology, has similar components as the Kinect with shorter sensing range but its size is much smaller and can be integrated into laptops. Among its many features like facial analysis and sound processing, it can track hands and fingers like Leap Motion.

The work by Dingler et al. [27] explores the possibility of seamless gesture interaction by adjusting the interaction modality based on the proximity of users to the screen of displays. If users just stand by the display, they can directly touch the screen to interact. Leap Motion controller captures the gesture input in gesture zone up to 0.5m in front of screen. The Kinect sensor complements and is responsible to interactions in interaction zones much further.

"Personal space" [41] also uses Leap Motion for gesture interaction near display. The de facto gesture pointing method for cursor manipulation has the "Gorilla Arm Syndrome", requiring the user to hold arm out in an unnatural position and causing shoulder and arm fatigue very quickly. To address this issue, this work proposes that the users rest her elbow on a surface while controlling cursor position based on the position of the palm hovering on the Leap Motion controller. Users' preferred space to be mapped to the screen space is defined through a calibration phase.

The Kinect has been used for interaction with public interactive displays where mice and keyboards are inconvenient. Chattopadhyay et al. [19] develop a touchless circular menus optimized for large displays allowing users to make simple directional movements for selecting commands.

Gesture based musical expression in [67] is feasible with hand tracking techniques powered by a Kinect camera. One of the music installations is a large dome with a radius of around two meters. Inside the dome, the position of users' two hands is mapped onto an invisible melodic table so that users could play notes with movements of hands. Music visualization is projected on the surface of the dome.

The skeletal tracking feature of the Kinect inspires researchers to apply hand pointing gesture to interact with smart environments. To control appliances, Point&Control [17] has a server determine which device the user is pointing at through collision detection and send the control interface of this pointed device to the user's smartphone where control commands are issued. A work in 2006 [28] also selects home appliances by pointing gesture but using normal color cameras. The pointing direction is inferred by detecting the user's face and segmenting the commanding hand with skin color cues. 10 predefined 1D or 2D hand motions further determine what command to issue on the selected device.

Novel vision based sensing technologies are to be developed and investigated. The Kinect captures depth data by emitting near-infrared light patterns projected on the scene. As stated in the work [1], the community is investigating sensing technologies with light spectrum from near-infrared light to the far-infrared band. Thermal cameras can sense heat traces left when a user moves fingers on a surface and does mid-air interaction besides and behind a thermal camera's field-of-view through thermal reflection. This work assesses the recognition accuracy of mid-air gestures sensed through thermal reflection in different settings and provides a material space describing surface reflection characteristics and a guideline to select interactive surface materials using thermal imaging.

Motion sensors, mostly inertial sensors, capture acceleration and/or gyro data of hand or arm motions and are small and cheap enough to be integrated into mass consumer electronics. When they are embedded in hand held devices like the smartphone and the Wii, users have a tangible interface to interact with gestures. GesText [40] is designed for text-entry on accelerometer-only devices. The accelerometer-only design eliminates compounding factors (e.g. the use of buttons) and makes the system to be distance independent compared with the IR-based systems and potentially 'eyes free'. The gestures are wrist-based performed by holding a Wii controller. Letters on a virtual keyboard on a display are selected by gestures for tasks of text-entry.

Gesture-aware remote controls by Bailly et al. [6] are focused on usability problem in the context of TV and "sofa interaction". With the constantly increasing number of available features and services (e.g. hundreds of channels and digital media in Web services), the navigation in long lists or deep hierarchical menus is cumbersome with buttons if traditional remote controls are used. They discuss the feasibility and advantages of mid-air gesture based remote controls to improve the interaction in the context of interactive television. Previous research has shown that mid-air gestures are well accepted by users in the domestic

CHAPTER 2. STATE OF THE ART

context [82]. This interaction manner will not consume real estate on the remote control and thus the input vocabulary can be enriched at no cost with regard to device size. No visual attention on the remote control needs be paid when interacting with the TV. Some gestures easy to remember can act as shortcuts for performing favorite commands. And finally, the required sensors like accelerometers or gyroscopes are low-cost. In this work, Wii remote control is used to explore user preference of in-air gesture types and efficiency of this gesture-based interactive method in item selection tasks.

“I’m home” [58] investigates the possibility of interacting with smart environments, e.g. smart home environments, with mobile phones, which are ubiquitous and equipped with multiple sensors and multi-touch screens. An application for Apple iPhone is implemented using accelerometer data for gesture recognition. In a preliminary study of this work, the TV, lamps, hi-fi systems and blinds are mentioned most often among the devices the participants would want to control.

Motion sensors can also be worn on the body. The touchless, gesture-based interaction framework for operating rooms [13], mentioned above, places on arms of surgeon a few wireless inertial sensors, avoiding dependence on illumination and line-of-sight which might be typical limitations of vision-based sensors. Since the operating room is a collaborative domain and each orientation sensor can be identified uniquely, the proposed framework supports multiple-user gesture interface. Ring ZERO by Logbar [83] is a ring-like commercial wearable device. Users perform mid-air gestures by pressing a button on the ring worn in the index finger. The official website shows applications on the control of presentations, home appliances, VR gaming and apps of the smartphone by gesturing with the ring, which is paired with an app on the smartphone. Users can write “TV” to switch on television and “s” standing for “selfie” to initiate camera, for example. It allows customizing gesture set and associating gestures and actions.

2.1.3 Gesture set design

Suppose that in a certain application scenario (in an operating room, in front of a large display or in a smart home environment) the movement of users can be captured by either vision-based or motion sensors. Then what gestures the users can do to interact? Not like the spoken languages, there is no commonly accepted mid-air gesture sets. The methods to create gesture sets have been investigated in recent years.

In the work [75] Nielsen et al. believe that the objective is “to develop a more efficient interface” for a given application and the most suitable kind of interface should be chosen. If it is determined that gesture interface is suitable then the design of a gesture interface should be specific: specific gestures should be found to make the specific application more efficient for a specific group of users. The technical based approach to choose gestures is criticized because this approach selects simple gestures to recognize but some of these gestures are stressing or impossible to perform to some people, and the mapping between functionalities and the gestures are illogical. The proposed approach is human based, which investigates the people who are going to use the interface. There is no universal gesture vocabularies

and a good gesture vocabulary may only match one specific application and user group. They come up with some principles to design gesture sets, taking into consideration of theories of usability, biomechanics and ergonomics to achieve these principles. For a given application they present a procedure to find an appropriate gesture vocabulary. In the first step, the functions needed by the application is found. Then participants are taken through scenarios under camera surveillance communicating the functions with gestures in a natural way. The recorded video is analyzed to extract the gestures vocabulary by e.g. checking the commonly used gestures, how consistently a gesture is used by different participants. The final step is to benchmark the resulting gesture vocabulary. The semantic interpretation, generalization, intuitivity, memory, learning rate and stress are tested. The test might lead to changes in the vocabulary. In a later evaluation study of the proposed gesture selection paradigm, it is shown that user profile influences the gesture choosing and that participants were not consistent in their choices. To examine what mid-air gesture set is appropriate to interact with omni-directional video (ODV), Rovelto Ruiz et al. [84] adapted the elicitation methodology by Nielsen et al. [75] and developed a user-defined gesture set for time and space manipulations on ODV in both single- and multi-viewer scenarios.

Wobbrock et al. [104] also advocate the participation of users in the gesture set design procedure and point out that the gestures created by system designers out of concern for reliable recognition for many surface computing prototypes, although appropriate for early investigations, are not necessarily reflective of user behavior. There would be a mismatch between the system designers' gesture set and the one chosen by users. To find out what kinds of gestures non-technical users make and what are the important characteristics of such gestures, a guessability study methodology is employed that presents the effects of gestures to participants and elicits the causes to invoke them. Rich qualitative data that illustrates users' mental models was collected by think-aloud protocol and video analysis. Customized software recorded quantitative data such as gesture timing for the performance measurements. The user study leads to a user-defined gesture set based on agreement score. In particular, the authors find that in the study they themselves, which are experts, are either thinking creatively or are hopelessly lost with regard to the elicited gestures from the 20 participants. A taxonomy of surface gestures and implications for technology and design based on the observation are also presented.

The work "I'm home" [58] by Kuehnel et al. shows that the design methodology by Wobbrock et al. [104] can be adapted for three-dimensional gestures in smart home domain. The participants were asked what home appliances they would like to control with their mobile phone in a preliminary survey based on which the referents/effects were developed and the guessability methodology introduced by Wobbrock et al. [104], described above, followed. After a user-defined gesture set is developed, a reverse guessability, i.e. given a gesture/signal imagine what referent would be triggered, to examine the distinctness of the vocabulary, and a memorability test to check gesture suitability are ran, both inspired by the work of Nielsen et al. [75].

With the increasing number of devices to interact with, users may often need to perform complex tasks involving two devices such as sending a media file from one device to another.

CHAPTER 2. STATE OF THE ART

The work by Kray et al. [55] investigates the possibility of simplifying these interactions by using 3D gestures of mobile phone. In order to understand what gestures people would produce naturally, a user study is designed where the participants are asked to perform a gesture for each of the 20 described activities involving phone and tabletop/public display/another phone and then rate the suitability of the gesture for this activity. The recorded video is annotated, and properties of gestures and time based measures of gesture planning and performing are noted down. Implications of the design of future mobile phone are conceived based on the analysis of the elicited phone gestures, in term of necessary sensors to include for gesture detection.

The work “Interaction proxemics” [27] explores a collection of user-defined gestures for display interaction in various zones divided by user’s proximity to the display, in three application scenarios: navigation in documents, controlling applications and map-navigation. The participants were asked to perform a 3D gesture for each action of the application scenarios in each zone and rate the appropriateness of the proposed gestures. It is found that finger gestures are mostly proposed by the participants when they are close to the screen, particularly when the task requires high precision. The entire hand or full body gestures are more feasible in distant zones.

Nebeling et al. [74] developed a Kinect Browser system based on previous study [71], which adopted the guessability methodology using Wizard of Oz, pointing out important limitations of guessability studies: when developing the user-defined interaction set, implementation issues, aspects of system design such as conflicts and consistency of the elicited gesture set are not considered. The Wizard of Oz method for interaction set elicitation encourages the participants to propose gestures without constraints but at the same time makes the interaction distinct from the case under real technical restrictions. To this end, a mixed-initiative elicitation method is proposed so that an unframed interaction set was first obtained using Wizard of Oz and then the participants were exposed to the implemented technology to be seen how they would reconsider their proposed interaction set.

Another reflection on the existing user elicitation study is the work by Lee et al. [62]. Their study investigates how users would manipulate digital content on a distant screen by hand gestures in a living room environment. They argue that a critical omission in the user elicitation study like the one by Wobbrock et al. [104] is feedback that may let a user revise his or her mental model and possibly take new actions. The conventional Wizard of Oz approach provides feedback by the designers but does not allow users to modify pre-defined gesture sets over time and the feedback is predefined. Then a modified Wizard of Oz method is proposed where both performer and recognizer are non-technical users, aiming to make the user study regardless of designers’ intention and current technology. After the user study with the modified Wizard of Oz combined with the user elicitation method, they conclude that the design methodology covers all the areas of expected, desired and sensed actions in design space.

2.1.4 Gesture taxonomy

When a user-defined gesture set was elicited through user study, some works developed gesture taxonomies, which can make design space of motion gestures clear for designers. In the study of motion gestures, which are performed three-dimensional with devices, for mobile interaction [85], a motion gesture taxonomy was constructed and the dimensions of the taxonomy became organized into two classes: gesture mapping and physical characteristics. The nature, temporal and context dimensions belong to the gesture mapping class, dealing with how users map motion gestures to device commands. In the physical characteristics class the kinematic impulse, dimensionality and complexity dimensions describe the characteristics of the gestures themselves.

There are four dimensions in the classification of gestures in the work [58], which investigated user-defined three-dimensional gesture sets using smartphones in smart home environments. Their nature, binding and flow dimensions correspond to the nature, context and temporal dimensions in the gesture mapping class in [85], while the form dimension corresponds to the second dimension of the physical characteristics class in [85] except that the static pose is not included in the taxonomy in [85]. Kinematic impulse and complexity are two dimensions proposed in [85] that were not included in the work [58].

2.1.5 Device selection

The aforementioned guessability study methodology introduces referents/effects to the participants who propose signals/gestures. The elicited gesture set can be used to interact with the system. Among the reviewed works of interaction systems, no matter whether the elicitation methodology has been adopted, the gesture set in some works has default interaction targets, such as the application for surgeons in operating rooms [13], a Kinect-based browser [74], interaction with mobile robots [51], interaction with displays and media center [27], [19], and audio-visual music installation [67]. But if there is more than one device to control, in the smart environment for example, it would be necessary to specify the target device before performing the commanding gestures. The device selection can be implicit such as the work [79] where gestures are highly associated with devices and each relates with an explicit controlling device. Other reviewed studies have an explicit device selection step. RFID tags and visual marks are used to register the device to control in [8] and [56] respectively. The natural pointing gesture is facilitated in [17] and [28] for device selection while the work “Put that there” [15], dealing with control of simple shapes on a large-screen graphics display surface (though no device control), augmented speech interaction with simultaneous pointing and made the voice expression natural and economic. In [58] and [12], this task is completed with a convenient GUI on touch screens of the smartphone. The control of home automation systems with “the gesture pendant” [88] allowed users to indicate the desired target by speaking the name, orientating the body to the device or using RF localization. An independent device selection step allows the reuse of the same gesture for multiple devices, reducing the size of the gesture vocabulary.

2.1.6 Usability of gesture-based interaction systems

Kollee et al. evaluated the usability of a prototype of a head-mounted device (HMD)-based gestural interaction system in a smart office environment [50]. The elicitation study drew out qualitative user feedback on the gestural (mid-air gestures) and non-gestural (voice command and touchpad) input techniques of the HMD and revealed that gesture-based interaction techniques were accepted by the users both in preference and emotional feedback. The second study, which measured task completion time and error rates using the different input techniques, however, did not show a clear winner. Nevertheless, the voice command was considered inefficient and the GUI list view controlled with touchpad was less flexible compared with gestural input when the number and complexity of tasks increase.

Usability experiments were conducted by Rahman et al. in [79] to evaluate their proposed gesture-based system for accessing to Ambient Media Services and more than 75% of the 15 users considered that hand gesture-based interactions are interesting.

In [43] the authors made a questionnaire-based study on the preferred gestures of 37 users to control a TV, VCR, lighting and 3D design software. It is relevant that results showed that different people usually prefer different gestures for the same task. In a second use case, in which users had to control 3D design software, the study underlines that gestures were considered as a natural modality to perform certain tasks, especially for commands with spatial association, in comparison to other interaction options (speech, RFID-based physical tangible objects, physical pointer devices and PDAs).

To elicit user preferences for free-hand gestures for common TV control tasks, authors in [94] conducted a user study with 12 participants. Among the guidelines derived from the experiment, they point out that “a clear preference was observed for drawing letters in mid-air”. Actually, “participants witnessed letters were easy to use and suspected they would also be easy to remember”. In particular, “one participant imagined popping up a volume control by drawing letter V”.

In an investigation of user-defined gesture sets for interoperations between devices like phone, public displays and tabletops, the authors in [55] observed the visible nature of gestures can be a desirable feature for tasks like sending and receiving data as gestures make it clear to others what is going on.

2.1.7 Social acceptability

Another important consideration in the gesture-based interface design is social acceptability because the gestures to perform are often new and socially embarrassing behaviors in certain usage contexts. Williamson et al. [103] evaluated the user experience of performing gesture-based interactions in public spaces “in the wild” in two different cultural settings. The participants used a mobile game application, which allowed gesture input with the head, the wrist and the device, over a span of 6 days. It was found that the capability to support both subtle and performative gestures is important to make the interaction acceptable and that acceptance can be more easily achieved through incremental steps rather than large changes.

Two studies were ran by Rico et al. [82] to determine the factors that influence social acceptability, aiming to improving gesture design and eliminating unacceptable gestures before implementation of gesture recognizer. A web survey, which used video prototypes and asked users their willingness of gestures in certain usage contexts, revealed that audience, location and gesture types (with use of mobile device or not) are among the considerations of individuals to decide to interact with a gesture-based interface. The following study further validated the results in the web survey in real world settings. The authors suggested that gestures similar to the style of the ones in daily life would lead to high acceptance. The observed differences between the two studies demonstrated the users would change their acceptance rates after multiple trials and it is recommended to run at least two trials in real world settings when evaluating social acceptability.

2.2 Gesture recognition techniques

As said in [80], gesture recognition “refers to the whole process of tracking human gestures to their representation and conversion to semantically meaningful commands”. The general pipeline of gesture recognition is shown in Figure 2.1, which is constructed by the author through reading the literature.

2.2.1 Hand tracking

There are two main types of tracking systems, namely the vision-based and inertial-based tracking in the reviewed literature on gesture recognition approaches. The vision-based hand gesture recognition system needs first detect the hand and track it to extract the features of interest for the purpose of recognition. The detection and tracking are basically computer vision problems. Some works in this category use traditional color cameras like [87, 31] while others recognize gestures based on RGB-D data, such as [96, 10, 105, 110, 9]. The gesture trajectory can be obtained directly through Kinect-like devices and software, and the works [101, 14, 107, 31] start the recognition from the obtained gesture trajectory. Although the vision-based tracking enables the users to do free gestures without cumbersome contact-based devices like data glove, it does have some limitations, such as being prone to be interfered by varying lighting conditions and cluttered background and relatively low sampling rate of normal cameras [21]. The inertial based recognition does not have the

Figure 2.1: The pipeline of gesture recognition

above limitations. Some works [18, 109] utilize accelerometer to record the gesture motions while others combine accelerometer with gyroscope [5, 54, 108, 52] and magnetometer [37]. However, the inertial based gesture recognition presents the drawback that the tracking is implicit and it cannot collect position and orientation information. To address the limitations of both tracking types, [21] and [65] fuse the data from both vision and inertial sensors, resulting in more robust recognition.

2.2.2 Gesture spotting

The signals of gestures generated by tracking systems are continuous data streams in which not all hand motions are meaningful gesture patterns. Gesture spotting is the procedure to determine the start and end points of the interested gestures in the data stream. The authors in [70] believe that the gesture spotting is difficult due to the segmentation ambiguity (it is hard to determine exactly the boundaries of the gestures) and the spatio-temporal variability (the signal of the same gesture varies). In addition, the transition of two adjacent gestures is also likely to be segmented as gestures, causing false positives. There are several intentions to address this issue in the reviewed literature, however. The gesture data can be segmented based on thresholds of velocity [21] or of adaptive magnitude [37], local velocity minima to obtain strokes that will be combined to characters after classified [31], trivial rule-based conditions of data points [109] and Gaussian models for rest positions and non-rest positions [110]. In this work however, it is assumed that the gesture signal is pre-isolated, i.e. the gesture signal to be recognized contains only one gesture and the start and end point of the gesture is already known.

2.2.3 Feature extraction

In general, once a gesture signal is segmented, the features of interest will be extracted from the signal and fed into the classifier to get recognition result. The features are the representations of the motion gestures. In some works, they are the (normalized and smoothed) original signal of a) accelerometer and gyroscope [5, 54, 52], b) linear acceleration, angular velocity, Euler orientation of the hand and its position relative to the shoulder center joint [110], c) accelerometer, gyroscope and hand position coordinates [65]. The research in [20] investigates the relative effectiveness of various feature dimensions for motion gesture recognition, including the different combinations of position, velocity, acceleration, orientation and angular speed. Its collected database provides a common ground to compare recognition performance given different tracking signals and approaches. Sometimes the features to extract are obvious and highly related to the classifier such as the case of [18] where the two RVM (Relevance Vector Machine) approaches requires the calculations of frame-based descriptor and DTW matching. Some features are specifically designed to better represent the characteristics of gestures, to fuse different data types, to cope with the variances of gesture performance and captured signal, or to be indexed for recognition:

- *Features for better representation.* Among the reviewed literature, the tangent values are extracted from gesture trajectories as features for the HMMs [101] and for the Most Probable Longest Common Subsequence classifier [31]. Features closely related to human activity such as motion energy, posture and shape related features are used in [108] to classify gestures.
- *Features for data fusion.* The random visual feature, extracted from visual data, and DCT (Discrete cosine transformation) features extracted from acceleration signals are fused by learning a shared hidden space with the penalized maximum likelihood estimation [21].
- *Features for invariance of view and performance.* To make the Kinect-based recognition position-invariant, the coordinates of gesture trajectory are transformed from camera-based to body-based and the displacement of the coordinates between test sample and template is compensated in [14]. Based on the Kinect as [14], the work of [107] aims at making recognition view-invariant by extracting shape features that are invariant to rotation, scale and translation transformation of the gesture trajectory. In addition, the orientation features indicating the rotation state of a trajectory are complement of the shape features and help distinguish gestures with similar shapes. The approach in [18] is enhanced and device-orientation-independent with a proposed tool called PrePCA. Sign sequences of acceleration data are extracted to compress data and to minimize the influence of variations between users in [109].
- *Features for indexing.* Based on the bag of feature model, the 3D enhanced motion scale-invariant feature transformation fuses the RGB-D data, and the features are clustered with the k-mean algorithm to learn a visual codebook to which the features will be mapped to have the recognition result [96]. In [87], the descriptors that characterize the local motion patterns are indexed and the gestures are recognized by matching against the database through a term frequency-inverse document frequency (TF-IDF) weighting scheme.

2.2.4 Classification

Regarding classifiers, they can be divided into deterministic and statistical. The template-based approaches belong to deterministic approaches, which measure the similarity of features between unknown gesture signals and templates, and select the template (or cluster classified based on similarity) that best matches the testing pattern as recognition result. The reviewed approaches of similarity measurement include Dynamic Time Warping (DTW) [3], Longest Common Subsequence (LCS) [91], Protractor3D [53] and Levenshtein edit distance [39]. On the other hand, the reviewed statistical approaches model the motions in a probabilistic manner such as Hidden Markov Models (HMMs) [112, 63, 20], or are based on probabilistic theory, like Probabilistic Neural Network (PNN) [97], which tries to asymptotically approach the Bayes optimal decision surface. On purpose of modeling the

CHAPTER 2. STATE OF THE ART

variability of intra-class gestures, the template-based classifiers are extended with probabilistic approaches and the Gaussian Mixture Model (GMM) is used to learn a probabilistic template for each gesture in [10] and [31], based on Dynamic Time Warping (DTW) and Longest Common Subsequence (LCS) respectively. The authors in [54] argue that the template-based approaches perform well even with few training samples and they are easier to implement and deploy on different types of devices since they do not require specialized libraries. What is more, the probabilistic assumptions on which the statistical approaches are established may not accomplish in practice, for example, the Markov assumption and the stationary assumption set by HMMs. The survey of gesture recognition [70] mentions that the HMMs are computationally expensive and require large amount of training data. Since the vocabulary collection, gesture dataset and experiment settings are different in these works, it makes no sense to compare directly the recognition accuracies.

The recognition system may detect gestures in user-dependent or user-independent manner. The former gives more creativity to the users and allows them to personalize the gesture vocabulary as they prefer so that the gestures are intuitive and familiar for them, leading to better learnability. With the latter, the user-independent manner, the gestures of users can be directly recognized without the boring training step required by the user-dependent manner, but the gesture vocabulary is normally predefined and fixed, and may not be accustomed for users. The authors in [64], which applied the DTW as recognizer, show the reasons why plenty of work is targeted at user-dependent gesture recognition only. First, user-independent gesture recognition is difficult, due to the great variation between different users even for the same predefined gesture. If the collected data are evaluated in a user independent way instead of user-dependent, the recognition accuracy decreases significantly, from 98.4% to 75.4% as the case of that paper. Second, user independent gesture recognition may not be as attractive as speaker-independent speech recognition because there is no standard or commonly accepted gesture sets. The template-based approaches seem not to be suitable in the user-independent situation because of the huge performance variations among different users, which mean many more templates to match to cover the variations, causing deficiency in computation. The statistical approaches, however, do not have this problem and have obtained impressive results on user-independent tests, such as in [20].

The approaches mentioned above are all learning-based, i.e. they can learn from the training samples to recognize gestures and adapt to the changes in the gesture vocabulary and features extracted from motion signals. On the contrary, rule based ones [100, 109] do not need the learning step and normally run very fast. They can work in user-independent manner. However, they need plenty of knowledge about the gesture set to define the judging rules and the vocabulary is less flexible because the rules have strong association with the vocabulary. The learning-based approaches are preferable taken into consideration that the gesture vocabulary varies with different application scenarios and that features of motion signals are distinct for different sensors and platforms.

2.2.5 Efficient training

If the learning-based approaches are to be used, to make the gesture input practical, it would be reasonable to design the system to be able to accurately recognize gestures with as little training from new users as possible. Several works have developed recognition systems requiring only one training sample per gesture class. Based on the bag of features (BoF) model, the work of [96] proposes a new feature that is a more compact and richer visual representation. After a visual codebook is learned, a sparse coding approach is applied for recognition, obtaining lower reconstruction error and better performance than traditional approaches of mapping. Making use of the training set of known users, the work in [52] learns the new user’s performance variation from her single training example and then maps the input gesture of the new user to the training set of a known user by linear transformation before applying the HMMs classifier. With a filter designed to reduce integral errors of inertial sensors and a minimal intra-class to maximal inter-class based template selection approach (min-max template selection approach), the DTW-based recognizer in [37] reaches high recognition rate with one training sample.

2.2.6 Criteria to choose a recognition approach

The authors in [54] recommend three criteria to choose gesture recognition algorithm: number of training samples available, performance constraints of the target devices, and time available for coding. They believe that template-based approaches perform well even with few training samples, and are easier to implement and deploy on different types of devices without specialized libraries. In a review of the Dynamic Time Warping (DTW) algorithm in [20], the authors comment this template-based approach to be useful for personalized gesture recognition in user-dependent applications, where a large set of training samples is hard to collect. In the user-independent case where large intra-class gesture variations exist, explicit statistical modeling of the variability is necessary due to performance and computational load.

2.3 Gesture-based identification approaches

In biometry, identification is a variation on a token-based security scheme, in which the token is some physical or behavioral attribute of the person. Biometric physical attributes include fingerprint, hand geometry, iris, retina, face, vein pattern, etc. Unfortunately, some of these characteristics like face and fingerprint can be easily intercepted. In addition, the identification procedure working on some of these attributes might be invasive to users. Behavioral attributes, those based on aware or subconscious human movements (signatures, keystroke dynamics, gait, gestures, etc.), are revocable and may contain significant information about a given person [38].

In particular, some previous works explore the potential and feasibility of gestures for user identification. As a first step to investigate the relationship between the way of tooth

CHAPTER 2. STATE OF THE ART

brushing and cognitive disease, the work [49] identifies users based on toothbrushing information using an embedded accelerometer. Guna et al. [32] propose an intuitive identification system that uses ‘implicit’ gestures, which include ‘pick-up’ and ‘shaking’ movement acquired by accelerometer in a Nintendo WiiMote remote controller. These loosely defined gestures are simple, non-invasive, user-friendly and for all users. The developed system could be used for applications such as user login to home multimedia services, with less strict security requirements. Apart from accelerometers, many of the reviewed work utilize the Kinect to capture the gesture movements. Eiji et al. [34] propose a system that can identify users based on a hand waving gesture of 2 seconds detected by the Kinect. The identification engine extracts features of body segment lengths and hand-waving gesture patterns. Instead of hand waving gestures, the system by Mendels et al. [69] is based on free-air hand signature-gestures acquired with 3D camera. For system training, an ‘interactive enrollment’ module indicates the variance of the performed motion-gestures and assists the user in obtaining a robust signature; this significantly improves the recognition accuracy. The proposed system can be integrated into gesture-based home entertainment systems with a group size of 3-5 and used for interface customization, content adaption and parental control. To access the tiled-display system for visualization, the users are required to gesture a series of digits as identity passwords in [111]. Wu et al. [106] make use of the ability of skeleton tracking of the Kinect to identify users by full body gestures like swing or punch. Their work shows that skeleton models contain unique user information.

The identification procedure is similar to the pipeline of gesture recognition shown in Figure 2.1. The gesture signal, either from hand gestures or body movements, captured by accelerometer or 3D camera, can be described by multi-dimensional time-varying sequences. In the reviewed literature, the identification problem is induced as a classification problem. For example, some works [32, 106] apply the Dynamic Time Warping (DTW) algorithm, which is a traditional template matching approach. In [32], a “super-gesture” concept, which combines acceleration features of multiple gestures in only one gesture template per user, is proposed to increase the computational efficiency and temporal stability. The DTW algorithm is chosen because it has the inherent abilities to compare time series of different lengths and sample rates, and it is simple to implement. A user evaluation spanning over 10 days tested both explicit signature gestures and implicit gestures and showed that implicit, intuitive gesture based user identification is possible. In [106], the DTW is used to identify users based on 8 body gestures and achieves an Equal Error Rate (ERR, the rate at which both acceptance and rejection errors are equal) of around 4%. Instead of using DTW to measure matching cost, the system based on free-air signature gestures [69] aligns the incoming samples to each cluster of training data of each user and calculates the distance in the Neighborhood Components Analysis (NCA) space between the aligned sample and the average cluster feature vector, based on which and a threshold, the sample is classified to the closest cluster so that the user identity is determined. The work claims average accuracies of 98%-92% for personal signature-gestures. The body-based user identification system in [34] trains a Support Vector Machine (SVM) classifier to identify users. After two lab studies, the system achieves impressive 0.5 to 1.6% ERR in the field study. The identification system

for tiled-display system [111] captures the trajectory movement of single hand joints in the x-y dimension and represents the trajectory of digits as the sequence of movement direction vectors that are quantized. A two-phase cascading classifier combining the classifiers HMM (Hidden Markov Models) and GSS (Golden Section Search) is utilized to mitigate the misclassification problem of pure HMM based solution.

2.4 Conclusion

As an interaction mode, gestures are suitable for certain application contexts. The research community has developed different gesture-based interaction systems for a wide range of application scenarios. In this Chapter, we have seen them in an operating room where the interaction system gives the surgeons direct control over the computerized medical systems, for entertainment where people use gestures to create melodies, in smart driving where gestures along with voice allow the driver to concentrate on driving, in the interaction with public displays in which mid-air gestures make the interaction more expressive and comfortable than touch-based methods, and in a smart environment where gestures become a natural and intuitive interaction option.

Most of the reviewed works utilize vision or motion sensors. The vision sensors work with hand-free gestures but might be subject to limitations of external environments like illumination and line-of-sight. Motion sensors normally have to be held or worn by the users and do not suffer the vision constraints. Some vision sensors are standalone like the Kinect and the Leap Motion while others are integrated like the gesture pendant [88] and RealSense camera. All the motion sensors (accelerometers) in the reviewed works are integrated in the smartphone, the Wii, or the Ring ZERO except the wireless inertial sensors in [13]. The type (vision or motion-based), the capability (sensing range, precision etc.) and the sensor's integration format (standalone or integrated into a computing platform, the size) will project constraints on the application scenario and performance of the application based on it.

Since there is no universal gesture vocabularies, researchers have been exploring methods to create gesture sets. Nielsen et al. [75] propose design principles and procedure of gesture set design in a human-based strategy. The guessability method introduced by Wobbrock et al. [104], which presents the effects (results) of gesture to participants and elicits the causes, turns out to be a classic method and has been adopted by many other works. Then Nebeling et al. [74] points out that the limitation of the guessability method is that it does not consider real technical restrictions and aspects of system design such as conflicts and consistency of the elicited gesture set. They recommend a mixed elicitation method considering both the unframed gesture set and an examination under implemented technology. Lee et al. [62] argue the lack of feedback that allow users to revise their actions in the guessability method and extend it by suggesting a modified Wizard of Oz methodology to make the user study independent of designers' intention and current technology.

To select the target to control in smart environment, the systems in the reviewed literature have used RFID, visual marks, natural pointing, combination of speaking and pointing

CHAPTER 2. STATE OF THE ART

(“Put-that-there”), GUI on touch screen and context information such as body orientation and localization.

Many works have conducted usability tests to evaluate the gesture-based interactions. Mid-air gestural input was preferred to head gestures and rated more favorably compared to the non-gestural input techniques available on existing HMDs (Head Mounted Displays) for tasks such as moving an image thumbnail from one display to another [50]. Gestures were considered as a natural modality to perform tasks of 3D design software, especially for commands with spatial association [43]. Gestures were also found useful in application scenarios of TV control tasks and interoperations between devices like phone, displays and tabletops.

Since doing gestures may result in new and embarrassing behaviors, we should pay attention to the social acceptability of gesture-based interaction proposals. Factors that influence the social acceptability are explored in the literature to guide gesture design, and suggestions to improve social acceptability are also proposed.

This work explores how to create a sound gesture-based interaction model, in particular to be applied in a smart environment for smart objects control and networking. The objective is that the proposed method can be implemented over different devices, based on both vision and motion sensors. The gesture set of the interaction system is pre-defined, in order to remove the potential factors of conflicts and inconsistency of user-defined gesture sets, since the main purpose of this work is not to elicit a gesture set but to evaluate a gesture-based interaction model in a quasi-real environment. In any case, the participants were asked to come up with their preferred gestures for certain tasks for later study. The control target and the operation are specified in a proposed grammar structure, which inherits from the imperative sentence in natural languages. Usability and social acceptability are subjective measurements depending on many factors like the application scenario, design and implementation of the interaction system, culture, etc. They are also evaluated in the user study.

The reviewed works on gesture recognition techniques are presented and organized around the pipeline of gesture recognition (Figure 2.1) and the discussed topics involve tracking systems, gesture spotting, feature types, classifiers, one-shot learning recognition and recognition algorithm design. As we will see in Chapter 4, the template-based approaches are more suitable to the design purpose and it will work based on both optical and inertial tracking systems. Instead of exploring complex techniques of automatic gesture spotting, the work will design gesture delimiters in user interface to explicitly mark the ends of gesture signals. The utilized features and classifiers will be described and an approach to reduce training samples will be investigated.

Behavioral attributes such as gestures are less invasive than the biometric physical ones for user identification and have the support of low-cost hardware like the smartphone and the Kinect, making feasible the gesture-based identification for home environment, as shown in many of the reviewed works. In addition to the availability of enabling technologies, the identification system should require small amount of training, provide personalized gestures and be user-friendly to non-expert users, as discussed in [69]. The gesture-based identification

CONCLUSION

system also faces challenges such as the similarity of gesture performance among different users and the variability of gesture performance of the same user over time. In Chapter 4, the requirements and challenges of gesture-based identification are discussed and an identification approach is proposed and tested.

Chapter 3

GENIE: A Gesture-ENabled Interaction method for smart Environments

To investigate design issues of gesture-based interaction in smart spaces, this Chapter considers the design space and proposes the interaction system and user interface.

3.1 The design space

The interaction system is to facilitate its users to control resources in smart spaces through in-air motion gestures. With this problem in mind, some design decisions will need to be made, e.g. regarding a) devices enabling the acquisition of user's gestures, b) kind and capability of the computing platforms, c) characteristics of the gesture recognition approach, d) degree of expressiveness to manage the attributes and functions of the resources to control, and e) impact of human factors like mental model and ergonomic requirements.

Certainly the gesture itself as a communication tool projects constraints, such as expressivity, memorability and intuitiveness.

Then the sensors have limitations on precision, signal-to-noise ratio, range of measurement, sampling rate, delay, sensing availability, etc., and external conditions have an impact on their performance. For example, optical sensors have the occlusion problem and lighting conditions affect the measured data. It is possible that the acquired data are an indirect or partial representation of the target phenomenon, or a mixture with other uninterested data.

After captured by the sensors, the gesture raw data are processed in a computing platform, which has a certain amount of computing power, modes of feedback and a degree of mobility. Comparing to the smartphone, a normal computer does not have haptic feedback and is less movable. Furthermore, the gesture recognition approach will present detection errors and introduce delays in the interaction flow.

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

Figure 3.1: The interface design described with the instrumental interaction model

The target resources to control also shapes the design space by constraining the set of user tasks.

Apart from the technical concerns, the human factors play an important role in the design process. People have a foundation of knowledge of the world and the design should take advantage of this knowledge to facilitate learning and understanding instead of violating it. In addition, it is very likely that the users get tired with high physical and mental demands from the interaction system.

Considering all these aspects, this Chapter covers the design of a system that supports gesture-based interaction models. This system has been designed to provide a natural, usable and highly engaging interface in terms of user experience.

The rest of the Chapter explains the interaction method design and implementation details.

3.2 A gesture-based grammar to command the smart environment

Among the controllable resources in smart space, it is possible to find e.g. smart home objects (blinds, an HVAC system, table lamps, ceiling lights, etc.), media/service presentation objects (a TV set, user's smartphone, a photo frame, etc.) or moving objects (e.g. robots). Each of these devices is capable of performing a set of actions: for example, a table lamp may be switched on or off, the light color can be modified and the light level, raised or lowered. They are scattered in the smart space and there is not necessarily only one instance of a certain type of resource in the whole space. For example, multiple rooms can have the same installation of lights.

Instrumental interaction model [11], proposed by Beaudouin-Lafon, "is based on how we naturally use tools (or instruments) to manipulate objects of interest in physical world". The interface design using the instrumental interaction model is shown in Figure 3.1, which

is adapted from the template for the interaction graphs [90], a handout of a course by the author of [11]. The gesture-based interaction system works as an instrument between users and target objects, transforming the user’s actions into commands and providing feedback.

To express commands, a gesture grammar is proposed as a metaphor of subject-predicate-object structure in imperative sentences. A previous work on gesture set for free-hand TV control [94] reveals that users prefer one-hand gestures than two hands and prefer drawing letters in mid-air to represent commands (e.g. C for close, H for help etc.). The interaction systems in [28] and [43] employ geometric motions for gestural commands (e.g. clockwise movement to turn on TV, directional motions to move 3D models). Inspired by the previous works, the subject or object in the grammar is represented by the in-air drawing of initial letter or acronym of the resource to control and the predicate is the operation associated with an intuitive geometric gesture. Therefore, the grammar-based gesture command specifies the type and operation of the resource. The location context further determines the concrete resource individual. The object component in the imperative sentence is not always necessary. The Object in the interaction model in Figure 3.1 is the controllable resource in the smart space such as blinds, lights and entertainments. In the current design, there is no order or dependence among the operations on different resources. To lower down a blind for instance, the user will do a “b-down” gesture chain and the interaction system transforms this action to the command to act on the blind device in her location. The user receives feedback from the interaction system and sees the blind moving down. An example of a command with object component could be “sending video from TV to tablet” with gesture chain “tv-backward-t”.

Since there might be multiple different resources to control and each resource has several commonly used operations in the smart space, assigning one gesture to each operation of each resource [79] is probably not practical due to the possible large size of gesture vocabularies to remember. The proposed grammar, which is a metaphor of the imperative sentence to make it easy to learn, provides a hierarchical structure to organize the resource entities and operations, facilitating the reuse of gestures representing operations and thus making the gesture vocabulary more intuitive, compact and efficient than the one-gesture-command in a multi-resource scenario. For example, the “switching on” operation of music, movie and radio can all be represented by a circle gesture with the proposed grammar. The one-gesture-command manner on the other hand would require three different gestures that might have little association to the operation.

The gesture vocabulary of the grammar used in this work include in-air drawings of letters and geometric trajectories like directional movements, circles and spirals. According to the gesture taxonomy presented in [85], the gestures are “in-context” since the resources specified by letter-drawing gestures depend on users’ location and the triggered operations depends on selected resources. The letter-drawing gestures are “symbolic” and the geometric ones are “physical”, and they are both “discrete”, “tri-axis” and “simple”. Although ideally they are in 2D or 1D, in practice the trajectories the users performed are in 3D space, not restricted in a straight line or a plane, and the captured gesture signals are 3D. So we call them 3D gestures in this article.

3.3 Multi-device GENIE architecture and implementation details

To create a gesture-based interaction system of high usability, engaging user experience and natural interaction, the system needs to have:

- High gesture recognition rate. Low recognition rate makes users frustrated and the control ineffective.
- Real-time response. The stages in the interaction workflow should be in real-time, including gesture input, recognition and feedback.
- Rich and clear feedback. Despite of the constraints on the resources that the combination of devices and computing platforms can provide, the interface should indicate the status of the system and interaction in multimode if possible and appropriate, supporting the users in the interaction and making the users feel control. The components of the interface should be specific and easy to understand.
- Low physical and mental exertion. For an application designed for daily usage, high physical and mental demand probably makes users resistant to use.
- Easiness to learn. The motion gesture-based interaction is relatively new and it would be important for the interface to be easy to pick up for non-expert users, facilitating technology acceptance.

In the rest of this Section, the architecture, the interaction workflow of the system will be discussed and we will see the system is implemented around the interaction paradigm of the gestural grammar. And in particular, Section 3.3.3 explores the design of user interface such as gesture input procedure, control components and feedbacks, considering constraints and characteristics of the three enabling technologies: the Kinect, the smartphone and the smartwatch.

3.3.1 Architecture

The interaction system was designed as client-server architecture, as shown in Figure 3.2, aiming at interface development. The client side equipped with sensors is able to collect time sequence of hand gesture motions and then recognizes them. In order to control the resources in smart spaces, as discussed in Chapter 3.2, the users will need to launch a command by composing a gesture sequence. Each time that a user performs a gesture, the recognized result is sent to the server, which then processes and maintains the analysis status of the (partial) gesture sequence and sends back different notifications expressing whether the (partial) gesture sequence is correct in terms of grammar definition and whether it is completed. Once the users have composed the correct gesture sequence, the server executes

Figure 3.2: The architecture of the interaction system

the corresponding command by sending requests to a home automation server or to the smart devices directly.

To serve the research purpose, the developed client side applications are built on top of three different interaction technologies: tangible/graspable, wearable and invisible natural interaction technologies, which are implemented for the smartphone, the smartwatch and the desktop environment respectively. As discussed in the contribution section in Chapter 1.3, very few previous works incorporate multiple independent gestural modalities in one interaction system and compare different technologies of gesture-based interactions. The first two devices sense gestures with embedded accelerometer and the third with a Kinect camera. The basic functionalities include a front-end user interface that guides the users to perform gestures and to control the process of gesture sequence input, as well as gesture movement capture, gesture training, gesture recognition (except the Kinect), localization and communication with the server.

The end-user interface on the client side is introduced in the end-user interface part later. Since current smartwatch technology still has low computational capability, the gesture data captured from the smartwatch are sent to the smartphone via Bluetooth for gesture recognition and gesture template selection in training. The smartphone paired with a smartwatch also works as a repeater between the server and the smartwatch. Users' localization information specifies the target resource instance to control, since multiple resources of the same

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

type may appear in the smart space while the gesture of the initial letter or acronym only indicates the type of the resource. An indoor localization module developed within the Research Group (MOBILOC) has been integrated in the smartphone application and powers this feature of location-aware resource selection for the applications in the smartphone and the smartwatch. Since the Kinect devices are stationary, their locations are hard-coded. In addition, to avoid training for each instance of the Kinect application (multiple Kinect cameras are deployed in different rooms, refer to Figure 5.1), the templates and gesture recognition modules reside in the server side for the Kinect clients.

The common tasks like gesture sequence analysis and gesture command execution are centralized in the server side. As proposed in Chapter 3.2, the grammar can be described as a resource followed by an operation ending with or without a parameter. It is formally described in the following Grammar (V_n, V_t, P, S) which is defined by a vocabulary of non-terminal symbols $V_n = \{S, A, B, R, O\}$, a vocabulary of terminal symbols $V_t = \{R_i, O_i\}$, a set P of production rules and start symbol S . The production rules are:

1. $S \rightarrow A$
2. $S \rightarrow B$
3. $A \rightarrow RO$
4. $B \rightarrow ROR$
5. $R \rightarrow R_i$
6. $O \rightarrow O_i$

The terminal symbols R_i and O_i stand for concrete instances of resources and operations respectively. The A represents the case when the object component is omitted. These production rules are used to determine whether a sequence composed of R_i and O_i conforms with the grammar definition.

The grammar's syntactic rules are hard coded in the program of the gesture sequence analyzer while the gesture vocabulary (and the syntactic type of each gesture) and semantic meanings are defined in database. Once the analyzer receives a gesture input, it takes into consideration the previous inputs and check whether it conforms to syntactic and semantic rules, pointing out grammar violation if there is any mismatch, without having to wait until the finish of gesture sequence input. It also accepts input flow-control messages issued from the client side to rollback one gesture, restart the composition procedure, confirm or negate the composed gesture sequence once it is correct. With the gesture sequence and users' location, the server sends REST commands or socket messages to control the resources.

Thanks to the client-server structure, the modules such as integration of controllable resources and gesture-interaction interface of end user have little coupling and redundancy, making the system scalable on the two ends and easy to maintain. In addition, the current implementation allows multiple clients to interact with the server simultaneously.

Figure 3.3: The interaction workflow chart

3.3.2 Interaction workflow

The main interaction procedure of the developed system to control resources in smart space is described in Figure 3.3 along with the training procedure. This workflow is consistent among the interfaces of the three types of clients, i.e. the Kinect, the smartphone and the smartwatch. Due to the user-dependent nature of the gesture recognition approach in this work, the user needs to be identified at the beginning. The identification approach can be traditional user login by typing user name and password, or gesture-based manner. For the introduction to the proposed gesture recognition and gesture-based identification approaches, please refer to Chapter 4. Since the gesture recognition approach is template-based, the system only recognizes the trained gestures and once the user enters the system identified, if there are no gesture templates of this user because it is the first time she uses the system for example, the training is required. Then, once the user has trained the gestures of the commands she will use, she will start to compose the gesture sequence of commands, performing gestures one by one. After each gesture, if the data are conveniently captured of course, the system will try to recognize it and if the gesture is too different from any of the gesture templates, the user will be notified that the gesture was not recognized. Otherwise, the gesture sequence analyzer checks whether the recognized gesture (it can be different from

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

user's intention due to recognition errors) violates the grammar rules given the previously recognized gestures in this gesture sequence composition. If the sequence is not completed yet, the user will continue for the next gesture component. In case the sequence is completed, the system will execute the command. Before the user performs the next gesture, she has the option to rollback to the previous gesture of the chain (deleting the last gesture) or restart the whole composition procedure if she finds that the recognized gesture is not what she wants because of a recognition error or her confusion. If one gesture input does violate the grammar rule, the user has to repeat this gesture to continue. In the implementation, this repetition is direct without having to issue a rollback command. However, as we will see later in the user study, this made confusion.

As for the training procedure, at the beginning the user chooses what gestures, defined in the gesture database, she wants to train and how many times she wants to repeat each of them. The interface guides the training by showing the next name of the gesture to do in random order up to finish the training. If the user does not like one gesture performance in the training, she can rollback and repeat it. The training procedure can be started anytime except during gesture performance as shown in Figure 3.3.

Figure 3.3 shows the normal interaction workflow. However, to give the users more control and convenience, the options to rollback one gesture and restart command composition are always available after each gesture primitive even the last recognized gesture violates the grammar rules. Similarly, the users do not have to finish the training procedure and can give up at any time.

3.3.3 The end-user interface

To support a smooth workflow, the end-user interface plays an important role. The research presented in [24] conducted a user study to evaluate a Kinect-based user identification system and many valuable user feedbacks were collected and inspired the design of the training function as well as the gesture input procedure of this work. This identification system enabled the users to identify or authenticate themselves by performing in-air gestures; in particular a question-driven service facilitated the collection of different gesture samples as responses to queries about the user, making it possible to disambiguate the user's identity thanks to the gesture chain. A feature of the system is that it launched a timer with a fixed duration during which the user performed gestures. Some users complained they did not have enough time to complete the long signature gesture when it was requested for authentication. In addition, the users had to wait until the timer started to do a gesture instead of starting a gesture whenever they wanted. There was no feedback of whether the gesture was performed well. The training required to repeat a gesture several times, then manually select another gesture and repeat for the same times, and so on until all gestures were trained. The users commented that the training was very repetitive and easy to lose concentration, mechanic and tiring. Additionally, it was not possible to go back after the training started. To address the users' concerns in that user study, the design of this gesture-based interaction aims at a) personalizing the gesturing time, b) giving users the control on when to perform gestures and

c) providing enough feedback of gesture performance. Since gesture segmentation is difficult due to ambiguity (it is hard to determine exactly the boundaries of the gestures) and the spatio-temporal variability (the signal of the same gesture varies) [70], and [92] suggests avoid temporal segmentation if feasible, the system in this work will record one individual gesture after another and segment gestures with artificial delimiters. As for the training procedure, the application will require as few training repetitions as possible, automatically switch to the next gesture to train and provide the option to redo gesture repetition(s). The workflow of training procedure shown in Figure 3.3 is the result of the reflection on this previous work [24].

This subsection describes the design of the gesture input procedure on the three technologies, i.e. the Kinect, the smartphone and the smartwatch, as well as the interface for training and control.

An individual gesture can be described in three phases, as stated in [95]: *registration* that indicates the beginning of the gesture, *continuation* that is the gesture movement and *termination* that marks the end of the gesture. As the gesture recognition approach in this work does not perform temporal segmentation of gestures, artificial delimiters are necessary to indicate the range of the interested gesture input and distinguish it from daily hand movements to avoid unintended interaction. In fact, gesture segmentation on continuous gesture data can introduce extra errors that degenerate recognition accuracy and require more computing power from the computing platform. Given the natures of the three technologies in term of gesture input and having the mind of keeping the beginning and end indication as natural and effortless as possible, the gesture input procedures were designed for each of the three technologies and they are described in the following.

Kinect

In the case of the Kinect, which tracks the 3D position of hands, the users do freehand gestures and the system considers the gesture has started when the velocity of the hand surpasses a threshold and ended if the velocity drops below that threshold. This speed threshold-based approach to segment gesture was also used in [28]. The value of the threshold was determined through experiments. The captured gesture data is time series of 3D positions of the hand during the two moments. Several auxiliary visual feedback gadgets are to help ensure the gesture input of high quality and an engaging user experience (see Figure 3.4d):

- *The silhouette.* Representing users with silhouette was partially inspired by the work [95], which researches on revelation strategies of initial gestures based on silhouette representation for interaction with displays. The system supports only one user's input at a time and the gesturing user may want to know whether she is being tracked. In the case that several persons appear in front of the device, it would be necessary to indicate who is being tracked. The user whose silhouette is in yellow is active and this silhouette clarifies all the above doubts.

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

Figure 3.4: Illustration of the gesture input procedure of the three technologies. Each column shows the gesture input manner and snapshot of users doing gestures. Columns from left to right: the Kinect, the smartphone and the smartwatch

- *The dot on the hand of the silhouette.* Users need to wave hand to activate hand tracking that is sometimes not stable, so it would be essential to provide feedback of the status of the hand tracking. To this end, a dot, which is like a cursor, appears and attaches to the hand of the silhouette of the active user when the application is tracking the hand and it moves along with the hand. The application shows the positions of the dots in the past few moments so that the moving trace of the hand is visualized. The color of the dots changes from red to green when the system starts collecting gesture data.
- *The sweet spot.* A similar strategy to guide users to a target position (within the sensing area of the Kinect in this case) in front of displays with visual cues in [4] is utilized. The Kinect v1 has 43 degrees vertical by 57 degrees horizontal field of view. It could be quite probable that the user's gesturing hand moves out of the view hence making the gesture data broken. In the application, a "safe" area relative to the camera is defined so that when users stand there even extreme stretched arm will locate inside the field of view of the Kinect. This area is called sweet spot, in that doing gestures there enhances the integrity of gesture data and benefits the gesture recognition and interaction. A visual feedback gadget shows color video from the Kinect in the lower left corner of the interface of the Kinect application, see Figure 3.4d. It notifies users that are outside of the sweet spot by rendering effects on the color video. For example, when the user is

Figure 3.5 shows the area of sweet spot. Let us approximate the space where the user's hand can reach in the interaction as cylinder by imaging the human body as a line that is the axis of the cylinder and the farthest horizontal distance from hand to this axis as radius, as shown in the circle O' . Then looking down to the ground, as shown in the horizontal view in the figure, the O' is the nearest position the user can stand and make the hand within the camera's horizontal field of view in all possible hand movements. In fact, the sweet spot is the area where the center of the circle can locate while the circle is within the zone defined by OA and OB . Therefore, the sweet spot, shown by the shaded area, is a zone defined by $O'A'$ and $O'B'$, with OA and $O'A'$, OB and $O'B'$ parallel. After the farthest horizontal distance from the hand to the body is determined, the OO' can be calculated. Of course, the nearest distance to the camera also subjected to the depth sensor's range and in the case of the Kinect v1 the minimum range is 0.8m. The far end of the sweet spot is limited by the maximum range of the depth sensor, which is 4m. The nearest distance to the camera is also constrained by the vertical field of view, shown in the vertical view in Figure 3.5. To show up the upper body in the camera, the user cannot stand too near. The human body length l in the plot should indicate the height of hand when the arm lifts up of the head. Suppose la is the length of the body above the horizontal line from the camera center and its value

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

depends on the relative height of the camera to the user. With the angle of vertical field of view and la , the nearest distance to the camera can be determined in the case of vertical view. With all the constraints mentioned above, including the field of view angle, depth sensor range, and the height of the camera, the sweet spot area can be finally determined.

Phone

The user starts a gesture with the smartphone by pressing the screen with finger and then performs the gesture holding the smartphone and maintaining the screen touched (“push-to-gesture” [58, 43]). As the gesture finishes, the finger lifts from the screen, as shown in Figure 3.4b. So the delimiters of gesture are the status changes in term of whether the screen is touched by the finger. The application of the smartphone makes use of the touch interface to determine the boundaries of the gesture and records gesture movement data from the embedded accelerometer between the two delimiters, i.e. while the screen is being touched. Since the smartphone is graspable, this design is supposed to make the users feel natural because they do grasp the smartphone when performing gestures. In addition, the finger pressing and releasing gives users arbitrary control over when to start and terminate a gesture as well as clear tactile feedback from the finger about the gesture input procedure.

Watch

The smartwatch records gesture movements with the embedded accelerometer. As a wearable device it mainly has buttons, the touch screen (not all have it) and accelerometer itself as input. Instead of clicking buttons or touching the screen, which would involve the help of another hand, a preserved wrist rotation gesture is used to mark the start of the gesture, and to terminate the gesture the hand keeps static after the gesture movement, as shown in Figure 3.4c. Few previous works talk about gesture input approach using the smartwatch. Compared with the technique in [108] to segment gestures, which relies on keeping hand stationary between gestures to delimit the gesture boundaries, the preserved wrist rotation gesture proposed in this work can better avoid false positives of gesture input detection. Here the similar strategy adopted in Kinect gesture input, i.e. setting acceleration threshold to start gesture input, is not used mainly because the accelerometer measurement by the smartwatch includes gravity so that it would be difficult to setup a suitable threshold and the calculation of acceleration module for each sample in real time can be computationally heavy for the smartwatch.

The time series of gesturing with the smartwatch is explained in Figure 3.6. From the point of view of users, they turn the wrist to start a gesture input and the smartwatch vibrates after it detects the wrist rotation and then they immediately turns back the wrist and starts the gesture performance. Once the gesture finishes, the hand keeps static and the smartwatch vibrates again.

The part of application that manages this procedure keeps detecting the wrist rotation once it starts by checking the sign of the z value of the acceleration vector. The z axis of the

Figure 3.6: The time series of performing gestures with the smartwatch

accelerometer points out of the watch screen and when the watch body is horizontal and faces up, the z value is measured negative with the absolute value of the gravitational acceleration. So when the watch faces down, the z value is positive and the application considers the wrist has “rotated” and starts gesture collection. This detection is very computationally efficiency and this gesture delimiter is assumed to avoid false positives since people normally wears the watch facing up while they lift hand (and hence while they prepare to do gestures) and rarely place the watch facing down. Note that it is not necessary to turn the wrist so that the watch faces down parallel to the ground to trigger the delimiter. Turning a little more downward when the watch is vertical already makes the z value positive. After the wrist rotation is detected, the watch application starts collecting data from accelerometer and sending them to the smartphone via Bluetooth for gesture recognition. At the same time, it keeps on checking whether the hand is static by calculating the mean and standard deviation of the recent acceleration data in a temporal window and if these values are below a threshold, the application believes the hand is static and stops collecting and sending data. At this time, the watch vibrates again to notify the end of gesture recording. Then before the application checks the wrist rotation again, there is a temporal gap to exclude the case that the hand ends up with the watch facing down, which may causes false positives.

Taking a close look at the acceleration data the watch collects, it could be found in Figure 3.6 that the vibration, although necessary as feedback, is also measured by the accelerometer and distorts the gesture data. Before the watch body really vibrates there is a short segment of recorded data due to the delay for the vibration command to become effective. Apart from the vibration, the recover action of the wrist is also recorded but it is not

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

the actual interested gesture data. A fixed length of data from the beginning is discarded on the purpose of removing the vibration data. The length was determined in an experimental way. Since it is fixed, it is possible that the data left for recognition contains a small part of vibration data or has part of actual gesture data removed.

Control and feedback in training and interaction

The gesture input procedure described above is the basic module based on which the training and gesture control functions are built by adding appropriate control components and feedbacks of workflow. The workflow of training and gesture-based control is shown in Figure 3.3. User studies in [43, 62] both indicate the necessity of supporting immediate undo and restart functions during gestural interaction since unwanted input could occur. The undo and restart operations are triggered by buttons in this work. The smartphone has touch-based buttons and the smartwatch has physical buttons. However, the users interacts with the Kinect via freehand gestures and clicking a traditional GUI button with mouse would disturb the interaction flow. To allow the users to control the workflow with freehand without leaving their position, a concept of virtual button was developed. The virtual button is a button near the user in the 3D world within the arm's reach, and it follows the user locating itself in a fixed position relative to the user. As one of the interface component, it is visualized on the screen like the user's silhouette with a transformation between 3D and 2D. To touch it, the user just moves the hand of the silhouette onto the visualized virtual button. Note that since the virtual button follows the user, it is always in reach. If the button is docked in a fixed position on the screen, the user would have to change her position to reach the button. The application checks whether the 2D position of the silhouette hand on the screen is within the area of visualized virtual button. If yes, then the button is touched and changes its color. Once the hand leaves the button area, the command associated with the button is triggered and the button recovers its original color, see Figure 3.7. So the virtual button inherits the mental model of the traditional buttons (same visual feedback properties and functional metaphor) and is a 3D extension operated by freehand movement. The virtual button developed in this work is similar to the Kinect buttons supported by the Kinect SDK, but it does not need a push gesture (the Kinect tile button) or hovering on the button for 2 seconds (the Kinect hover button) to trigger.

With respect to feedback, the applications on the three technologies all utilize text, icons and images to show information. The progress of training and gesture command composition is indicated through a progress bar. When the users compose gesture command, to notify whether there is grammar violation, the Kinect application shows the status by a semaphore UI gadget (it makes use of the color and blinking of lights to represent the states of grammar checking result and hand tracking, and also provides feedback of rollback and cancel operations, see Figure 3.8) while the smartphone and the smartwatch applications use text messages and vibrations with different patterns indicating different statuses.

Due to the highly constrained resource in the smartwatch, the application on it collaborates with an application on the smartphone and they communicate via Bluetooth. Most

CONTROL AND FEEDBACK IN TRAINING AND INTERACTION

Figure 3.7: The virtual button to redo one training repetition

Figure 3.8: Finite state machine where the semaphore shows the status of gesture sequence composition

of the functions and UI components such as the buttons and the text notifications are duplicated in the applications for the smartwatch and the smartphone. Users can control via either side and the two interfaces keep synchronized.

Figure 3.9, 3.10 and 3.11 illustrate the user interface of the applications based on the

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

		Kinect	Smartphone	Smartwatch
Input	Gesture	Free-hand	Graspable	Wearable
	Intra-task control e.g. undo	Virtual buttons	Touchscreen button	Physical buttons (or touchscreen buttons on phone)
	Others	Traditional GUI components	Touchscreen input controls	-
Feedback channels		Visual (silhouette, video for sweet spot, dots on hand, semaphore)	Haptic (vibration) and visual	Haptic (vibration) and visual

Table 3.1: Input and feedback

Kinect, the smartphone and the smartwatch, respectively. The main functioning modules, namely the training and the control modules, are shown and the interaction workflow of the control module is illustrated with a “small w - backward” gesture command, which will retrieve and display weather forecast information in the browser or the watch face. The auxiliary functions such as account management (creation and switch of user account), gesture recognition test, configuration of confirmation step (enabling or disabling the confirmation step, which asks the users for the permission to execute the gesture command after they have performed the gesture chain correctly), and localization options (automatic, manual or default localization) are not shown in detail. On one hand, the interaction workflow is consistent over the three applications and there are some common user interface components: the complex of training configuration, the progress bar, and the rollback option. On the other hand, depending on the interaction manner and the resources available of the three applications, some input modes and feedback channels vary among the three and are summarized in Table 3.1. The vibration feedback of the smartphone indicates whether there is a grammar error while the different vibration patterns of the smartwatch are feedback for grammar errors as well as the start and end of gesture input. As shown in Figure 3.11, the interface of the smartwatch and the smartphone are synchronized and complementary.

3.4 Conclusion

This Chapter begins with a discussion on the design space of the gesture-based interaction system aiming at controlling resources in smart spaces; its possible constraints are considered. Then the interface design is described through the instrumental interaction model featured with the proposed gesture grammar to specify control commands. Then, given the abstract interface design and design requirements, the design implementation is summarized with the system architecture, which has a client-server structure, and the implementation of the gesture sequence analyzer is introduced. Backed by the server, the user-end workflow of

Figure 3.9: Illustration of the interface of the Kinect-based application

gesture-based interaction to control resources in a smart space and the training procedure are described. The end-user interface is partially inspired by the feedbacks in a previous user study and the gesture input procedure for the three technologies, the control components and feedback in training and gesture control units are discussed.

The constraints in the design space were considered and shaped the interface design. The sweet spot component in the Kinect interface is calm and gives users feedback of the sensor's range. The silhouette and dots on the hand informs the users about the status of the body and hand tracking of the Kinect sensor. The design of the wrist rotation gesture

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

Figure 3.10: Illustration of the interface of the Phone-based application

as well as the data transmission to the smartphone for gesture recognition is due to the limited input modes and computing power of the smartwatch. The interface is designed to be user-adapted. The fact that the gesture delimiters are designed for gesture segmentation can also be seen as a limitation of sensors since they cannot directly detect the beginning and the end of gestures. The delimiters enable the users to control when a gesture starts and how long the gesture is. The control components of workflow, such as undo and restart buttons, especially the virtual button for the Kinect application are also user-adapted. The interaction applications provide rich feedback about the interaction status with the designs of

Figure 3.11: Illustration of the interface of the Watch-based application

the sweet spot, the silhouette and the dot on hand, the semaphore of the Kinect application, and vibrations of the smartphone and the smartwatch applications. On purpose of reducing the physical demand of the training, one of the tasks of the Chapter 4 is to examine how many training repetitions are necessary for a decent gesture recognition accuracy.

It is supposed that the gesture performance with the smartphone should be more natural than the other two because the finger pressing and releasing is both delimiter and feedback of low delay (the finger has tactile feeling), and the touchscreen interface technologically allows this gesture boundary mark, which is very precise and of little delay. On the other hand, complicated computer vision operations in the case of the Kinect-based application

CHAPTER 3. GENIE: A GESTURE-ENABLED INTERACTION METHOD FOR SMART ENVIRONMENTS

introduce perceptible delay between gesture performance status and visual feedback in the current deployment and the users have to adapt to the velocity threshold to start the gesture input. For the smartwatch-based gesture input, the extra wrist rotation and the vibration feedback delay introduced by the detection of the static hand at the end of gesture might also deteriorate user experience.

There are three important concepts in the interaction: smart resources, the gesture vocabulary and the commands associated to the gestures to control the resources. From the point of view of implementation, the client-server structure allows the easy scaling of the smart resources integrated in the interaction system. The gesture recognition approach is learning-based and enables to personalize the gesture vocabulary. Then with the help of database, the association between gestures and commands is flexible. Therefore, the designed interaction system supports the delivery of personalized gesture vocabularies and the easy deployment on a new smart space. Since the grammar-based interaction is location-aware, when the users change the room, the control keeps effective but targets to the resources where users are located. Furthermore, the interaction might also be portable to another smart space given a seamless integration between the server in that smart space (or in the cloud) and the user-end application, including the reuse of existing gestures and the configuration of the gesture command meanings. This will be a valuable extension to the work in the future.

The design of the system also has limitations. Since the initial letters or acronyms are used to represent the resources, the gesture vocabulary for resources is restrained in alphabet and it is probable to encounter resources with the same initial letter or acronym. Another limitation is the control granularity. On one hand, the gesture can only specify simple parameters with few possible values like the color of the light (in fact, the gesture sequence commands without gesture for parameters include default parameters, e.g. the parameter of the pulling-down-blinds command is extending the blinds “100%”). It would be very difficult to name with gestures a complicated parameter (such as a file name among many to open). On the other hand, if multiple resources of the same type are to be controlled differently in the same room, the room-level location context is not enough to distinguish these devices.

Now that the interaction system is designed and implemented, Chapter 5 comes up with several research questions and evaluates the design with a user study. Before that, Chapter 4 discusses the design and off-line evaluation of gesture recognition and gesture-based identification approaches.

Chapter 4

A strategy for gesture-based recognition and identification

Generally speaking, the target gestures to recognize to implement the interaction method explored in Chapter 3 are 3D in-air motion hand gestures. To make a computer program understand a gesture, the gesture has to be sampled and digitized by tracking systems based on certain kind of sensors. The motion can be tracked explicitly with the position and orientation from optical tracking, or implicitly with the acceleration and angular speed by inertial sensing. Then a gesture recognition approach interprets the captured data, which is a time series, and determines if they correspond to a predefined gesture or to an unrecognized one. Alternatively in the case of user identification, the captured gesture data as a biometric input is used to distinguish the active performer from a set of users.

4.1 Challenges and design objectives

The procedure of gesture recognition poses different challenges. First of all, unlike speech, gestures lack a standardized vocabulary, although there are socially accepted gestures to code different messages. Moreover when performing a gesture, there are variations not only between different users but also between the different repetitions of the same individual. The variations include the speed of the motion, the shape of the trajectory, the order of the strokes of letters, and the orientation of the hand, to name a few. Additionally, the gesture data captured by optical tracking may vary when users perform gestures in different positions and/or with different orientations relative to the camera even though the gesture performances are exactly the same. In addition, the hand motions are continuously recorded by the tracking systems and the meaningful gestures reside in the data stream. To recognize the gestures, the starting and ending points of a gesture should be found. This process is usually referred as “gesture segmentation”.

As previously explained, gesture recognition can be used in application scenarios such as gesture character input [31], medical diagnosis [9] and command-control. The applica-

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

tion context imposes additional requirements on the gesture recognition approach. In this work, the goal to develop a sound and practical gesture-based interaction system to control resources in smart space imposes the following requirements to the gesture recognition approach:

- High recognition rate
- Real-time
- Support of personalized gesture vocabularies
- Need for few training repetitions
- Adequate to work on data coming from different sensors and workable in multiple platforms
- Easy to implement

Enabling personalized gesture vocabularies makes the system extensible and adaptive to users' preferences. The requirement of few training repetitions is to keep a low physical demand and enhance the acceptability of the system. In addition, to serve the research purpose and make the interaction system adaptive to diverse application scenarios, the recognition approach should be able to work with different types of data (position and acceleration acquired by inertial and vision-based devices, respectively in this case) and run in different platforms (e.g. the desktop and the smartphone). To save development cost, one unique solution is preferred. Finally, it is desirable if the approach is easy to implement and light to run, especially in resource limited devices.

Complementary to gesture recognition, the use of gestures as biometric inputs for user identification have a similar working procedure. After a gesture sample is captured, rather than recognizing what gesture a user has performed, the gesture-based identification approach tries to figure out who has done it. The gesture-based identification problem presents similar challenges and design objectives as that of gesture recognition, except that the gesture performance variation between users is what the identification would make use of. As far as this work concerned, the gesture-based identification approach is investigated based only on the Kinect, since the smartphone and the smartwatch are normally personal devices and there is no need to identify users. The Kinect is normally not a personal device, however.

In this work, a Dynamic Time Warping (DTW)-based strategy is employed to implement the gesture recognizer. First of all, it is learning-based and can be trained with the data from different vocabularies for the recognition (opposite to a rule-based recognition approach, which defines fixed rules to detect gestures and the rules are highly associated with a particular gesture vocabulary), and the gesture vocabulary would be easy to configure and personalize. Then the authors in [54] argue that the template-based approaches - the DTW is one of them - perform well even with few training samples and they are easier to implement and deploy on different types of devices since they do not require specialized libraries.

A CLASSIFICATION ALGORITHM TO ENABLE GESTURE RECOGNITION AND IDENTIFICATION

A user study carried out in [43] shows that different users usually prefer different gestures for the same task. So the recognition approach is designed to work in user-dependent mode to support a personalized vocabulary. In the review of Dynamic Time Warping (DTW) algorithm in [20], the authors comment it to be useful for personalized gesture recognition in user-dependent applications, where a large set of training samples is hard to collect. In the user-independent case (see [100] for example) where large intra-class gesture variations exist, DTW algorithm might have much increased computational load due to larger amount of templates to compare and statistical approaches that explicitly model the variability would be necessary. In the off-line evaluation of this Chapter, the performance of the approach including recognition accuracy, time complexity, dependence of the recognition accuracy on the number of training repetitions will be examined with both position and acceleration data. Based on the same DTW classifier, a gesture-based identifier is developed and its accuracy and time complexity is examined in an off-line test as well.

4.2 A classification algorithm to enable gesture recognition and identification

4.2.1 Similarity measurement based on Dynamic Time Warping algorithm

The Dynamic Time Warping (DTW) algorithm developed in this work is based on the approaches introduced in [86] and [72]. DTW is a dynamic programming (DP) based time-normalization algorithm, which is popular in spoken word recognition field. It was introduced in [86] to deal with the highly complicated nonlinear fluctuation in a speech pattern's time axis. The algorithm is to solve an optimization problem, which tries to minimize the time-normalized distance between two pattern series. The minimized distance is a measurement of similarity between them. Then given a limited set of templates, a testing pattern can be recognized by calculating the time-normalized distances between itself and each of the templates and selecting the template which is “nearest” to the testing pattern as the recognition result.

Let us suppose that we have two sequences of feature vectors, denoted as:

$$A = \vec{a}_1, \vec{a}_2, \dots, \vec{a}_i, \dots, \vec{a}_I \quad (4.1)$$

$$B = \vec{b}_1, \vec{b}_2, \dots, \vec{b}_j, \dots, \vec{b}_J \quad (4.2)$$

A warping function F is defined as a sequence of points that indicates the mapping relationship between sequence A and B

$$F = c(1), c(2), \dots, c(k), \dots, c(K) \quad (4.3)$$

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

where $c(k) = (i(k), j(k))$, $1 \leq k \leq K$, $1 \leq i(k) \leq I$, $1 \leq j(k) \leq J$. The distance, or the mapping cost between $\vec{a}_i(k)$ and $\vec{b}_j(k)$ is:

$$d(c(k)) = d(i(k), j(k)) = d(\vec{a}_i(k), \vec{b}_j(k)) \quad (4.4)$$

Then the mapping cost of two sequences is defined as the expected value of the summation of distances between the mapping pairs determined by a warping function F :

$$E(F) = \sum_{k=1}^K d(c(k)) \cdot w(k) \quad (4.5)$$

where $w(k)$ is the weighting coefficient for the k -th mapping pair. $E(F)$ is a function of F and provides means of measuring the goodness of the warping function. The optimal time-normalized distance between pattern sequences A and B is defined as

$$D(A, B) = \min_F \frac{\sum_{k=1}^K (d(c(k)) \cdot w(k))}{\sum_{k=1}^K w(k)} \quad (4.6)$$

It is the best matching between arbitrary sequences A and B in terms of matching cost. The minimization is obtained through a dynamic programming algorithm. For a concrete implementation, we have to specify the definitions of distance $d(c(k))$ and weighting function $w(k)$. The authors in [86] and [72] also propose some restrictions on the warping function F taking into consideration the characteristics of feature sequences, such as monotonic conditions [86], continuity conditions [86], boundary conditions [86], adjustment window conditions [86] and local continuity conditions [72].

4.2.2 Recognition and identification approach with the similarity measuring engine

This subsection gives an overview of the gesture recognition and user identification approaches built on top of the DTW's capability of measuring the similarity of two gesture data series.

The general architecture of the gesture recognition module is designed as a layered structure, as depicted in Figure 4.1.

Raw gesture movement data are collected from sensors that are, in this Chapter, those embedded in the Kinect and the smartphone devices. The Kinect tracks the 3D position of the center of user's palm relative to the camera, while the accelerometer of the smartphone senses the phone's acceleration relative to its body frame with three axes.

Before passing the movement data to the classifier, a pre-processing procedure is applied on it, including data refinement, feature extraction and down sampling, for recognition accuracy improvement, platform difference isolation and computation efficiency enhancement, respectively. These stages are following described.

RECOGNITION AND IDENTIFICATION APPROACH WITH THE SIMILARITY MEASURING ENGINE

Figure 4.1: Architecture of the gesture recognition module

Data refinement. Silence parts may appear at the ends of the raw gesture signal, corresponding to the moments when the users are preparing to perform the gesture or waiting for the termination of the recording (i.e. the gesture making hand is approximately static). The data refinement step in Figure 4.1 is focused on removing the silence parts at the end of the signal; in the work, this is done by setting a threshold to the module of the difference between two adjacent sampling moments.

Feature extraction. This step extracts the features of the movement signal which can be employed by the classifier to distinguish one gesture from another. In this work, the original sensed data types, i.e. the position data sampled by the Kinect and acceleration data of the mobile phone, are sent to the classifier because they reflect the kinematic features of movements. For Kinect data, to make the trajectory of the hand independent of the local coordinate system based on the camera, the coordinate system is translated to the one with the origin as the center of the volume that contains the trajectory of the gesture movement. As for the acceleration data, no transformation was utilized as it is already based on a relative coordinate system (body frame of the phone). Furthermore, the acceleration signal is very noisy and transformation may introduce significant deviation.

Down sampling. To make the computation more efficient, the data sequence is shortened by down sampling. It is applied on raw data processed after the operations of data refinement and feature extraction, from both the phone and the Kinect. The data are passed through a Butterworth low pass filter and then length normalization is applied. The parameters of

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

cutoff frequency of the low pass filter are different due to the different sampling frequencies in the smartphone and the Kinect, which were empirically set. The low pass filtering is to guarantee that the Shannon-Nyquist sampling theorem is maintained during the length normalization step. The reduced sampling rate after this low pass filtering should be smaller than the sampling rate when the signal is reduced to the normalized length (20 in this paper). The raw data from the phone without low pass filtering (but with length normalization step, the normalized length is similar to the original length) were tested and the result was similar to the one with low pass filtering and length normalization to 20, but the computing time was significantly longer.

Suppose that the original sequence is $R(n), n = 1, 2, \dots, N$. Then the normalized sequence is given as:

$$\tilde{R}(\tilde{n}) = (1 - s) \cdot R(n) + s \cdot (R(n + 1)) \quad (4.7)$$

where $\tilde{n} = 1, 2, \dots, \tilde{N}$,

$$n = \lfloor (\tilde{n} - 1) \cdot \frac{N - 1}{\tilde{N} - 1} + 1 \rfloor \quad (4.8)$$

and:

$$s = (\tilde{n} - 1) \cdot \frac{N - 1}{\tilde{N} - 1} + 1 - n \quad (4.9)$$

This normalization linearly interpolates or compresses the length of the original feature sequence into a fixed length $\tilde{N}$. For down sampling, $\tilde{N} < N$.

In summary, the preprocessing is highly dependent on the properties of platforms, such as data collection procedure, data type, sampling rate. This step deals with the diversity of distinct platforms to make the classifier transparent to the difference of platforms, and tries to feed the classifier with relatively uniform data targeted at decent classification performance in terms of accuracy and computational complexity.

The recognition system creates templates in the training stage. Their creation has great importance to the recognition accuracy so that the templates should be “standard” enough. In this work, the minimum selection [48] is utilized for templates creation. The criterion of this approach is the intra-class DTW distance, which is the sum of DTW distances between one certain training sample and all the other samples within the same class. The sample with the minimum intraclass DTW distance is selected as the template of this class. Compared with other template selection approaches like normal selection, average selection and multiple selection described in [48], the minimum selection is the computationally lightest. The gesture recognition results in [37] show similar accuracies using acceleration data with minimum selection and with “min-max” selection, but the “min-max” selection has more complexity.

Once the gesture templates are prepared, a gesture sample is recognized by calculating its DTW distance to all the templates and the gesture whose template is the most similar to the gesture sample is the recognition result.

The proposed gesture-based identification approach (only on Kinect data) shares the same data preprocessing and training procedure as that of gesture recognition approach. It

identifies users by verifying whose gesture template is most similar to the testing pattern via the calculation of DTW distances.

4.3 Off-line evaluation of the gesture recognition approach

4.3.1 Testing scenario

In order to test the gesture recognition approach, we collected one dataset based on the Kinect from 11 right handed users, who repeated 10 times each gesture in the vocabulary in Table 4.1, and another dataset based on mobile phone from 11 right-handed users repeating 5 times per gesture on the same vocabulary. Each user completed the data collection in the same day he/she started to do it.

Gesture commands in the form of the grammar proposed in Chapter 3.2 contain letters that specify smart resources and geometric gestures that represent operations. Possible gestures that can compose gesture commands in this format are thought out in this vocabulary considering a smart space environment. For instance, the letters are the initials or acronyms of smart resources, ‘d’ for door, ‘w’ for window, ‘p’ for phone, ‘tv’ for television, ‘l’ for light, ‘ac’ for air conditioning, ‘r’ for robot, ‘f’ for photo frame, ‘m’ for music and ‘s’ for scale. The geometric gestures can represent operations like adjusting volume (‘up’ and ‘down’), switching on devices (‘circle’) and transferring messages (‘spiral’). Since it is not clear how well the recognition approach works on small letters and capital letters and which one users prefer, both of them are included in this off-line test. Theoretically, users can use either small or capital letters in the interaction since both of them are syntactically allowed in the grammar (they might have different semantic meanings depending on the implementation of the system, though) and they can be recognized after proper training. The directional/geometric gestures are illustrated in Figure 4.2 while drawings of letters are not shown since there are variations of drawing ways between users.

The Kinect and the smartphone are selected for the test since they represent optical and accelerometer based tracking techniques. As the smartwatch has a similar accelerometer to that of the smartphone (sensed acceleration including the force of gravity, device body frame-based coordinate system) although the body parts of motion tracking (the wrist and the hand) and sampling rates are different, it is assumed that the performance of the gesture recognition approach with smartwatch data would be similar to that with smartphone data. The smartwatch is not included in this offline test although it is tested as an enabling technology in a user study in Chapter 5.

Data collection

The application to collect Kinect (v1) data runs on a desktop computer connected with the Kinect and it was programmed in OpenNI in C++. The graphic user interface of the

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

Table 4.1: A primitive vocabulary for experiments

Action primitives	Individuation primitives	
Directional/geometric	Small letters	Capital letters
Up	d	D
Down	w	W
Left	tv	TV
Right	p	P
Forward	l	L
Backward	ac	AC
Clockwise circle	r	R
Counterclockwise circle	f	F
Spiral to the left	m	M
Spiral to the right	s	S

Figure 4.2: Illustration of directional/geometric gestures

program informs the user of when the Kinect starts to record the movements and then visible signs are shown to count down the time left to the end of the performance, so that the users can synchronize the gesture movement with the time slots of data collection. The application can track and record the 3D position of the user's palm center relative to the camera, no matter what the posture of the writing hand in the air is (e.g. with an opened palm or with

a pointing posture) and the obtained data has a sampling rate of about 30 Hz.

To collect data in the mobile phone, an Android application was developed. The chosen device was a Google Nexus 4, whose accelerometer's sampling rate is around 200 Hz. The sensed accelerations are based on a 3D coordination system relative to the device. In our first implementation, users trigger the data collection procedure by pressing a button on the screen and then a vibration launches to inform them of the beginning of the performance of the gesture. When they finish the movement their hands keep static until the second vibration comes. To unify the manner of holding the phone and at the same time to make the manner as natural as possible, the subjects were required to hold the phone as if they wrote on blackboard with a chalk and the upper-left corner of the phone is the counterpart of the tip of the chalk. The subjects were asked to restrict the movements of the body of the phone to translations, i.e. theoretically no rotations are allowed. These requirements simplify the time series and the processing of signal for the classification.

Waveforms of the gesture 'w' of one user captured by the Kinect and the Phone are shown in Figure 4.3a and 4.3b, respectively. The raw data from the data collection software of the Kinect is based on the coordinate system of camera. However, to present the signal better, the raw data is translated in Figure 4.3a so that the origin is the first sample of the signal. The trajectory of the gesture in 3D space is illustrated as well. Figure 4.3b shows the raw signal from the Phone, which is relative to the Phone's body frame-based coordinate system. With the sampling rates known, we can see the duration of the Kinect signal is about 3s and the one of the Phone signal is almost 2s. Readers may also have noticed that there is large part of non-gesture status at the beginning and the end of the Kinect gesture since the position of the hand is almost static.

Finally a dataset of the gestures in Table 4.1 was collected from 11 users with each gesture repeated 10 and 5 times from the Kinect and the smartphone, respectively. This dataset can be reused in the future to test other recognition approaches.

Evaluation approach

The developed recognition program was written in Matlab running on a desktop with Intel 2 Quad CPU (2.5 GHz, 2.5 GHz) and 4 GB memory. The evaluation approach is Leave-One-Out Cross Validation (LOOCV). The idea is to split the data into two parts: one part contains the samples for estimating the performance of the algorithm and the remaining part is used for training the algorithm. Suppose that there are N samples labeled with one certain gesture. For the Leave-One-Out Cross Validation (LOOCV), the data are split so that there is only one sample in the validation part and the remaining $N - 1$ samples are to train the algorithm. The data are split $N - 1$ more times so that every sample is selected into the validation part. The selection of gesture templates uses the minimum selection approach mentioned in [48].

In this work, the gesture data were recognized in a user-dependent manner and the templates of one certain user were generated from his/her own gesture samples. Generally speaking, user-dependent system has higher recognition accuracy than user-independent sys-

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

(a) Signal from the Kinect

(b) Signal from the Phone

Figure 4.3: Captured signal of the gesture ‘w’

tem and is easier to implement because to recognize gestures independent of user, the system usually needs much larger gesture database to adapt the diversity of users’ behavior but the database may not include all the templates or models of different users, resulting to degradation of accuracy. Accordingly, the computational cost may also increase due to the searching or matching in a larger database. In addition, the user dependent system allows users to personalize the vocabulary so that the user can interact with the system with his/her fa-

vorite way and its degree of independence on the predefined vocabulary is larger than the user-independent manner.

Moreover, we assume that before recognizing the gesture, the system prioritizes gesture search on a particular gesture set: directional/geometric actions, small letters or a capital letters. We can assume this because the gesture type of each part of the gesture command series is determined according to the gesture syntax.

Within the framework of the user-dependent manner and the assumption of already known gesture type, the construction of validation set and gesture template set is now formally described. Suppose that the m -th sample of each gesture of each user is selected as validation data ($1 \leq m \leq N$, $N = 10$ for the Kinect dataset and $N = 5$ for mobile phone dataset). Denote $gest_{u,i,j}$ as j -th sample of the i -th gesture for the u -th user and in our case $1 \leq j \leq N$, $1 \leq i \leq 10$ and $1 \leq u \leq 11$. The maximum of i is 10 because we test the three groups of gesture vocabulary independently and each group contains 10 movements. Let set V be the validation data set, and T the gesture template set. Here is how to calculate V and T when the m -th sample data are selected as validation data:

1. Set $V \leftarrow \emptyset, T \leftarrow \emptyset$
2. For $u \leftarrow 1$ to 11
3. For $i \leftarrow 1$ to 10
4. Set $V \leftarrow V \cup \{gest_{u,i,m}\}, T \leftarrow T \cup \{gest_{u,i,x}\}$, where

$$x = \arg_k \min_{\substack{1 \leq k \leq N, \\ k \neq m}} \sum_{\substack{1 \leq n \leq N, \\ n \neq m, k}} D(gest_{u,i,k}, gest_{u,i,n})$$

5. End for i
6. End for u

As a result, the validation set V contains all the testing samples of gestures of all the users and set T has the templates of all gestures of all users.

Let $v_{u,i}$ and $t_{u,i}$ be the testing sample in V and template sample in T respectively of the u -th user's i -th gesture. The sample $v_{u,i}$ is recognized as the G -th gesture by comparing it with all the templates of the u -th user,

$$G = \arg_{i'} \min_{1 \leq i' \leq 10} D(v_{u,i}, t_{u,i'}) \quad (4.10)$$

Change the value of m and repeat another $N - 1$ times the above procedure. The average of all the results in these N times is the final gesture recognition accuracy.

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

Implementation of the dynamic time warping algorithm

In the concrete implementation of the algorithm, Eq. (4.4) is calculated with Euclidean distance. Denote the weighting coefficient as

$$W_c = \sum_{k=1}^K w(k) \quad (4.11)$$

If it is considered independent of the warping function F , as in [86], Eq. (4.6) is simplified as

$$D(A, B) = \frac{1}{W_c} \min_F \left[\sum_{k=1}^K d(c(k)) w(k) \right] \quad (4.12)$$

In the test, we utilized weighting function in asymmetric form mentioned in [86]

$$w(k) = i(k) - i(k-1) \quad (4.13)$$

Assume that $i(0) = 0$. Then $W_c = I$ according to Eq. (4.11) and I is the length of signal A which is defined by Eq. (4.1)

Eq. (4.12) is solved by a dynamic programming algorithm with the following initial condition

$$g_1(c(1)) = d(c(1)) w(1) \quad (4.14)$$

And its recursion equation

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k)) w(k)] \quad (4.15)$$

The final result is

$$D(A, B) = \frac{1}{W_c} g_K(c(K)) \quad (4.16)$$

We implicitly assume that $c(0) = (0, 0)$. According to the definition of the weighting coefficient in Eq. (4.13) for the asymmetric form the initial value is $w(1) = 1$.

Type I local continuity constraints [72], as illustrated in Figure 4.4 are used to further specify the calculation of $g_k(c(k))$ in Eq. (4.15) which now can be rewritten as Eq. (4.17). These constraints restrict the slope of the warping path between 0.5 and 2.

$$g(i, j) = \min \left[\begin{array}{l} g(i-1, j-2) + \frac{(d(i, j-1) + d(i, j))}{2} \\ g(i-1, j-1) + d(i, j) \\ g(i-2, j-1) + d(i-1, j) + d(i, j) \end{array} \right] \quad (4.17)$$

In addition, the normalized length of sequences is empirically set as 20 and the size of the adjustment window, which restricts the search space [86], is 6.

Figure 4.4: Illustration of the Type I local continuity constraint

To show the effectiveness of the approach, the plot of Figure 4.5a illustrates frequency-normalized histogram of all possible intra-class DTW distances of the samples of ‘D’ and ‘P’, and the histogram of all possible inter-class DTW distances between the samples of these two movements, using one user’s Kinect gesture data. For clearly showing the range of the histograms, the intra-class distances of ‘P’ and the frequencies of inter-class distances are set minus. If we call $D(A, B)$ the distance from A to B , then the left histogram of inter-class distance shows the statistics of sample distances from ‘P’ to ‘D’ and the right histogram describes the statistics of distances from ‘D’ to ‘P’. Note that because we used an asymmetric weighting function in the DTW algorithm, the two inter-classes histograms are not symmetric around the vertical axis. Most of the intra-distances and inter-distances are distributed in different parts of the distance axis, but small parts of the histograms of intra-class distances and inter-class distances are intersected, which means that ‘D’ and ‘P’ may be confused if we classify them by calculating all DTW distances between the testing sample with all samples from different classes and selecting the class which has the sample nearest to the testing sample in terms of DTW distances.

Thanks to the template choice with the minimum selection approach, the discrimination of the histograms is improved in the plot of Figure 4.5b, which shows the distribution of distances by simulating the LOOCV procedure introduced above, i.e. calculating the distance between the testing sample and the templates chosen by the minimum selection approach. The left histogram of inter-class distance is the statistics of distances between testing samples of ‘P’ to the templates of ‘D’. Intuitively, the minimum selection approach selects as template the sample which is nearest to the center of the cluster formed by the samples in the same class. Comparing with Figure 4.5a, the histograms are more concentrated. Furthermore, the minimum value of inter-class distance is larger than the one in Figure 4.5a, and the maximum values of intra-class distances are smaller, which is beneficial to the separation of the histograms and the accuracy of classification. All the histograms are separated in this plot.

4.3.2 Results

Results show how the proposed algorithm behaves both for the Kinect and smartphone datasets, in terms of accuracy and computation time. Additionally, it is obvious that the

Figure 4.5: Frequency-normalized histograms of intra- and inter- class DTW distances

processes of a) training the system personally before interaction and b) identifying the user before starting the interaction processes are two aspects that should be as easy as possible for the user, or even disappear, if feasible. For this reason, results also show how to optimize the training stage and explore how the system can adapt to new users without explicit training.

Recognition accuracy

Because we tested the recognition approach in user dependent manner, let us first have a look at the accuracy with respect to users in the two platforms, as shown in Figure 4.6. In

practice, the gesture vocabulary has three independent groups: directional movements, small letters and capital letters, since according to our interaction grammar we can assume that the system already knows to which group the gesture movement belongs before recognizing it. The results were obtained by, for each user, averaging the recognition accuracies of the movements in each group respectively. Then based on the results in Figure 4.6, we calculated the mean and standard deviation of the accuracies of each gesture group among all the users, and compared them based on the two platforms in Figure 4.7. The average accuracies for the Kinect are {99.18%, 98.91%, 97.82%} and for the Phone {94.18%, 95.27%, 91.45%}, with the order {directional movements, small letters, capital letters}. The standard deviations are indicated by the half length of the error bar on the top of the bars.

Another dimension to analyze the accuracy is in terms of gestures. The three plots in Figure 4.8 compare the recognition accuracies based on the two platforms with respect to different gestures in the three groups of the vocabulary. The results with respect to gestures are calculated by averaging all the accuracies of gestures for each user. Most of the gestures were recognized with high accuracy in both of the two platforms and some of them were detected with accuracy 100%, including ‘down’, ‘forward’, ‘counter-clockwise circle’, ‘spiral to the left’, ‘d’, ‘l’, ‘s’, ‘L’, ‘M’ based on the Kinect, ‘spiral to the right’, ‘r’, ‘S’ based on the phone. Wrong detections recognize one gesture with another. Confusion matrices, which show the percentage of recognizing one gesture as one of all the gestures in the vocabulary, can help analyze the discrimination of gestures in the vocabulary. Readers can find the confusion matrices of the three gesture groups on the Kinect and phone platforms in Appendix A.1. We have found from these matrices that the capital letters ‘D’ and ‘P’ were confused a lot in both platforms. Before the test, we made a survey of handwriting of the capital and small letters in the gesture vocabulary to see the diversity of the forms and manners of writing. A scanned copy of the handwriting by some volunteers is displayed in Appendix A.2. The gestures of directional movements and small letters are easier to recognize than the capital letters mainly because of the existence of some confusing letters such as ‘D’ and ‘P’ in the capital letter group.

Here it is important to notice that this work does not aim at comparing recognition results on the Kinect and the phone on purpose because the test settings have more variables than platform difference. The idea is to demonstrate that the same Dynamic Time Warping algorithm could obtain high recognition accuracy both on the phone and the Kinect and that it could provide a unified recognition solution on different platforms. All test results are presented under the described testing scenarios.

Responsiveness

Since instant response to the user is a necessity in real-time application like human-machine interaction with gestures, and the recognition algorithm may be implemented in resource restricted platforms, the approach should be light in computational cost.

There are two main steps to recognize one gesture signal: pre-processing and DTW matching. The computing time for each stage is gathered in Table 4.2 under the testing

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

(a)

(b)

Figure 4.6: Recognition accuracy of users on the two platforms

scenario introduced before. The running time in a real environment is influenced by multiple factors, such as the computational power of the platform, the efficiency of the compiler, the programming language, etc. Thus direct comparison of runtime makes limited sense. However, the data listed in Table 4.2 can give a general idea about how the response time is in a real testing configuration. The computational complexity of one DTW matching in the Kinect and the phone, in terms of big O notation, should be the same because the tests in these two platforms were carried out with the same implementation of the DTW algorithm having the same parameters. The lengths of the samples were all normalized as

Figure 4.7: Mean and standard deviation of accuracy of the gesture groups on the two platforms

Table 4.2: The comparison of time cost on the Kinect and the Phone (ms)

	Pre-processing (P)	One DTW matching (M)
Kinect	0.3	5.1
Phone	1.8	5.2

20 before matching. However, the pre-processing phase in the phone consumed much more time than that of the Kinect. The main reason is that the sampled gesture data of the phone is larger than the Kinect, due to the different sampling rate: 200Hz for the accelerometer of the phone and 30Hz for the Kinect. The durations of recording gestures on the Kinect and the phone were similar so the length of phone data is about 6.7 times of the length of the Kinect data corresponding to each gesture. However, the relationship between the time consumptions of pre-processing on the Kinect and the phone is not that clear and direct, although according to the data in Table 4.2 the phone needs 6 times long of the Kinect in this phase. The main load of computation in the pre-processing step is low pass filtering which has time complexity $O(n \log(n))$, where n is the length of the signal, along with some calculation with linear complexity and the pre-processing in the Kinect has extra linear complexity calculation for adjustment of coordinate system.

Only when the DTW distances between input signal and all the templates are calculated, the recognition result comes out by selecting the nearest template. So let us assume that we already have all the pre-processed templates, and then the total computing time to recognize one gesture sample is

$$t = P + M \cdot numTemp \quad (4.18)$$

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

(a)

(b)

(c)

Figure 4.8: The recognition accuracy with respect to the three gesture groups (for. = forward, back. = backward, c. c. = clockwise circle, cc. c. = counter-clockwise circle, s. l. = spiral to the left, s. r. = spiral to the right)

where t is the total computing time, P is pre-processing time, M is the time of one DTW matching and $numTemp$ is the number of templates the input signal has to be matched with. In our test, $numTemp$ is 10. So in the testing scenario, the system needs 51.3ms to recognize one gesture sampled from the Kinect and 53.8ms to recognize one gesture from the phone.

The influence of the amount of training samples

From our data collection experience, we can say that repeating too many times each gesture to have the system trained may be tiring to the users. It would be interesting to investigate the dependency of recognition accuracy on the number of training times (the repetitions of one certain gesture). To this end, we further took a Leave-p-Out Cross Validation on the same dataset and testing scenario to see how the recognition accuracy evolves as the number of training samples change.

The plot in Figure 4.9 shows the performance of the approach in accuracy using different number of training samples. In our case, N is 10 for the Kinect and 5 for the phone. The horizontal axis is the number of training samples for each gesture and the vertical axis is the percentage of accuracy value. We still did the test in a user-dependent manner and first calculated for each user the average recognition accuracy of the three independent gesture groups on the Kinect and the phone: directional movements, small letters and capital letters. Then based on these data, we averaged all the users to get the final average accuracy and

Figure 4.9: The evolution of the mean and the standard deviation of recognition accuracy with the number of training samples (directional=directional movements, small=small letters, capital=capital letters)

corresponding standard deviation, plotted in the figure. The half-length of the error bar indicates the value of standard deviation of the accuracy. Off-line tests in [13] and [43] also checked the sensibility of their approaches to the number of training repetitions and the accuracy with 2 training samples is similar to that with 7 [13] and 10 [43] samples respectively. Our result is similar: the result with 3 training samples almost reaches the accuracy using 9 samples.

We can see from Figure 4.9 that the recognition accuracy improves slowly, although with small ripples in the phone platform and the standard deviations of the accuracy decrease as the number of training samples rises; this tendency of the results of phone data is not that obvious as the Kinect because phone gesture data has fewer repetitions.

Suppose that we have four gestures and each of them has 10 samples. With the DTW algorithm, the distance between any two samples can be calculated and we can display these samples as nodes in a 2D plane to show their relative mutual distances, as illustrated in Figure 4.10. Because the users cannot repeat one gesture movement exactly the same, the distances are not zero between these repetitions and the nodes which represent them are separated and form clusters in one area of the 2D plane. To simplify the explanation, let us further suppose that the four gestures are designed so well that the samples of every gesture form one cluster and there is no overlap between any two clusters.

When the number of samples is small, such as 1 or 2, it is quite likely that the samples in

THE INFLUENCE OF THE AMOUNT OF TRAINING SAMPLES

(a) Nodes in the rim are selected as templates

(b) Nodes in the center are selected as templates

Figure 4.10: The illustration of gesture clusters. Different shapes represent different gestures. Shaded nodes are selected as templates.

the rim of clusters are selected as templates and this case is illustrated in Figure 4.10a where the shaded nodes are selected as templates. As one unknown sample is recognized as the gesture whose template is nearest to this sample, some of the circle and diamond samples will be confused with other gestures in this graph. As the number of training samples increases, the selected templates gradually converge to the center of the cluster since we use the minimum selection approach which tries to choose the sample that has minimum intra-class distance as template. Figure 4.10b illustrates the situation when the samples in the cluster center are selected as templates, which will lead to better recognition result than the Figure 4.10a. This explains why the accuracy increases if more training samples are available. Here we should note, however, that this increasing tendency does not guarantee that more training samples always make better rate than less training samples especially

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

when small amount of samples are available, because in such cases more samples may not lead to better template selection. This is what happens in the slight drops in the evolution on the phone platform. On the other hand, the convergence process of the template selection makes the selected templates more stable (no matter how we change the subset of training samples, the cluster center sample is always selected as template unless it is used as test sample itself) in each test so that the recognition results vary less if the testing samples do not have large variety, which means the standard deviation of the accuracy decreases gradually.

User-independent test

Following the same idea of reducing the amount of training samples, it is interesting to explore whether the existing users' templates can be used to recognize the gestures of new users who have not trained the system. With the same gesture dataset we can simulate the following scenario: the system contains templates of N existing users who have trained the recognizer, and then a new user comes. Now we want to know what the recognition accuracy is if the new user performs gestures directly without training. The testing gesture is compared with the templates of all of the existing users. In our case, we have 11 subjects in total. For a certain N , the gesture samples of the $11 - N$ users are tested with templates of the N existing users. We did the test C_{11}^N times to cover all the possible combinations of N existing users.

The N varies from 1 to 10 and we get the average recognition results shown in Figure 4.11. To the best of the author's knowledge, there is in the literature no work that examines the evolution of recognition accuracy on the number of existing users in user-independent test. As what we expected, the rate increases if more users' templates are available making it more possible that the template similar to the new user's gesture exists. However, the increasingly available templates from different users also raise the risk that one gesture template of existing user is similar to a performance of another gesture of the new user. This may explain the ripple in the evolution for directional gestures on the Kinect platform.

After analyzing results on different platforms and on gestures in detail, it can be noted that the recognition results are better when tested on the Kinect than on the phone. This is reasonable due to the fact that the users hold the phone in various manners resulting to distinct acceleration signals even if they were performing the same motion, which complicates recognition. The Kinect, however, simply tracks the position of the palm, so that the recognition on the Kinect is more robust to hand postures. What is more, the gestures with uniform performing manner such as the straight movements and the capital letter 'S' tested on the Kinect have high recognition accuracy and very small standard derivation.

We can conclude based on this test that in user independent scenario, it is better to select motions with unified performing manners as elements of the vocabulary as this benefits recognition accuracy as well as time and storage efficiency due to that there's no need to maintain too many templates for one gesture. As for platforms, the Kinect outperforms the phone platform in user-independent scenario. This test also provides a mechanism to measure

Figure 4.11: The evolution of the mean and the standard deviation of recognition accuracy with the number of existing users

the degree of variation of a certain gesture between different users. The movements with lower user-independent recognition rates, in another point of view, contain more personal information and thus are suitable for user identification purpose.

Finally, with the results of this subsection and the previous one, we may propose a possibility that the users may do not need to explicitly train the system. As we have seen, for most of the gestures the average user independent test results are around 80% or above on the Kinect. And the user-dependent test with one training sample is above 90% in the worst case. Based on these two facts, the user can directly perform gestures to interact without training on purpose and the input gesture is recognized in a user-independent manner. At the same time, the input signal is stored as a training sample. When the user performs the second gesture, the system can recognize it in user-dependent manner and 90% recognition accuracy is already ensured. The risk of this idea is the possible frustration to the user in the first recognition caused by the low user-independent recognition rate. In any case, this approach will not work for a mobile phone platform due to the extremely low user-independent recognition rate.

4.4 Off-line evaluation of the gesture-based identification approach

4.4.1 Testing scenario

Gesture-based user identification in this work, as introduced in Chapter 2.3, is to use behavioral attribute (the gesture) of one person to identify her from a set of users. The dataset collected with the Kinect (only the Kinect will be evaluated in this Section since the smartphone and the smartwatch are normally personal devices) to evaluate the identification approach is the same as that of the test of the gesture recognition approach. The testing environment is the same, i.e. running in Matlab on a desktop of the same specification. Suppose V is the validation set and T the templates set in the Leave-One-Out Cross Validation (LOOCV). Let $v_{u,i}$ and $t_{u,i}$ be the testing observation in V and template observation in T respectively of the u -th user's i -th gesture. Then if we identify the user in a system-guided manner, i.e. the gesture type of $v_{u,i}$ is already known since the system proposes a gesture to the user to do, it is only compared with the templates which belong to the same gesture type. The observation $v_{u,i}$ is identified as the U -th user's performance,

$$U = \arg_{u'} \min_{1 \leq u' \leq 11} (D(v_{u,i}, t_{u',i})) \quad (4.19)$$

If the user has the freedom to perform any of the gestures in the vocabulary, in other words, the system does not know what type of gestures the user has performed, then $v_{u,i}$ has to be compared with templates of all gestures of all users. In this case

$$U = \arg_{u'} \min_{1 \leq u' \leq 11} \left(\min_{1 \leq i' \leq 10} D(v_{u,i}, t_{u',i'}) \right) \quad (4.20)$$

The concrete implementation of the Dynamic Time Warping algorithm and the gesture template selection approach in training procedure is the same as that of the gesture recognition approach.

4.4.2 Results

In the following, the performance of the identification approach is examined given different input data: single gesture, known concatenated pairs and unknown concatenated pairs.

Analysis with one symbol / System-guided configuration

First let us suppose that the system identifies different users with one single gesture and assume that the system knows exactly what gesture the user has performed. With the LOOCV procedure, each gesture is identified as a certain user by Eq. (4.19) and the Correct Classification Rates (CCR) for each gesture is gathered in Table 4.3.

ANALYSIS WITH CONCATENATED SYMBOLS / SYSTEM-GUIDED CONFIGURATION

Table 4.3: Identification accuracy with single gestures

Gestures	Accuracy (%)	Gestures	Accuracy (%)	Gestures	Accuracy (%)
Up	72.73	d	89.09	D	89.09
Down	64.55	w	90.00	W	79.09
Left	70.91	tv	93.64	TV	92.00
Right	79.09	p	82.73	P	83.64
Forward	83.64	l	92.73	L	80.91
Backward	81.82	ac	88.18	AC	87.27
Clockwise circle	91.82	r	88.18	R	90.91
Counterclockwise circle	92.73	f	89.00	F	88.18
Spiral to the left	92.73	m	92.73	M	85.45
Spiral to the right	95.45	s	88.18	S	90.91

The average CCR of all the 30 gestures is 86.24%. Circle and spiral movements and letter drawings favor more to identify the users than directional movements (identification accuracy with ‘down’ is only 64.55%). Small letters and capital letters as identifiers perform similar, although in general, the accuracy is a little higher with small letters than with capital letters. We can conclude that users tend to have similar templates for directional movements and the diversity of the writing of capital letters is weaker than the one of small letters.

Analysis with concatenated symbols / System-guided configuration

The identification results in Table 4.3 can be improved. Intuitively, the concatenation of gestures can provide more features to the identifier and it is possible to enhance the identification accuracy. Considering that the application scenario may need both directional/geometric movements and drawings of letters to interact with devices and that the writing of small letters has more diversity than the one of capital letters, we apply gesture chains that have the form $\langle \text{small letter} \rangle + \langle \text{directional/geometric movement} \rangle$. It is first tested with the system centric configuration, where Eq. (4.19) is modified as

$$U = \arg_{u'} \min_{1 \leq u' \leq 11} \sum_{k=1}^2 D(v_{u, i_k}, t_{u', i_k}) \quad (4.21)$$

where v_{u, i_k} and t_{u, i_k} are the k -th gesture in the gesture concatenation of user u , $1 \leq i_k \leq 10$.

The CCR has significantly improved when using gesture concatenations as shown in Table 4.4, with average accuracy of all possible combinations of directional/geometric movements and small letters 96.43%. The combinations of small letters and forward, circle, spiral movements distinguish users almost perfectly. Combined with small letters, the identification with directional movements also works much better than the corresponding single gestures shown in Table 4.3. The writings of different letters have similar identification accuracy so

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

Table 4.4: Identification with concatenated gestures, system-guided

Accuracy (%)	d	w	tv	p	l	ac	r	f	m	s
Up	95	95	96	93	96	93	95	93	96	95
Down	92	95	96	94	95	94	98	95	95	96
Left	93	91	91	93	95	91	93	95	92	93
Right	96	95	95	95	97	93	96	96	95	95
Forward	98	96	97	98	98	98	97	96	98	96
Backward	93	95	95	94	96	95	96	95	95	98
Clockwise circle	98	95	96	99	98	96	97	97	97	97
Counterclockwise circle	99	95	97	100	100	99	99	98	100	100
Spiral to the left	99	99	100	97	99	97	98	97	98	99
Spiral to the right	99	99	99	99	99	99	98	98	99	99

that the result of gesture concatenation mainly relies on the type of directional movements in it.

Analysis with concatenated symbols / User-centric configuration

In an alternative service scenario, the system may employ implicit identification to get better user experience, and it may need to identify the user with the gestures she is doing for other purposes than identification during interactions. In this case, it is assumed that although the system does not know what gesture the user has performed exactly, it does know what type of gesture it is, i.e. it is a directional/geometric movement or small letter(s) (we can assume this because the user should respect the proposed grammar to build meaningful interaction sentences), then Eq. (4.20) is extended as:

$$U = \arg_{u'} \min_{1 \leq u' \leq 11} \left(\sum_{k=1}^2 \min_{1 \leq i' \leq 10} D(v_{u,i_k}, t_{u',i'}) \right) \quad (4.22)$$

The identification results are shown in Table 4.5 and the average accuracy is 95.51%, which is a little lower than the system-guided case. It is reasonable because for this test both the user and the gesture have to be identified. As the gesture cannot be detected perfectly, the whole identification accuracy suffers a small loss of accuracy compared with the system-guided case. The user identification performance in this case shows similar accuracy variation tendency and distributions along rows and columns as the ones analyzed in the system-guided case test.

Computational efficiency

With the test specifications introduced at the beginning of this Section, where the lengths of the two matching data sequences are both normalized to 20, each DTW matching costs

Table 4.5: Identification with concatenated gestures, user-centric

Accuracy (%)	d	w	tv	p	l	ac	r	f	m	s
Up	95	95	95	92	95	92	94	92	95	95
Down	92	94	95	94	95	94	96	95	95	96
Left	91	90	90	93	95	91	92	94	92	93
Right	95	95	95	95	96	93	95	95	95	95
Forward	97	95	96	97	98	98	95	95	98	96
Backward	91	95	95	94	96	95	95	95	95	97
Clockwise circle	97	94	95	96	95	94	95	95	96	96
Counterclockwise circle	98	94	97	99	100	98	97	96	100	100
Spiral to the left	97	96	96	95	98	96	95	95	98	98
Spiral to the right	98	97	98	98	99	97	96	96	99	99

around 5ms and the preprocessing needs about 0.5ms.

Let us suppose that we already have the templates for each gesture of each user and that there are U users to identify, G_{sl} small letters and G_m simple movements in the gesture vocabulary. Gesture chains having L concatenated gestures are used to identify the users. Then the test with known gesture concatenations needs $U \times L$ DTW matchings and $U \times L + 1$ preprocessing procedures for each recognition of testing concatenation. On the other hand, the test with unknown gesture concatenations (consider the format of concatenation $\langle \text{small letters} \rangle + \langle \text{simple movements} \rangle$) needs $U \times (G_{sl} + G_m)$ DTW matchings and $U \times (G_{sl} + G_m) + 1$ preprocessing procedures.

In the above test, $U = 11$, $G_{sl} = G_m = 10$, $L = 2$, thus the program needs 125.5ms and 1210.5ms to identify the user for the system-guided case and user-centric case respectively, including DTW matching and preprocessing. Considering that in the second case the time requirements to perform identification may not need to be so strict (as the user is not expecting any feedback, just interacting with other purposes), we can state that both performance times are perfectly valid for services in smart spaces.

4.5 Conclusion

After reviewing the challenges to recognize gestures in terms of gesture vocabularies, variations of gesture performance and captured data, as well as gesture segmentation, the design requirement of the gesture recognition approach is specified for the gesture-based interaction system in this work. The gesture-based user identification problem has similar design challenges and requirements. The Dynamic Time Warping (DTW)-based strategy is chosen due to its simplicity, small demand of training and the support of customizable gesture vocabulary. Then based on the similarity measurement by DTW, which is introduced in Chapter 4.2, the architecture of the recognition and identification approaches are described, featuring the preprocessing procedure, the template selection approach in the training, and

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

the recognition and identification via matching the test sample with the templates. In the rest of the Chapter, the proposed recognition and identification approaches are evaluated in offline tests, whose testing scenarios (dataset, data recording procedure, running environment and evaluation manner) and results are described in detail.

As for the gesture recognition approach, although the signals retrieved from the Kinect and the smartphone differ, the test result demonstrates that the same recognition algorithm based on Dynamic Time Warping can successfully work for both input types. The average gesture recognition accuracy of the proposed DTW algorithm is 98.64% for Kinect-based recognition and 93.63% for smartphone-based one respectively on a 30-gesture vocabulary (with the assumption that the gesture type is known). To recognize one gesture, a little more than 50ms is enough for both data types in the testing scenario. Since settings of off-line test such as gesture set, number of participants, repetitions, sampling rate of sensors, data collection approach, running environment, evaluation approach (Leave-One-Out cross validation, n-fold cross validation, etc.) vary in almost every work, direct comparison of recognition accuracy and running time with the literature would not be reasonable. In addition, the algorithm is user dependent and requires its users to train their own gestures to perform the best. The analysis of the impact of training amount on recognition accuracy in Chapter 4.3 shows that three repetitions are enough to get nearly the best recognition accuracy. Reviewing the design requirements mentioned in Chapter 4.1, the offline test result demonstrates that the DTW-based recognition approach meets these requirements.

The gesture recognition approach addressed some of the design challenges pointed out in Chapter 4.1. First of all, to concretize the recognition target, a gesture vocabulary including directional/geometric movements, small letters and capital letters is defined as grammar components given the application scenario in this work. Running in a user-dependent manner, the recognition approach avoids the influence of performance variation between users. In addition, the time-normalization feature of DTW is robust to the speed variation of gestures. However, its tolerance to other variations such as the gesture trajectory shape, the stroke order and hand orientations needs further evaluations. The view variation introduced by optical tracking is partially compensated by the coordinate transformation in the pre-processing step so that the recognition is independent of the gesture's (hence the user's) location. The users' relative orientation to the camera however, is not considered. As far as the gesture segmentation is concerned, the gesture data collection program is designed so that the gesture boundary is marked manually.

In order to improve the identification accuracy with single gestures, the identification on pairs of concatenated gestures (smaller letters + geometric movements) is tested as well. This choice has significantly improved the identification accuracy (10% in average), making the system performance feasible for non-critical services. Both system-guided and user-centric cases are considered, reaching average accuracy 96.43% and 95.51% respectively. In the user-centric mode, the system lets the user perform whichever pair of gestures under the predefined structure, usually as part of a more complex interaction flow. In this case, the system has to perform gesture recognition as the initial part of the identification process. As a consequence, an additional running time is needed compared to the system-guided mode,

and there is also a small loss in the identification accuracy. In the system-guided case, on the other hand, the system performs almost immediate identification. The scale of the system, i.e. the number of users and the number of gestures in the vocabulary should be controlled so that the response time is not unbearably long for the users. However, it is important to note that in the user-centric mode, the time factor is not so critical, as the identification is transparent to the user, who is not necessarily waiting for a feedback. Due to distinct settings of offline test (as mentioned above when the results of gesture recognition are discussed), direct comparison of the accuracy and running time to other works would not be reasonable.

The DTW-based gesture recognition and user identification has different performance given different gesture types. In the user-dependent scenario, as shown in Chapter 4.3, the directional movements and small letters are better recognized than the capital letters. In the user-independent case, motions with unified performing manners like straight movements and capital letter ‘S’ tested on the Kinect are recognized with high accuracy and small standard deviation. It seems that the gestures of circle, spiral movement and letters, which have lower rate in the user-independent recognition test, have more “personal information” and facilitate user identification as shown in Chapter 4.4. These observations provide an empirical guide to select the gesture vocabulary in different scenarios when this DTW-based strategy is employed. On the other hand, it also indicates that the promise of arbitrary personalization of gesture sets will not be guaranteed owing to the discrimination of the DTW-based approaches in performances on different gestures due to the limitation of the DTW classifier.

The investigation on gesture recognition and gesture-based identification has limitations. For example, the proposed approaches and the tests did not consider the gesture performance variations of one user in different days. False positive happens when a gesture out of the predefined vocabulary is performed and recognized as one gesture in the vocabulary or in the case of user identification an unknown user is identified as one trained user. The current approach considers a gesture input “not recognized” if the DTW distances between the input sample and templates are larger than an empirical threshold. Further investigation on approaches to avoid false positives is needed. In addition, as aforementioned, the approach did not consider the user’s relative orientation to the Kinect camera and the holding-posture variation when gesturing with the phone. These limitations are interesting directions to work in the future. Other possible extensions include the integration of probability-based approaches (like GMMs in [10]) to model the variations of gestures to improve the training and recognition, identification test with acceleration data, the evaluation of the impact of training repetitions on user identification, further investigation of the strategy to optimize the training proposed in Chapter 4.3, and an approach or tool to guide the design of a gesture vocabulary with high recognition rate by DTW.

The offline evaluation of gesture recognition gave positive results and is part of the design foundation of the user study as we will see in Chapter 5. For example, since the DTW-based approach works on both position and acceleration data, the Kinect, the smartphone and the smartwatch are included in the user study. Small letters instead of capital letters are selected as part of the gesture vocabulary due to the slightly higher recognition rate. The test on

CHAPTER 4. A STRATEGY FOR GESTURE-BASED RECOGNITION AND IDENTIFICATION

the impact of training repetitions reveals that with three training samples the recognition accuracy is almost as high as with nine (or four in the case of the phone) and it would be reasonable to ask the participants to train with less repetitions in the user study to reduce fatigue. On the other hand, it is very necessary to evaluate the recognition approach in a quasi-real environment since the gesture data in this Chapter is collected in an isolated and restricted setting, and the possible errors occurred in the gesture collection step are well controlled. In addition, the time responsiveness of the recognition approach is important for a practical interaction system and highly related to the deployment and implementation. Moreover, since the computational time increases linearly with the size of vocabularies for the DTW-based approach, it would be meaningful to check the running time of a DTW matching to determine the reasonable size of the vocabulary to ensure real-time respond.

The gesture-based identification approach works with high accuracy on concatenated gesture pairs. Users may perceive it less invasive as it does not use human recognizable features (e.g. voice, face, etc.). It can work as part of an interaction model. However, this approach is not integrated in the user study in Chapter 5. Instead, in the work of Cuevas et al. [24], the gesture-pair based identification, enhanced by additional filters and effective trimming strategy, is implicitly integrated in a sequence of double-option questions that aims at configuring the smart space based on the user's preference. Ten users tried and provided positive feedback.

Chapter 5

User study

Once the gesture-based interaction system including the gesture recognition approach has been designed and implemented, this Chapter explores research questions regarding its use in a real environment, through the design of a user study that evaluates the interaction system and more importantly, derives a set of design recommendations for novel gesture-based interaction tools.

5.1 Research questions

This Chapter aims at analyzing diverse issues related to the real operation of the proposed gesture-based interaction method. In particular, the research questions to answer are related to: the gesture recognition performance, the adequacy of the grammar of the gesture command and the goodness of interaction experience as a whole.

- **The gesture recognition performance.** The first group of research question is related to the fact that the test scenario of the offline evaluation on the recognition approach in Chapter 4 is highly restricted. This user study aims at measuring the performance of the recognition approach in a quasi-real environment where gesture signals are collected through the gesture input procedure introduced in Chapter 3, the system is trained with few samples and the gesture is recognized online, with real-time feedback in the interaction. The metrics include not only the recognition accuracy and response time, but also the performance of the gesture input procedure and the impact of user postures. As far as the author knows, comparison of gesture recognition performances among the three technologies: the Kinect, the smartphone and the smartwatch, and examination of effect of postures on recognition accuracy in online test, which will be part of the user study, are not considered in the existing literature.
- **The grammar of the gesture command.** The proposed grammar of gesture commands in Chapter 3 is here used to organize the control of a set of smart resources

in a smart space. The second group of research questions explores the users' opinions regarding a) the proposed vocabulary, b) the use of location-based context information to filter the object to control (enabling the portability of the gesture recognition), c) the acceptance and usability of the grammar, as well as d) the limitations of the grammar and other possible control paradigms.

- **The interaction experience.** The third area of the research questions focuses on the user interface, user experience and usability of the system, comparing the three enabling technologies of the Kinect, the smartphone and the smartwatch. The test with a voice-based system makes a comparison between gesture-based and voice-based interaction modes. In addition, social acceptability and physical exertion are also considered.

5.2 The experimental design

5.2.1 Design

A within-group design, in which the same group of subjects serves in more than one treatment, is adopted in this user study. Compared with between-group design, it requires a smaller sample size and reduces the impact from individual differences on the collected data. Smaller sample size facilitates the recruitment of participants and the organization of the user study. Individual differences should be controlled since they might influence the dependent variables and hence introduce conditions not of interest. However, this design has the disadvantage that it is subject to impact from learning effects and fatigue. To control the learning effect, the order of the experimental conditions will be randomized, and several trials by the participants will be done before the formal test. Opportunities to take breaks will be provided to address the problem of fatigue.

5.2.2 Participants and apparatus

Twelve subjects (10 males and 2 females) aged 23-54 ($M = 32$, $SD = 9.0$) participated in the user study (four of them participated in the gesture data collection for the offline test in Chapter 4 in the first half of 2013). The participants, with different professional profiles (not necessarily technical ones), were recruited among the personnel in the working environment and in no case were related to the project. One of the participants is left-handed. Seven of them usually wear watches or bracelets. Five of them use Microsoft Kinects sometimes or with higher frequency, and all use their smartphone daily. Most of them seldom use the Leap Motion or the Wii. Two of them wear a smartwatch daily or often, and the rest had never used it. A single participant often uses gesture-enabled devices not mentioned above and the rest seldom or never uses them. Five, seven and two out of the twelve users have experience of developing applications of the Kinect, the smartphone and other gesture-enabled devices, respectively. At the end of the test, the subjects received some chocolate bars as reward.

The user test was carried out in September 2015 in the “Experience Lab of Future Spaces” (ExpLab), a $120m^2$ installation in the UPM’s Montegancedo Campus. This Lab is equipped with sensors, actuators and multimedia resources that may be used to simulate different environments (e.g. office, home, hospital, hotel room, lounge, etc.). To set the full GENIE interaction system to work, it was necessary to use the infrastructure deployed in the ExpLab, which includes a Smart Home Controller (with APIs to manage the sensors, actuators and smart objects in the lab) and a Media Controller (in charge of managing multimedia contents among the devices in the space by using DLNA standard). A Bluetooth-based Positioning System is used to deliver location information to GENIE. As described in Chapter 3, the interaction system has client-server architecture. The server program is implemented in C++ using Qt, and runs in a Mac mini. Three types of clients are implemented based on the Kinect, the smartphone and the smartwatch. The Kinect v1 devices are deployed in three rooms, as shown in Figure 5.1 and each is powered by a computer with a display and an instance of the client application, which is implemented with Qt as well. A Samsung Galaxy S4 smartphone and a Pebble Classic smartwatch are used running Android and Pebble client apps respectively. In addition, the watch app works via Bluetooth with a supplementary Android app in the smartphone, which takes charge of data processing and communication with the server.

5.2.3 Procedure

The user study is organized into four main stages: Training, Testing of the Recognition System, User Journey and Voice-based Interaction (Table 5.1). The Training stage enables the gesture recognition system to work. Then the second stage evaluates the online performance of the gesture recognition approach on the three technologies considering the posture factor. The participants complete a series of daily operations with the gesture-based interaction system in the User Journey stage to evaluate the grammar concept and the interaction system. In the final stage of the study, a similar set of tasks to the User Journey is accomplished with vocal commands. Although the result of the off-line test of gesture-based user identification on the Kinect is encouraging, it is not perfect, and to focus on the gesture-based interaction and simplify the user study, gesture-based user identification is not studied in this test.

Table 5.1: The user test sequence

Introduction
Greeting and introduction to the purpose of the study and the procedure
Questionnaire for user profiling
Training (Hall)
T0. Assign the order of technology with complete randomization
T1. Explanation of the interface and how to do gestures
T2. 2-5 pre-test trials of performing gestures, until the user feels comfortable

Table 5.1: (Cont.) The user test sequence

T3. Train the 10 gestures one time for each with a specific technology in standing posture
T4. Questionnaire for the specific technology
T5-T12. Repeat T1-T4 2 more times with the remaining technologies
T13. Global questionnaire of training
Recognition test (Hall and meeting room)
T14. Assign the order of technology-posture combination with balanced Latin square
T15. Explanation of the interface and how to complete this test
T16. Complete the test (random sequence of 10 gestures, 3 times each) with the certain technology-posture condition
T17-T26. Repeat T15-T16 five more times for the other five conditions
User journey (Hall, meeting room, relax room)
T27. Explanation of the grammar
T28. Questionnaire on gesture preference
T29. Explanation of the user journey and the location-aware feature
T30. Assign the order of technology, which is the same as that in Training
T31. Explanation of the interface of a certain technology
T32. Pre-test trials of several tasks, until the user feels comfortable
T33. User journey with the technology
T34. System Usability Scale (SUS) questionnaire
T35. User Experience Questionnaire (UEQ)
T36-T45. Repeat T31-T35 2 more times for the remaining technologies
T46. Comparative questionnaire
Voice-based interaction (Meeting room)
T47. Explanation of this voice-based test
T48. Pre-test trials of several tasks, until the user feels comfortable
T49. Tasks using vocal commands
T50. System Usability Scale (SUS) questionnaire
T51. Voice-gestures comparative questionnaire
T52. Gratitude and compensation

Since the recognition approach works in a user-dependent way, the participants need to train the system to interact. The gesture set in the user study is system-defined. Guessability [104] is a classical method to design a user-centered gesture set. However, as Nebeling et al. [74] pointed out, the elicited gesture set by this methodology can have conflicts and lack consistency, and some gestures might have low consensus. Free customization of the gesture set by the users is not adopted since too similar gestures, difficult to distinguish can be proposed. However, the gestures proposed by the participants are collected in a questionnaire in the user study to explore the characteristics of personalized gesture sets. To control

Figure 5.1: Views of the three zones in which the user study are carried out: Hall area, Meeting room and Relax room

the learning effect, the conditions should be counterbalanced. The balanced Latin square is better than the normal Latin square, as it ensures that each condition appears before and after each other condition an equal number of times. However, the technology variable has 3 levels and balanced Latin squares do not exist for odd-order squares. Therefore, complete randomization, which contains all the permutations, is used in this case for counterbalancing. To further control the learning effect, the participants will try to perform several gestures before the formal training. The 10 gestures involved in the whole user study include six small letters or acronyms (w, b, h, l, tv and r), and four geometric motions (backward, forward, down and clockwise circle, illustrated in Figure 4.2) and all of them will be used in the User Journey stage. The participants will train the 10 gestures in random order and repeat one time for each gesture in a standing posture. After the training is completed with one technology, the participant will evaluate the training by filling a questionnaire. Once the three implementations are trained, a summary questionnaire is filled out.

The Recognition test has two independent variables: technology with three levels (the Kinect, the smartphone and the smartwatch) and the user's posture, with two levels (standing and seated). The 6 conditions by the combination of the two variables are counterbalanced

CHAPTER 5. USER STUDY

by a 6-order balanced Latin square. For each technology-posture condition, the test has three rounds and the 10 gestures are performed one time in each round in random order. The Kinect-seated condition is tested in the Meeting room since the Kinect device in the Hall is too high for the seated posture, while the rest of the conditions are in the Hall (see Figure 5.1).

The concept of the grammar is first introduced to the participants in the User Journey stage. Then they fill out a questionnaire on the preference of gesture vocabularies of operations for nine commands, including raising temperature, checking weather forecast, pulling down blinds etc. After this questionnaire, the procedure of the User Journey test is described to the participants as a daily scenario - “You are going to have a journey through various rooms that simulate a home environment: in the Hall, you will check the weather forecast. Then in the Relax room, you will pull down the blind and switch on HUE light in red. In the Meeting room, you will want to watch a video in the TV set and before playing the video you will turn on the lights in TV mode and ask the robot to bring a beverage”. The location-aware feature, which maps the target resource considering the user’s location information, is introduced as well. To simplify the test, the order of the technologies is the same as in the Training for the same participant and the order of the tasks does not vary. The tasks are listed in Table 5.2. These six tasks include managing one web-based service (the weather forecast information, which appears on the TV screen for the Kinect application and on the smartphone and the smartwatch applications), four actuator-based services (blinds, HUEs, lights and a robot, the latter navigates along a predefined route with the help of visual marks on the floor) and one multimedia service (playing video on the TV). The tasks are frequent and common operations of daily life. In addition to the services to manage lights, blinds and TV, which are considered in previous works [79, 57, 28], this work evaluates additional services such as the weather forecast query and the robot commanding service. The order of technologies is the same as that in the Training and it should not affect the quality of the data since the dependent variables in the User Journey stage and the Training stage will not be compared. The fixed order of the tasks will introduce biases from learning and fatigue effect when comparing the metrics between tasks. However, the comparison of the tasks is not the primary research question in the test. Before a journey with a certain technology, the facilitator will tell the participant the gesture commands for each task and the participant will try to complete the task. During the formal journey, the participants will have to recall these commands. The standard questionnaires of System Usability Scale (SUS) [16] and User Experience Questionnaire (UEQ) [60] are filled out along with some open questions about the user interface after the journey with the given technology. Once all the journeys with the three technologies are completed, a summary questionnaire is applied to collect the opinions and feelings about physical exertion, usability of the grammar, social acceptability of the interaction with the system, user preference on the three technologies, etc.

In the final stage, the participants will complete the four tasks listed in Table 5.3 with vocal commands using a Kinect-based application. Three of the tasks (the second, fourth and sixth) in Table 5.2 are duplicated in Table 5.3 on purpose for comparison between gesture-based and voice-based interactions. The order of the tasks is also fixed as in the

Table 5.2: The user journey tasks (C.C. = clockwise circle)

Order	Task description	Location	Posture	Compared with voice-based interaction
1	Check weather forecast (w-backward)	Hall	Standing	No
2	Blinds down (b-down)	Relax room	Standing	Yes
3	Set HUE to red (h-forward-r)	Relax room	Standing	No
4	Set lights in TV mode (l-forward-tv)	Meeting room	Seated	Yes
5	Tell robot to approach (r-backward)	Meeting room	Seated	No
6	Play movie (tv-C.C.)	Meeting room	Seated	Yes

Table 5.3: Tasks with the voice-based application

Order	Task description	Vocal command
1	Switch blue lights on	Enciende luz azul
2	Pull blind down	Baja estor blanco
3	Play music	Reproduce música televisión
4	Play movie	Reproduce vídeo televisión

User Journey stage. Before the test, the participants will be told the vocal commands and try several tasks. Then in the formal test, they will have to recall the commands. Finally, the participants will be asked to complete the System Usability Scale (SUS) questionnaire on the voice-based application and to compare gesture-based against voice-based interaction.

In summary, the Training and Recognition test stages collect 360 samples (12 participants $\times$ 10 gestures $\times$ 3 technologies) and 2160 samples (12 participants $\times$ 10 gestures $\times$ 3 repetitions $\times$ 3 technologies $\times$ 2 postures) respectively. In the User Journey stage, the participants perform at least 504 gestures (12 participants, 3 technologies, 4 tasks of 2 gestures, 2 tasks of 3 gestures) since they will have to repeat if a gesture is not recognized well. During the whole test, each participant will perform a number of 252 gestures excluding possible repetitions and retraining, and each of the ten gestures is performed at least 8 or 9 times with each technology. Thanks to the usage of the grammar, gestures in one command can be reused in another. For example, the gestures of backward, forward, r, and tv are reused among the commands in Table 5.2.

The user study lasts around one hour and a half with maximum of around two hours, and it is video recorded for posterior annotation and analysis. The participants are requested to “think aloud” during the entire test. The facilitator will provide help if after 3 trials the task still cannot continue, i.e. for example, the participant cannot successfully register a gesture, or forgets the vocabulary or the grammar.

5.3 Results

5.3.1 The collected data and the analysis procedure

To investigate the research questions in Chapter 5.1, the user study is designed to measure the metrics listed in Table 5.4, along with their data collection approaches. The metrics mainly cover task success, task time, error types and subjective evaluations in questionnaires. There are three sources of collected data: questionnaires, data logging and video recording. Readers can refer to [98] for the entire questionnaire, which is presented in Google Form and was filled out in a tablet in the user study. The questionnaires are also presented in Appendix C for convenience. The software of the applications for the Kinect, the smartphone, the smartwatch and the server were adapted to log data such as timestamps, ID, length, storage path and data of gesture samples, results and time of gesture recognition, etc. In addition, the whole procedure of the user study was recorded by the moderator, including oral comments of the participants.

Table 5.4: Metrics and data collection methods

Target modules	Metrics	Explanation	Data collection method
Gesture data collection	Success of enabling/disabling gesture recording	Effectiveness of gesture delimiters	Logging (e.g. ID, length, data of gesture samples), video recording
	Time to complete a gesture	The interval between recording enabled&disabled	Logging of timestamps
	Errors	Failure to collect correct and complete gesture data	Logging and video recording
	Questions on UX, preference	-	Questionnaire
Training	Success	All gestures are trained	Video recording
	Time	Time spent on training	Logging of timestamps
	Errors	Mistaking gesture; gesture collection error	Video recording, results from gesture data collection
	Questions on repetitions, UX etc.	-	Questionnaire
Gesture recognition	Accuracy	No. recognized divided by trials	Logging of recognition results
	Time complexity	Recognition time (preprocessing and DTW match)	Logging of timestamps
	Errors	Confusion of algorithm or gesture collection error	Results from gesture data collection module
	Task success	The service is triggered	Video recording

THE COLLECTED DATA AND THE ANALYSIS PROCEDURE

Table 5.4 – continued from previous page

Target modules	Metrics	Explanation	Data collection method
	Task time	From moderator tells task to service is triggered	Video recording
	Errors	Errors from gesture collection, recognition, grammar and UI design	Logging and video recording
	Efforts	Physical exertions	Borg Scale in questionnaire
	Questions on grammar, usability, UX etc.	-	Questionnaire
Services	Success	Service triggered and completed	Video recording
	Time responsiveness	Delay between end of command and start of service	Video recording
	Errors	Services not triggered or not completed	Video recording
Voice-based interaction	Task success	Service is triggered and completed	Video recording
	Task time	From moderator tells task to service is triggered	Video recording
	Errors	Errors of recognition, grammar	Feedback from application, Video recording
	Questions on preference, UX, etc.	-	Questionnaire

Starting with the collected raw data, the analysis procedure had three stages. At first, the raw data were checked for possible errors such as outliers and missing values. Then they would be coded if necessary. For example, the answers to open questions in questionnaires were coded and the recorded videos were annotated. The open questions include comments on interface feedback, suggestions for improving training and user interface, general comments on user interface, alternative grammar and possible solutions to vocabulary conflicts, and any additional comments (See Appendix C). Since the questions are not that “open” and the responses from the participants are short (one or two sentences), the analysis of the texts did not go through rigorous coding and validation methodology for qualitative data analysis but instead, the responses were summarized and presented in this subchapter. In addition, the purpose of analysis of the texts is to find potential defects of the system and figure out ways to improve. Regarding the video annotation, which was conducted with ANVIL, a free video annotation tool developed by Michael Kipp [47], the extracted metrics include error types of: gesture performance, user journey tasks, services, the grammar and the vocabulary, and voice-based tasks, time: task time, service response time, “think

aloud” comments and observations. Similar to the analysis of responses to open questions in questionnaires, the “think aloud” comments and observations from the video annotation were just summarized. A drawback of the analysis of the responses to open questions and video annotation lies in the possible low reliability of the analysis results because they were processed by a single coder in one time, so that the analysis could suffer from a bias by that coder’s subjective opinions at that particular time of the analysis. In the third stage of analysis, statistical methods were applied on the data (preprocessed if necessary, e.g. calculation of score of System Usability Scale (SUS) from raw responses). The statistical analysis can be based on the calculation of means, medians and standard deviations, or comparison of means with statistical methods such as t-test, ANOVA, etc. Readers can find a brief introduction to the statistical methods used in this article in Appendix B. Additionally, some forms and standard methodology for evaluation in the literature have their own analysis techniques, such as User Experience Questionnaire [60], Borg Scale, gesture agreement analysis [104], etc. The specific analysis methods that have been used will be mentioned when analyzing the results. The following three Sections present the results and their analysis, and basically corresponds to the three research question areas in Chapter 5.1.

5.3.2 Evaluation of training and gesture recognition performance

The information gathered within the Training and Recognition tasks is analyzed in order to evaluate issues such as recognition accuracy in real time, the main sources of error (intrinsically linked to design weaknesses) and the efficiency of the training and gesture recognition, for the three technologies.

Analysis of the correct recognition rate

Figure 5.2 demonstrates a summary of the results of the recognition test, while a closer look into the results is shown in Figure 5.3. Prior to applying any statistical test to compare the accuracy results for the three technologies in the standing and seated postures, a Shapiro-Wilk test has been performed. It shows that it is not possible to reject the null hypothesis that accuracy data come from a normally distributed population. Moreover, Mauchly’s test shows that the sphericity hypothesis is not rejected for the existing data (the variances of the differences between all combinations of the related groups are equal). Thus a two-factor repeated measures ANOVA can be applied to test if the three technologies are equally efficient to enable gesture-based interaction in smart spaces independently of the posture. The test shows that the null hypothesis of equal means of the recognition rates cannot be rejected neither for the factor ‘technology’ ($F(2, 22) = .68, p = .52$) nor for the factor ‘posture’ ($F(1, 11) = 1.1, p = .32$) or the combination of both of them ($F(2, 22) = .08, p = .93$), for a 5% level of significance. This means that there is not substantial difference in the obtained recognition rates for these independent variables.

Statistical significance of the difference between the recognition rates of directional (non-letter) and letter-drawing gestures for each technology is examined. In the case of the

ANALYSIS OF THE CORRECT RECOGNITION RATE

Figure 5.2: Average recognition rate with the three technologies. Error bars show one standard deviation

standing position, paired-sample t-tests are carried out to check the null hypothesis “the difference between the average recognition rate for letters and directional gestures is 0” for each technology. The null hypothesis is rejected for the cases of the Kinect ($p = .03, t(11) = 2.52$) and the Phone ($p = .01, t(11) = 2.99$), while it is not rejected for the Watch ($p = .13, t(11) = 1.64$), at the 5% significance level. No significant difference was found in the case of the seated position.

The confusion matrix for the Kinect in Table 5.5 shows that the best recognized gesture is the ‘small l’ followed by the ‘down’ gesture. Gestures such as the ‘small h’ and ‘r’ have a low recognition rate and are usually confused. In the case of ‘h’, it is confused with ‘b’, while in the case of ‘r’, ‘w’ is the recognized gesture. In the case of the acronym ‘tv’, it is usually confused by the letter ‘w’. The gestures are unequally recognized when standing (77.5% correctly recognized) or seated (75.6%), providing the Kinect a 76.5% average recognition rate. In the standing position, seven gestures are correctly recognized in more than 83% of cases, while in the seated position the highest percentage reaches 83.3% for the ‘forward’, ‘b’ and ‘l’ gestures. On the contrary, ‘r’ and ‘tv’ are better recognized in the seated position (20% and 10% improvement respectively). When compared with the off-line analysis in [99], it can be seen that the recognition is substantially degraded: the average recognition rate was around 98% with one training sample.

The Phone is providing an average recognition rate of 76%. The ‘small h’ is usually confused with the ‘b’ and the ‘r’, the ‘small l’ with the ‘h’ and the ‘tv’ gesture with the ‘w’. The ‘r’ is significantly better recognized when compared to the Kinect, being confused with the ‘l’ and the ‘b’. The recognition performance in standing posture is slightly higher than that in seated posture (77.5% vs. 74.4%). The performance is significantly worse than in the off-line test, in which the average recognition rate was of around 90% with one training sample.

The Watch is providing an average recognition rate of 71.1%. Once more, the ‘small h’ and ‘tv’ are the gestures delivering the worst recognition rates. The first one is insistently

(a)

(b)

Figure 5.3: Recognition rates of different gesture types with the three technologies in two postures. Error bars show one standard deviation

confused with the ‘b’ and the second, with the ‘w’, as in the Kinect case. The ‘backward’ gesture is confused with other linear gestures and the ‘small b’ is confused with ‘h’ and ‘l’. Regarding the difference between standing and seated positions, the percentages are quite similar (72.2% vs. 71.1%). The ‘small h’ is particularly misrecognized when seated (38.9%), being confused with the ‘b’ in a 44.4% of cases.

Although the test settings are different from the previous works in the literature and direct comparison might not be appropriate, the results of similar user studies are following summarized. The recognition rate in [79] using IR camera and emitter is 37.25% in the first trials of the participants. With five training samples, the mean recognition rate with the smartphone in [57] is 51.33%. The tasks in [28] equipped with color cameras achieved high success rate of 96.5% but the gesture vocabulary was relatively simple: up, down, left, right, clockwise and counterclockwise.

ERROR ANALYSIS IN GESTURE INPUT AND RECOGNITION

Table 5.5: Confusion matrix with the Kinect (C. C. = clockwise circle, recog. = recognized)

	Recognized as on the Kinect (%)										
	Down	Forward	Backward	C. C.	small b	small h	small l	small r	small w	small tv	not recog.
Down	86.1	4.2	2.8	6.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Forward	1.4	84.7	12.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4
Backward	4.2	11.1	80.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.2
C. C.	4.2	12.5	1.4	81.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
small b	0.0	0.0	0.0	0.0	84.7	4.2	2.8	1.4	2.8	4.2	0.0
small h	0.0	0.0	0.0	0.0	15.3	61.1	6.9	2.8	1.4	9.7	2.8
small l	0.0	0.0	0.0	0.0	0.0	8.3	88.9	0.0	2.8	0.0	0.0
small r	0.0	0.0	0.0	0.0	1.4	13.9	8.3	54.2	16.7	5.6	0.0
small w	0.0	0.0	0.0	0.0	1.4	1.4	0.0	0.0	81.9	15.3	0.0
small tv	0.0	0.0	0.0	0.0	1.4	0.0	0.0	2.8	31.9	61.1	2.8

Table 5.6: Confusion matrix with the Phone

	Recognized as on the Phone (%)										
	Down	Forward	Backward	C. C.	small b	small h	small l	small r	small w	small tv	not recog.
Down	77.8	15.3	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.8
Forward	8.3	79.2	11.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4
Backward	1.4	1.4	95.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4
C. C.	5.6	6.9	6.9	80.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
small b	0.0	0.0	0.0	0.0	75.0	9.7	5.6	6.9	1.4	1.4	0.0
small h	0.0	0.0	0.0	0.0	15.3	63.9	1.4	13.9	2.8	1.4	1.4
small l	0.0	0.0	0.0	0.0	2.8	13.9	65.3	2.8	5.6	1.4	8.3
small r	0.0	0.0	0.0	0.0	8.3	5.6	13.9	70.8	1.4	0.0	0.0
small w	0.0	0.0	0.0	0.0	0.0	2.8	1.4	0.0	87.5	8.3	0.0
small tv	0.0	0.0	0.0	0.0	8.3	0.0	0.0	2.8	22.2	63.9	2.8

Table 5.7: Confusion matrix with the Watch

	Recognized as on the Watch (%)										
	Down	Forward	Backward	C. C.	small b	small h	small l	small r	small w	small tv	not recog.
Down	84.7	5.6	5.6	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Forward	9.7	80.6	9.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Backward	15.3	19.4	65.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C. C.	5.6	4.2	5.6	84.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
small b	0.0	0.0	0.0	0.0	63.9	12.5	11.1	6.9	4.2	0.0	1.4
small h	0.0	0.0	0.0	0.0	36.1	50.0	5.6	4.2	2.8	0.0	1.4
small l	0.0	0.0	0.0	0.0	2.8	12.5	79.2	0.0	0.0	1.4	4.2
small r	0.0	0.0	0.0	0.0	9.7	1.4	8.3	72.2	1.4	0.0	6.9
small w	0.0	0.0	0.0	0.0	0.0	2.8	5.6	1.4	76.4	9.7	4.2
small tv	0.0	0.0	0.0	0.0	6.9	4.2	1.4	1.4	25.0	59.7	1.4

Error analysis in gesture input and recognition

The following analysis includes both the training and gesture recognition tasks, with the aim of getting more knowledge about the error sources that hinder gesture recognition in the gesture input procedure. Very few works make a detailed analysis on errors on the gesture input procedure. Most of the reviewed papers [58, 79, 43] utilize reliable “push-to-gesture” strategy. The preferences of the users on methods to activate the gesture recognition, such

CHAPTER 5. USER STUDY

as voice commands and wireless handheld switch is explored in the gesture-based medical system [13], but the influence of the delimiter on recognition performance is not covered. This ignorance also applies to [28, 7], which employ a speed threshold and a pinch gesture to segment gestures respectively. This work examines the errors on the gesture input procedure to find out how to improve recognition performance.

Let us start by considering the Kinect case. The observation of the gesture performances has revealed some vulnerabilities in the design of the gesture input procedure with this type of technology. As described in Chapter 3.3, the captured gesture is the part of the movement of which velocity is larger than a threshold. It turns out that this can cause unintended gesture inputs, imprecise gesture boundaries from the point of view of the user and failure to start a gesture:

- *Unintended gesture input (UGI)*. The Kinect v1 requires the users to wave the hand to start hand tracking. Some participants waved hand strongly and the waving motion was considered as a gesture input or part of the input from the application. The gesture recording was also accidentally enabled when some participants tried to reach the virtual button on the screen.
- *Imprecise gesture boundaries (IGB)*. It is found that most (if not all) of the participants habitually had their hand rest in an area before and after the gesture performance e.g. the area at the height of the shoulder in front of the body. The movements from this resting position to the start point of the gesture or from the end of the gesture to the resting point was captured in the test of some participants because the velocity of the hand in these motions surpassed the threshold although they did not intend to gesture in these moments. Similar extra captured motions were found when gestures were performed consecutively and the hand moved from the end of the previous gestures to the start of the next one. Another misinterpretation of the boundaries of the gesture by the application happened when not all the gesture motion had the velocity larger than the threshold; for example, some participants accelerated the hand very slowly so that when the velocity was larger than the threshold a certain part of gesture trajectory had finished and that part of gesture data was lost.
- *Fail to start gesture (FSG)*. Contrary to the case of the unintended gesture input, some participants moved their hands more slowly than expected when performing gestures and failed to enable gesture recording.

Apart from the errors rooted in the design of the gesture input procedure (“errors” here mean any reason that make the application fail to collect correct and complete gesture data), some were caused by the *instability of hand tracking* (IHT) of the device or the *confusion by the participants* (CP) (for example, some performed ‘counterclockwise circle’ gesture while she should have done ‘clockwise circle’).

The errors were counted in the video annotation analysis and grouped by the error types described above. The error rate distribution, calculated by dividing the error counting by

ERROR ANALYSIS IN GESTURE INPUT AND RECOGNITION

Figure 5.4: The distribution of gesture performance errors with the Kinect. (UGI - unintended gesture input, IGB - imprecise gesture boundaries, FSG - fail to start gesture, IHT - instable hand tracking, CP - confusion by participant)

the total number of gesture performances (with and without errors) from all participants in each test section, is plotted in Figure 5.4. The main error sources are IGB, FSG and IHT, covering 88% of all errors. The IHT (instable hand tracking) error in the recognition test in seated posture stands out in the plot. Compared with the test environment in the recognition test of standing posture (in the hall with normal lighting condition), the test in seated posture was in the meeting room where the Kinect device faced strong direct sunlight. This could be the main factor leading to the frequent loss of hand tracking although further investigation would be necessary since other distinct conditions like the farther distance to the device from the participants and the different body posture can also contribute to this significantly more error occurrences than the test in the hall. Comparing the recognition test with the training section, obvious error rate reduction is shown in the FSG and CP error types in Figure 5.4 and also in UGI and IHT errors for the recognition test in the standing posture, demonstrating that the gesture performance improves along with more practice with respect to these error types. The IGB error rate is high and it is found that one participant committed around half of the total IGB errors in the recognition test. Even though this outlier is excluded, the improvement over training test is slight. Maybe the participants did not consider anything inappropriate when committing this type of errors and there was no intention to avoid this error. The relatively high rate of UGI error in the recognition test in seated posture is partially promoted by the frequent loss of hand tracking (IHT errors) because after the hand tracking interrupted, the participants had to initiate it by hand waving, which can cause UGI errors.

It was also observed that the ‘backward’ and ‘forward’ gestures were the top gestures leading to FSG errors and that ‘backward’ gesture caused the most IHT errors among all gestures.

In the case of the smartphone and the smartwatch, the recorded video and logged data in the training and recognition tests are not enough to determine whether the gesture input

Figure 5.5: Gesture performance error comparison among the three technologies in the training and recognition test (UGI - unintended gesture input, FSG - fail to start gesture, IHT - instable hand tracking)

procedure on these two devices captured gesture data with precise gesture boundaries. However, comparing the number of collected gesture samples and the number of actual gesture performances by the participants, the UGI (unintended gesture input) error and FSG (fail to start) error can be deduced.

It is categorized into the UGI error group that some participants shortly touched the screen of the smartphone mistakenly when the application was ready to collect gesture data. With the smartwatch, the gesture data recording starts when the participant turns the wrist such that the watch's screen faces downwards. This design caused unintended gesture input when the participants adjusted the watchband or naturally rested the arm down at the beginning of the test while the gesture recording was active. On the other hand, some participants rotated the wrist slightly and did not start gesture recording.

The gesture input errors of UGI, FSG and IHT are compared in Figure 5.5, although there are other error types and the IGB error counts are not available for the smartphone and the smartwatch. It is shown that with the smartphone only a few UGI errors were observed. The test with the smartwatch saw a relative high number of FSG errors with the design of wrist rotation to initiate gesture recording for the smartwatch. Contrasting to the high IHT errors with the Kinect, the accelerometers in the smartphone and the smartwatch always record the movement of the hand between the designed input delimiters, introduced in Chapter 3.3.

Apart from the errors described above, it was observed that some participants hesitated after activating the gesture recording with the smartwatch and this simply stopped the recording because the hand was detected as static although they had not yet done any gesture.

Figure 5.6: The mean training time with the three technologies. Capped error bars represent a 95% confidence interval

Time analysis

In particular, mean training time and gesture performance time provide hints of the efficiency of training and gesture input, and the time analysis of gesture input workflow indicates the weights of different phases in the workflow and what phase(s) of the workflow should be improved to enhance the efficiency of the gesture input. These metrics are rarely measured in the literature.

In the Training stage, the participants were first introduced to the application interface and were encouraged to try several gestures until they got familiar to the interface. The training time was then measured, repeating each of the ten gestures once. The result is shown in Figure 5.6.

A one-way repeated measures ANOVA shows that the training time presents a statistically significant difference among technologies ($F(2, 22) = 18.687, p < .0005$, partial $\eta^2 = .629$). Post hoc analysis with a Bonferroni adjustment revealed that the training time with the smartphone ($M = 34, SD = 7s$) was statistically significantly smaller than the time with the Kinect ($M = 69, SD = 20s$), ($p < .0005$), and smaller than the smartwatch ($M = 58, SD = 16s$), ($p = .002$). However, there was no significant difference between the training time with the Kinect and the smartwatch ($p = .358$).

Let us now analyze the time required to complete gesture recognition. The following analysis is carried out over the duration of the gesture, i.e. the time measured from the moment the client starts collecting data to the moment it stops the collection. For example, the duration of a gesture using the Phone is from the moment when the finger presses the screen to the moment when the user pulls the finger up from the screen (see Figure 3.4b). Table 5.8 summarizes some statistics on the available data. As the reader may notice, gestures last longer with the Kinect and the Watch, while the Phone is the faster technology for the users. The user posture seems not to influence much in the gesture duration. There is a significant variability among the duration of the different proposed gestures, being the

CHAPTER 5. USER STUDY

Figure 5.7: Average duration (ms) of each gesture with the Kinect, for all users. The error bars are ± 1 average absolute deviation

directional the faster ones (down, forward, backward), followed by the ‘simplest’ letters (r and l). Clockwise circle, b, h or w drawing takes longer, being the acronym tv the longest gesture of the series. Figure 5.7 shows the average gesture duration for the Kinect, the same pattern is repeated for the rest of technologies.

Table 5.8: Gesture duration statistics

Technology	Posture	Average (ms)	Average absolute deviation	Min. duration	Max. duration
Kinect	Standing	2012.3	408.34	1195.6 (forward)	3255.5 (tv)
	Seated	2063.2	471.98	1250.9 (forward)	3200.8 (tv)
	Combined	2037.75	440.16	1223.25 (forward)	3228.15 (tv)
Phone	Standing	1381.88	364.41	962.83 (backward)	1911.2 (tv)
	Seated	1369.41	295.68	961.25 (backward)	2114 (tv)
	Combined	1375.63	330.048	962.04 (backward)	2012.6 (tv)
Watch	Standing	1992.03	304.59	1444.7 (backward)	2817.5 (tv)
	Seated	1973.73	262.25	1419.9 (forward)	2907.5 (tv)
	Combined	1982.87	283.42	1448.5 (backward)	2862.5 (tv)

For each gesture and every technology, paired-sample t-tests shows that it is not possible to reject the null hypothesis of zero difference for the average duration of the gestures when performed seated or standing at 5% significance level, thus for a given user, the gesture duration is to be similar, independent of the posture.

A Shapiro-Wilk test shows that it is not possible to reject the null hypothesis that the average duration data of any gesture with any technology come from a normally distributed population (with 4 exceptions out of 30 groups of data: ‘down’ ($p = .014$) and ‘h’ ($p = .004$) with the Kinect, and ‘clockwise circle’ ($p = .038$) and ‘b’ ($p = .027$) with the Phone). Mauchly’s test on sphericity for technology does not reject the null hypothesis. Then, assuming normality and sphericity, a two-factor repeated measures ANOVA test is performed on the average duration of the gestures independent of the postures to analyze the impact of

		t_{pre}/t_{pro} (%)	t_m/t_{pro} (%)	t_f/t_{pro} (%)	t_{dtw} (ms)	t_{pro} (ms)	t_{wf} (ms)	t_{pro}/t_{wf} (%)
Kinect	Stand.	24.71	56.72	18.57	13.23	212.70	2225.00	9.51
	Seated	20.07	46.86	33.08	15.89	314.30	2377.50	13.16
Phone	Stand.	2.52	97.48	0	21.50	108.40	1490.28	6.85
	Seated	2.52	97.48	0	21.42	107.30	1476.71	6.75
Watch	Stand.	5.75	94.25	0	2.41	14.11	2006.14	0.66
	Seated	5.47	94.53	0	2.34	13.90	1987.63	0.64

Table 5.9: Time and proportions of the phases in gesture input workflow

the technologies on the gesture duration. ANOVA test rejects the null hypothesis of equal duration of the gestures for the three technologies ($F(2, 22) = 65.72, p \ll$) at 5% significance level. A Bonferroni post-hoc test shows that gestures with the Phone are significantly shorter ($p \ll$) than the Kinect or the Watch gestures, which are quite similar between them.

Figure 5.8 shows the average duration of all gestures per user with the three technologies. In the case of the Kinect, the figure enables to distinguish users that perform gestures fast (average time under 1800ms, U6 and U7) and others that perform gestures slowly (average time over 2300ms, U1, U5 and U8). In the Phone case, U4 is the fastest user (1030ms), while U8 performed the slowliest (1927ms). As for the Watch, most of the users have the average performance time around 2000ms, with the maximum of 2242ms from U8 and minimum of 1722ms from U6. Comparing the data of each user over the technologies, the plots in Figure 5.8 agree with the above analysis that the Phone is the faster technology, then the Kinect and the Watch.

Finally, let us analyze the delay introduced by the gesture recognition process in the overall gesture input for each technology.

Figure 5.9 shows the average duration of the gestures for the Kinect, the Phone and the Watch in standing and seated postures, together with the average processing times for the different phases of the recognition process:

- t_{pre} , or preprocessing time, which is the time interval spent on preprocessing the gesture templates and the test sample.
- t_{dtw} , which is the time interval invested in completing a single Dynamic Time Warping (DTW) matching. The duration of this time interval depends on gesture length (number of samples), which depends on sampling rate and gesture duration.
- t_r , recognition time, which is the time spent on recognizing a single gesture sample. It includes the preprocessing time for the test sample and templates, and depends on n - the number of templates: t_r is slightly larger than $t_{pre} + n \times t_{dtw}$ due to extra calculations such as the procedure of making decisions on the recognition result.

Figure 5.8: Average duration (ms) of all gestures with the Kinect, the Phone and the Watch for each user. The error bars are ± 1 average absolute deviation

- t_g , gross time is a Kinect-specific processing time. It refers to the time from when the

Figure 5.9: Time analysis on the phases in the gesture performance-recognition workflow. K = Kinect, P = Phone, W = Watch, Po1 = standing, Po2 = seated, Phase 1-4 are gesture performance, preprocessing, recognition, wireless transmission, respectively

Kinect client sends out the gesture sample data to the server (in which the recognition process is performed) until when it receives the result of the recognition.

- t_f , or flying time, is another Kinect-specific processing time. It is an estimation of the delay introduced by the wireless communication ($t_f = t_g - t_r$). The actual value should be a little smaller than this result because the calculation in the server side includes other operations in addition to recognizing gestures.

The applications of the Phone and the Watch do not have “gross time” and “flying time” because the gestures are directly recognized in the client side.

The values measured are highly dependent on the deployment and the implementation so they cannot be compared directly on time efficiency between different technologies. However, from these data we can get an intuitive view on how each stage in the gesture performance and recognition workflow contributes to one time cycle with the implemented interaction system.

There are some implementation differences in gesture recognition between the Kinect and the Phone / the Watch, especially the preprocessing part. Before the DTW matching, the normalized length of gesture samples is the length of the test sample in the Phone and the Watch while in the Kinect it is 20. In the Kinect, there are additional coordinate transformations and sample filtering to remove noise.

The DTW time is much smaller in the case of the Watch because the gesture samples are shorter. The sampling rate of the accelerometer of the Watch was set to 25Hz while the sampling rate on the Phone was a little smaller than 100Hz. The gesture recognizers for the Watch gesture data and the Phone data have the same implementation and both run on the same smartphone.

Table 5.9 lists the weights of the phases of preprocessing t_{pre} , DTW matching time $t_m = t_r - t_{pre}$, data transmission t_f in processing time t_{pro} , and the proportion of processing

time in the whole gesture input workflow t_{wf} . The average duration of a single gesture (performance and recognition) lasts around two seconds (note that it does not include the interpretation of the gesture chains by the gesture sequence analyzer and the response of the smart space). It can be seen that the most time consuming task is the drawing of gestures in the air, which in the case of the Watch represents in practice more than 99% of the interaction duration. The wireless communication time becomes relevant for the Kinect device that is further located in the infrastructure (the computer where server program ran and the Kinect for recognition test in standing posture were in Hall while the Kinect for recognition test in seated posture was in Meeting room. See Figure 5.1). With these data, it can be confirmed that the recognition system is not introducing a relevant delay, although it has to be minimized as it can be triplicated when a 3-item grammar is built.

5.3.3 Evaluation of the grammar use for smart space control

This section aims at gathering relevant results on the effectiveness and efficiency of using the proposed grammar-based interaction, which requires the users to concatenate several gestures to interact with the space.

Error analysis when commanding the space with the proposed grammar

There are 72 (12 participants $\times$ 6 tasks) task instances in the user journey for each technology. Four main types of errors were observed: *gesture collection errors* as introduced before, *gesture recognition errors*, *interface confusion errors* and *grammar and vocabulary errors*. The third error was due to the design of the undo button that would delete the last accepted gesture element but many participants thought it deleted the last performed gesture. This made confusion in the gesture chain composition. The grammar and vocabulary errors happened when the participants forgot or mistook the grammar rules or gesture elements of the gesture chains.

As shown in Table 5.10 (note that the error types are not mutually exclusive in one task instance), most of the task instances with errors were interfered by the gesture recognition errors. Gesture collection with the Kinect has much room for improvement and the undo button caused confusion among all the three technologies.

Some participants forgot or mistook the gesture element in gesture chains. For example, they did “w→forward” instead of “w→backward” to check weather forecast. The action gesture was omitted by some participants in the three-element gesture chain. “h→r” was thought as the command to set Hues in red instead of “h→forward→r”, for instance. Some just recalled the first gesture and could not remember the rest of the chain.

The grammar and vocabulary errors grouped by task type and technologies are shown in Table 5.11. Most of the grammar and vocabulary errors happened on the three-element gesture commands of Hue color setting and light mode setting. The “play movie” task also had a relative high error number. Few errors were found in the other three tasks.

Table 5.10: Task errors (“TI” means “task instance”)

	Kinect	Phone	Watch
No. TIs	72	72	72
TIs with errors	35	34	37
TIs with gesture collection errors	18	0	5
TIs with gesture recognition errors	26	32	34
TIs with interface confusion errors	10	6	7
TIs with grammar and vocabulary errors	11	9	11

Table 5.11: TIs with grammar and vocabulary errors distributed by tasks and technologies

	Kinect	Phone	Watch	Total
Weather	0	1	1	2
Blinds	0	0	0	0
Hues	4	2	3	9
Lights	4	4	5	13
Robot	1	0	1	2
TV	2	2	1	5
Total	11	9	11	31

The distribution of the number of grammar and vocabulary errors on task type can be partially explained by the gesture agreement analysis, which will be introduced later (See “gesture agreement” part and Figure 5.11). The blind and robot tasks had gestures with high agreement score and the gesture chain proposed by the participants coincided with the predefined gesture chain. The rest of the tasks in the user journey had gestures with low agreement score and the predefined commands diverted to their proposals. The weather forecast checking was an exception. Despite of low agreement score, it had few errors, probably because it was the first task to do and the participants had a fresh memory, or the participants’ proposals compromised with the predefined command. The length and the complexity of the command of the Hues and Lights could have additionally caused the high error numbers.

There are similar error numbers among the three technologies.

Task time and service success

Task time was measured from the moment the facilitator started to tell the participants the task name to the moment when the service was triggered (the blind starts dropping, the Hue turns on etc.). Due to the task errors, in particular the bad gesture recognition performance, which made participants repeat some gesture several times and even retrain the gesture, some tasks took an extremely long time, with a few of them reaching 3 or 4 minutes. The plot in Figure 5.10 though, only shows the average time of the tasks without errors.

Figure 5.10: The average time of tasks without errors

The longest average time was from the light-mode setting task with the Kinect (22s) while the shortest was from the robot task with the Phone (7s). The large standard deviations of some tasks, shown with error bars, indicate the high variation of user performances.

The time of successful tasks is mainly composed of the time of recall of the gesture commands, the gesture performance and the service response time. Despite of the trials before the formal user journey test, the new gesture grammar and gesture interface seemed to have lengthened the task time, although further study is necessary to confirm this point.

All the tasks using the Phone had the least average time for completion while the tasks with the Kinect and the Watch took a similar time. The order of the technologies was counterbalanced to avoid the learning effect of e.g. gesture commands, gesture interaction in general. Then the tasks with different technologies should have a similar amount of time of recall of gesture commands and service response. The Phone is more efficient than the other two from this point of view.

The tasks with two-gesture commands (weather, blinds and robot) had a smaller task duration than the ones with three-gesture commands (Hues and lights). The relatively long TV task time, although it has two gestures in the command, was probably because of the fatigue introduced by the fixed task order (the TV task was always the last one).

In very few cases, the home automation server did not respond to gesture commands or responded to the second trial. The response time of the TV service, estimated from video annotation, was a little more than 3 seconds and the other five were less than 2 seconds.

Evaluation of user interface

With a 7-point scale (1-not useful at all, 7-extremely useful), the participants rated the usefulness of different elements in the interface, i.e. the sweet spot ($M = 5.6, SD = 1.3$), the progress bar ($M = 4.9, SD = 2.3$) and the semaphore ($M = 4.0, SD = 1.7$) in the Kinect interface and the vibration feedback ($M = 4.5, SD = 1.7$) of the Watch.

Nine of the twelve participants thought the length of vibration of the Watch was long enough to indicate the start of gesture recording.

General comments on the applications, especially on the user interface, based on the three technologies. Regarding the Kinect, two participants complained about the undo button which comes to be annoying after doing some gestures. These participants referred that it was easy to activate the button by mistake. One participant said he became very tired after the test with the Kinect. However, one participant said the application worked quite well and the visual feedback was very useful for the user.

With the Phone, one participant gave positive comments, saying that he liked the interaction format and he would use it at home, because it worked well and gave feedback immediately. Two participants thought the command composition with continuous concatenated gestures could be faster than the current manner. One participant said the recognition did not always work as expected while another stated that leaning back on the sofa could have complicated the recognition due to the possible variation of the Phone orientation, when compared with the straight back posture. This participant also recommended adding a reasoning functionality that could be aware when the recognized gesture was not the desired one, when the gesture was not recognized constantly and when the users were repeatedly doing gesture and then deleting it.

When using the Watch, it took time to understand the way to do gestures with it, one participant commented, and the vibration feedback was very short. Other comments are that there was unintended gesture input and it was a dilemma wearing the watch on the left wrist (normal practice) or right wrist (to do the gesture more comfortably). It would be great to do gestures in a consecutive way. Another interesting observation was that one participant found that, on the contrary to the training section, in the user journey test the user tended to move the watch to read the feedback on the screen, which might have had the watch collect gesture signals different from the ones in the training and degraded the recognition accuracy. One participant gave positive comment saying he liked using the watch.

Suggestions for interface improvement. In the case of the Kinect, with respect to the UI components, one participant liked the progress bar and others thought the silhouette and semaphore were not necessary. In addition, the green dots should be quicker to follow the hand on the screen. As for the functionality, some participants complained about the failure of the hand detection; additionally, network communication problems occurred occasionally and there were also some bad gesture recognition episodes. The ‘forward’ and ‘backward’ gestures caused a lot of hand tracking failures so one participant suggested that the interface should inform the users to place the hand out of the body area to relieve this error. Two participants mentioned the undo button which confused the users, not sure whether a wrong gesture could be overwritten or it had to be deleted. One interesting proposal was gesture suggestion, similar to autocomplete, i.e. predicting and showing the possible gestures to select to speed up the gesture command composition. The training functionality should be integrated into the gesture control part, one participant suggested, so that once a gesture was not recognized constantly, it could be retrained on the fly.

When using the Phone, there were recognition mistakes in the test and one participant

CHAPTER 5. USER STUDY

thought that the accuracy might improve with the continuous usage while another recommended that the application provides a list of possible gestures to manually select when a gesture is not recognized persistently after several trials. The icon of the button to restart the gesture chain composition was not intuitive and the functionality corresponding to the gesture as well as the gesture name should be displayed. One participant suggested adding the visual or haptic feedback when the button was pressed to do gestures. Visual feedback for grammar error notification was preferred to vibration by one participant. Audio feedback could be included if possible, another one recommended.

Finally, regarding the Watch, one participant thought that the interaction system was a bit slow for standard users and a possible improvement could be performing gesture components continuously instead of separately. Another did not feel the vibration strong enough and suggested that the system should provide the option of personalizing the vibration. Two participants mentioned the avoidance of similar gestures like ‘h’ and ‘b’ to facilitate recognition. Some emphasized using more visual feedback on the Watch and including sound feedback and clearer vibration patterns. The Phone was necessary for localization and gesture recognition purpose but it would be better if the Watch could give feedback independently and users could just leave the Phone in the pocket instead of holding on hands, one participant commented.

Gesture agreement and other issues affecting grammar acceptance

Gesture agreement. Before the user journey test, the participants were asked to propose what gestures they would use to represent an operation for each of nine tasks in a questionnaire (see Appendix C.3.1). The gestures to represent resources and parameters were given as initial letters or acronyms. These tasks include raising temperature, checking weather forecast, pulling down blinds, setting Hue color in red, setting lights in TV mode, making robot approach to the user, playing music, closing movie and playing on a tablet the movie in TV. Agreement scores as proposed in [104] were calculated and shown in Figure 5.11. The maximum possible score is 1, meaning that all participants came up with the same gesture. The lower the score, the diverse the proposed gestures are. The operations related with clear spatial movements seem to have a stronger gestural agreement, such as the tasks with the three top scores. According to the user study carried out in [43], users feel gestural commands natural especially for simple commands with spatial association. It would be interesting to further investigate the correlation between agreement scores and naturalness of gestures.

Social acceptability for gesture-based interaction. In addition to features like high gesture recognition rate, natural gesture performance, and intuitive gesture sets, the social acceptability is also a key factor with respect to the adoptability of gesture-based interface. After the user journey test on all the three technologies, the participants rated the social acceptability of the gestures in the test on a 10-point Likert scales on two factors: locations and audiences. The questionnaire was inspired by [7] and [82].

GESTURE AGREEMENT AND OTHER ISSUES AFFECTING GRAMMAR ACCEPTANCE

Figure 5.11: Agreement for each task in the questionnaire, sorted in descending order

Table 5.12: p values for pair-wise comparison of locations. p values have been adjusted using Bonferroni's correction and $p < 0.05$ is significant (shown with figures in bold)

	Home	Pavement	Driving	Passenger	Pub	Workplace
Home		.040	.000	.002	.003	1.000
Pavement			.285	1.000	1.000	1.000
Driving				1.000	1.000	.001
Passenger					1.000	.246
Pub						.285
Workplace						

The mean social acceptability scores indicate that the gestures are acceptable (mean score is larger than 6) in home and workplace, in front of all the audiences but strangers, as shown in Figure 5.12. The ratings on the social context of home, alone, in front of partner and family had relatively smaller standard deviation. For other social contexts, the participants had more divergent opinions.

Friedman test revealed there was statistically significant difference between the ratings on social acceptability among different locations ($\chi^2(2) = 40.317, p < .0005$) and across audiences ($\chi^2(2) = 34.254, p < .0005$). The results of pair-wise comparison are shown in Table 5.12 and 5.13. With regard to locations, the score in home is statistically significantly different from all other locations except workplace, and another significant difference is between driving and workplace. As for audiences, the score on social acceptability when the audience is not familiar is significantly different from the score when the audience is a partner, or the family, or when gestures are performed when the user is alone.

These significant differences indicate that the types of locations and audiences play an important role in social acceptance of gesture-based interaction. Locations that provide more privacy like home and workplace, and performing the gesture when alone or in front of audiences with which the performers are familiar like partner, family members seems to

Figure 5.12: Average ratings of gesture acceptability by location and audiences. Error bars show one standard deviation

give more comfort to use gestures. These findings are similar to the results in [82]. However, in [82], almost every type of location/audience is different from all other locations/audiences. Here there is no significant difference between the public locations and between familiar audiences. This may be due to culture difference, the year of the test and the design of the tests.

Perceived exertion of the user journey test. The Annex C of the ISO-9241-9 (Borg Scale) was used to measure the physical demand. The average physical demand is moderate in the arm ($M = 2.88, SD = 1.25$), light in the shoulder ($M = 1.83, SD = 1.61$) and very light in the neck ($M = 0.75, SD = 0.81$).

Users were also asked about their preferences regarding the use of small or capital letters, which seems to be a non-relevant issue, as the twelve participants showed equal preferences with respect to small and capital options. Eight of twelve participants claimed that they preferred defining the interaction gestures themselves to “learning” a set of predefined ges-

GESTURE AGREEMENT AND OTHER ISSUES AFFECTING GRAMMAR ACCEPTANCE

Table 5.13: p values for pair-wise comparison of audiences. p values have been adjusted using Bonferroni’s correction and $p < 0.05$ is significant (shown with figures in bold)

	Alone	Partner	Friends	Colleagues	Stranger	Family
Alone		1.000	1.000	1.000	.003	1.000
Partner			1.000	1.000	.013	1.000
Friends				1.000	.181	1.000
Colleagues					.246	1.000
Strangers						.011
Family						

tures. Related results are found in [58] where 4 out of 17 users prefer a given set, 3 prefer personalized set and 6 would like a predefined set with possibility to adjust. The rest 4 users did not give answers.

To evaluate the acceptance of the grammar-based interaction, six Likert scale questions about gesturing, the grammar and application design were posed:

1. “It is cumbersome to gesture in the air”
2. “The grammar is easy to learn”
3. “The grammar can be effective to command the environment”
4. “I would like to perform gestures of a gesture chain in one go instead of doing one gesture, waiting feedback and then doing another gesture”
5. “I like that the service is location-aware, being effective over the objects in the space where the user is”
6. “I would like an additional step to confirm or negate the command after performing the gesture chain”

As shown in Figure 5.13, a majority of participants thought the grammar was easy to learn (58% strongly agreed or agreed), the grammar could be effective to command the environment (50%), and would perform connected gestures (50%), and had a great agreement on location-aware service (83%) and thought that a confirmation step was not necessary (50%). However, there were divergent opinions on whether gesturing in the air was cumbersome.

Open questions on alternative grammar formats and solution to letter conflicts. The participants were encouraged to propose other grammar formats. One proposed to reduce the number of gestures in the commands with more context information such as user’s location. Three participants suggested selecting objects by pointing or through combining pointing and initial letters (or acronyms). Someone stated he would avoid commands with three or more gestures. Another mentioned using different gestures for each specific action of each specific object but she also said this would be very demanding to memory.

Figure 5.13: Results of the six Likert scale questions (Scale 1 - perspective to gestures, Scale 2 - learnability of the grammar, Scale 3 - effectiveness of the grammar, Scale 4 - performance of multi-gesture, Scale 5 - the location-aware feature, Scale 6 - the confirmation step)

The participants provided many proposals for the situation where there were conflicts of letters, for example, there were both robots and radios in the smart space and they would both be assigned ‘r’. Appending more letters to the initial letter (or acronym) is the most proposed solution. This is like the two-letter province acronym on the license plates to distinguish provinces with the same initial letter. Similar proposed solutions were using capital and small letters, appending numbers and using full names. Another thought was to combine the initial letter or acronym with pointing. One participant recommended the interested object could be inferred with context such as the users’ locations and/or their orientations. If there was still ambiguity, the system could ask the user to disambiguate. Another interesting opinion was looking for synonyms or nicknames of objects to avoid the conflict.

5.3.4 Analysis of technology preferences: technology comparison, usability, user experience

Comparison against voice-based interaction

As previously said, by the end of the test, participants were asked to perform some tasks with a voice-based interaction system.

Task errors. Two participants mistook the verb in the command to play music. The application did not recognize well one word in the command by one participant to pull the blinds down. Other types of errors were due to low sound volume of the command or not speaking toward the microphone. Alternatively, when the blind was lowering down, the motor made the environment a little noisy and failed recognition was observed if the participant was not near the Kinect. It was also observed that the pause between words in the command also led to recognition failure. Therefore, the voice-based interaction also had the problems of data collection, recognition and grammar but the error number is significantly smaller than that of gesture-based tasks. The participants committed more

COMPARISON AGAINST VOICE-BASED INTERACTION

Figure 5.14: Comparison of average time of task without errors

grammar errors when using the gesture-based application than the voice-based (31 task instances with grammar errors out of 216 task instances vs. 2 out of 48).

Task time. The task time of blinds, lights and TV tasks in voice-based interaction is compared with the results in gesture-based interaction, presented in Figure 5.14 and they are all from results of tasks without errors. The average time of tasks without errors was smaller when using voice-based application than when using any of the three gesture-based applications.

The task time with voice has the same definition as the time with gestures. They are mainly composed of the recall time of the command, the speaking time and the service response. Given the same home automation system, the service response time should not have been affected by the type of technology. In the video annotation, we observed much shorter time spent on the command recall. The average task time of voice-based interaction was the smallest (around 5s). All the vocal commands had three words but instead of speaking/inputting them separately like the gesture-based interaction, they were connected words and no explicit delimiters are necessary to start and stop the voice input. Therefore, the vocal input was more efficient in this comparison. There was smaller variability of task time using voice among different participants as shown by the error bars, which indicate the standard deviation.

General comments on voice-based interaction. All in all, most of the participants gave positive feedback on the voice-based interaction, saying that “I was surprised that it worked so well”, “the voice control is the best but I liked the watch too”, that it worked fine, that users had comfortable control and that it was easier, more natural and more robust than gesture-based interaction. One participant mentioned that the voice-based interaction might have a problem in multi-user scenarios. Another said that in an unconstrained situation he would prefer voice that is more natural while in some environments, he could prefer gestures that are not too complex and weird (just wrist rotations or minor hand translations) but in general, he found the gestures much more inconvenient.

Figure 5.15: Box plots of the results of the Likert scale ratings on the training section

Subjective ratings and comments on technology acceptance factors for training tasks

After completing the training task with each technology, the participants rated with 5-point Likert scale on five aspects of the training task: the naturalness to start and finish a gesture, the boredom of the training, the easiness of usage of the training application, the efforts required and the feeling of confusion during the interaction. The results are shown with box plots in Figure 5.15.

A Friedman test was run to determine if there were differences in the ratings among the

SUBJECTIVE RATINGS AND COMMENTS ON TECHNOLOGY ACCEPTANCE FACTORS FOR TRAINING TASKS

three technologies for each of the five aspects. Pairwise comparisons were performed with a Bonferroni correction for multiple comparisons. Statistical significance was accepted at $p < .0167$ level.

The difference of the rating on the naturalness to start and finish a gesture was statistically significant for the three technologies, $\chi^2(2) = 10.067, p = .007$. Post hoc analysis revealed statistically significant differences between the ratings for the Phone ($Mdn = 5$) and the Watch ($Mdn = 4.00$) ($p = .007$). There were no significant differences between the Kinect and the Phone ($p = .034$), and the Kinect and the Watch ($p = .783$). Significant difference was also found in the ratings on the easiness of usage of the training application $\chi^2(2) = 12.400, p = .002$. Post hoc analysis indicated significant rating difference between the Phone ($Mdn = 5.00$) and the Kinect ($Mdn = 4.00$) ($p = .011$), and the Phone and the Watch ($Mdn = 4.00$) ($p = .008$). However, there was no significant difference between the Kinect and the Watch ($p = .919$).

There was no statistically significant difference among the ratings on the boredom of the training, the efforts required during the training and the feeling of confusion during the interaction.

User feedback on the training interface. For the Kinect, most participants gave positive comments on the feedback provided by the training interface. They used the phrases “easy to use and understand”, “intuitive”, “clear”, “adequate” and “well designed” to describe the interface. The sweet spot video feedback, the back button, red and green dots were specially considered as well designed. However, it was also indicated that “feedback with green dots may affect the speed to make a gesture” since there was delay when they followed the hand and the icons representing “forward” and “backward” gestures were confusing.

In the case of the Phone, most of the participants mentioned that the training was “smooth” and “showed a good response to mistake amendment” and the interface is “nice”, “easy to use and understand” and “intuitive”. It was also referred that the big button to do gestures with by pressing on the screen can be confusing and one participant tried to draw gestures on the button. Another participant noticed there was no feedback on whether the phone is tracking the gestures because the button was static (the button did not change visually, which is to be improved). “I lost the control over the button” when performing “tv” gesture, one participant described, because this gesture had two letters and the letter ‘t’ had two separate movements.

For the Watch, it was mentioned by several participants that the vibrations of the smartwatch were a little short. Retrieving feedback from the smartwatch on the screen of the smartphone was thought “weird” (the names of the gestures to train were only shown on the smartphone). However, “it is better than reading the display of the watch every time I do a movement” and “visual feedback on the phone would have been much clearer and comfortable”. Another participant considered using two devices a little annoying and suggested the watch show the gesture name itself. In terms of turning wrist to start gestures, some participants found it weird and pointed out that gesture recording was unexpectedly started. One participant complained about the abrupt termination of some gestures in uncomfortable positions; it was required that the hand was kept static there. One participant thought

CHAPTER 5. USER STUDY

that the users had to wear the watch tightly to control well the movements, which is a little uncomfortable. Handedness was also mentioned. One participant tried to wear the watch on the left wrist at first but after several trials, he changed to the right hand because “writing letters was uncomfortable with the left hand”. Some participants gave positive comments, saying the interface was intuitive once it is understood, “one practice is enough”, “vibration feedback is very useful” and the training was “smooth”.

At the end of the questionnaire, the participants were asked to rank the three technologies by preference on doing gestures with. Nine of the twelve chose the smartphone as their favorite and the three that did not choose the smartphone rated the Kinect as the favorite. On the other hand, seven chose the smartwatch as the least favorite, four chose the Kinect and one chose the smartphone.

Suggestions on improving the training. One participant stated he liked that the training only required a single repetition while the other said that more repetitions would help, so the user could memorize your performance manner. One participant thought that the feedback of the training with the Watch should be improved and another one did not like the ‘tv’ gesture and suggested deleting it.

Gesture order. Seven out of the twelve participants considered that it would be better for the training gestures to appear in random order instead of fixed order.

Analysis on usability and user experience for the interaction system

To evaluate usability, the well-known System Usability Scale [16] is applied. The participants have been asked to rate ten issues of the system regarding system complexity, frequency of use or learnability, in 1(strongly disagree)-5(strongly agree) Likert scales. SUS value (0-100) is computed from these answers.

The SUS scores are shown in box plot in Figure 5.16. The assumptions of repeated measures ANOVA were violated because there were outliers in the SUS scores of the Phone and Voice, and the scores of Voice violated the normality assumption. A Friedman test indicated there were statistically significant differences between the SUS scores on the four technologies ($\chi^2(2) = 23.822, p < .0005$). Pairwise comparisons were performed and the statistical significance was accepted at $p < .0083$ level, with the Bonferroni adjustment. Wilcoxon signed-rank test revealed that the score on voice-based application was significantly higher than for the Kinect ($z = 3.062, p = .002$), the Phone ($z = 2.634, p = .008$) and the Watch ($z = 3.061, p = .002$). When only comparing the scores on the Phone and the Kinect, the Wilcoxon signed-rank test indicated that the score for the Phone was significantly higher than for the Kinect ($z = 2.096, p = .036$). No statistically significant differences were found between the scores of the Watch and the Phone ($p = .070$), and the Watch and the Kinect ($p = .373$).

With respect to user experience (UX), the questionnaire proposed in [60] has been applied. The questionnaire handles 6 different factors evaluating attractiveness (overall impression of the system), perspicuity (easiness to get familiar), efficiency (without unnecessary effort to solve user’s tasks), dependability (feeling in control), stimulation (excitement and motiva-

OVERALL PREFERENCE ANALYSIS FOR GESTURE-BASED INTERACTION TECHNOLOGIES

Figure 5.16: The SUS scores of the four technologies

tion) and novelty (innovation) through 26 questions. The User Experience Questionnaire (UEQ) does not produce an overall score for user experience, since it makes no sense to build such an overall score that cannot be interpreted properly.

The results of the UEQ are described in Figure 5.17. Friedman test showed there was no statistically significant difference between the user experience of the three gesture-based technologies in factors: attractiveness ($p = .127$), perspicuity ($p = .089$), dependability ($p = .517$), stimulation ($p = .311$) and novelty ($p = .266$), *except* efficiency ($p = .034$). A pairwise comparison with a Bonferroni correction for multiple comparisons (statistical significance was accepted at $p < .0167$ level) revealed the efficiency score of the Phone was significantly higher than that of the Kinect ($p = .014$). No significant difference was found between the efficiency scores of the Watch and the Phone ($p = .066$), and the Watch and the Kinect ($p = .540$).

The benchmark dataset of the UEQ contains data from studies concerning products of business software, web pages, web shops and social networks etc. A direct comparison between the UEQ result of the gesture-based systems and the benchmark might not be reasonable. However, the results can serve as a benchmark for next design iterations.

Overall preference analysis for gesture-based interaction technologies

Rankings of preference of the three technologies, the Kinect, the Phone and the Watch, for the six tasks in the user journey test revealed that more than half of the participants rated the Phone as the favorite for all the tasks. In addition, more than half of the participants rated the Kinect as the least favorite for all the tasks, except for the task of “checking weather” (half of them thought the Kinect was the least favorite and the other half thought the Watch was).

Eight out of the twelve participants ranked the Phone as the favorite in terms of gesture recognition. Moreover, eight of twelve thought the Kinect as the least favorite.

Supposing that all the three technologies are available, half or more than half of the

Figure 5.17: Results of the UEQ, with +3 the most positive and -3 the most negative

participants would prefer to use the Phone when at home or in the workplace, and the Watch when in the street, while driving, on public transportation means and when at the pub or restaurant. More than half of them rated the Kinect as the last preferred method in all the above locations.

As shown in Figure 5.18, a majority of the participants preferred voice-based interaction to gesture-based (64%), Kinect-based (83%), Phone-based (67%) and Watch-based (83%).

Determining factors of preference for voice and gestures. Noise level in the environment, level of fatigue, naturalness of performance, ease and comfort of usage, recognition accuracy,

Figure 5.18: Technology preference questions in Likert scale format (G = Gesture, V = Voice, K = Kinect, P = Phone, W = Watch)

comfort or privacy concerns are among the mentioned factors. Some participants said that voice was easier and straight, and it was easier to learn natural language than the gesture grammar.

5.4 Design recommendations

The design procedure of an interaction system can be compared to an optimization problem that feeds an incremental process. Before proposing a design solution, we have some design objectives in mind, such as user experience, efficiency, learnability, etc., and constraints like limited resources related to technologies and platforms and human factors. Then we propose a solution to optimize the design objectives given the constraints. The evaluation with user study measures the design. At the same time, the evaluation reveals new design constraints/considerations or adjusts the weight of each constraint, and some design recommendations are summarized for next design iterations. From the experience of the design and evaluation of the proposed interaction system, some design decisions to take when designing a novel gesture-based interaction system are following discussed.

The design of single gesture input procedure Results show that the Phone is the favorite technology to do gestures with, and it is natural and easy to use. Comparing designs and test results for the three technologies, some design recommendations on gesture delimiters would be:

Proactive control. The users decide when to start and end a gesture input. This agrees with the idea that interaction should be personalized and adapted to each user as mentioned in [93].

Effortless and low precision action. The finger pressing and releasing of the Phone is an effortless and low precision action. The velocity threshold of the Kinect and the wrist rotation of the Watch require precision, leading to FSG (failed to start gesture) errors, and the users had to adapt to the system, which is to be avoided.

CHAPTER 5. USER STUDY

Low false positives. The design should avoid unintended inputs. The design for the Kinect and the Watch caused many unintended gesture inputs and this made the participants feel loss of control.

Smooth transition between delimiters and gesturing. The wrist rotation design was not naturally connected to the hand gesture and at the beginning of the test, some participants hesitated after turning down the wrist before gesturing. Because the wrist rotation is not comfortable when the arm is completely stretched, some participants turned the wrist at a comfortable position (near the body for example) and then moved the hand to the initial position of the gesture to do the gesture, incorporating extra data in gesture samples.

Crisp detection and immediate feedback. The system should be sensitive to the delimiter action and give feedback immediately on whether it is successful. The delay of the static-hand detection on the Kinect and the Watch forced the users to keep the hand in an uncomfortable position at the end of the gesture for a while and caused fatigue.

In addition, the sensing by the accelerometer is more robust and stable than the Kinect sensor, which showed limitations in strong light conditions, field of view, line-of-sight and hand postures (the Kinect tends to have difficulty to detect the hand when the palm is parallel to the ground).

Gesture recognition From the analysis of confusion matrices, we have seen that the definition of gesture sets has an influence on the recognition accuracy. In addition, the impact of the quality of the captured gesture data, the variation in the gesture performances among different users and the hand tracking system was also observed. The comparison between offline and online evaluations reveals some determining factors that condition the accuracy performance. Here are some design considerations for the gesture recognition approach.

Gesture set with members easy to distinguish. Some gestures were quite similar and confused a lot, so these should be avoided from the vocabulary if possible.

Consistent gesture recording. The mismatch between the gesture templates and the samples is the fundamental reason leading to wrong gesture recognition. Specifically, the observed causes include gesture segmentation errors that end up with potentially inconsistent gesture recording (for example, due to the IBG (imprecise gesture boundaries) errors with the Kinect, there is no guarantee that the recorded signals of the same gesture have the same boundaries). Bad training that introduces incomplete and distorted gesture templates made the gesture recognition performance worse as well.

Robust to variability from gesture performance. Although the online test of gesture recognition was carried out in the same day for each participant, a substantial difference among the iterations for the same user was observed between training and gesture recognition phases, as well as between the repetitions in the gesture recognition phase. For example, the order of the two strokes of the letter ‘t’ could change.

Robust to variability from hand tracking systems. For the Phone and the Watch, some obvious variations of device orientation (the posture of the device body when users grab or wear it to do gestures) were observed when some participants performed the same gesture

in different iterations. Some if not all of the people did not pay attention to the orientation of the device and grab it the same when doing gestures. This orientation variation is also due to trivial behaviors such as the ones made when adjusting the watchband to read the screen during the interaction, or the body posture (such as leaning on the back of the sofa or sitting tall when gesturing). The analysis of results demonstrates that the posture did not have significant influence on recognition rate and gesturing time. However, in the training and gesture recognition test (where the analysis of impact of the posture on the recognition performance was made) of the Phone and the Watch, the participants sat straight back on a puf but in the user journey test, there were recognition problems when some participants leaned on the back of sofa. Therefore, different sitting postures could cause significant gesture signal variation especially for the Phone and the Watch whose gesture samples depend on the posture of the device.

Notice the difference between offline evaluation and real environment test. Compared with the recognition results in the test with one training sample in Chapter 4.3, the current evaluation has much lower accuracy (the Kinect / the Phone: around 77%/76% vs 98%/90%), although the recognition algorithm is essentially the same. Apart from a similar and smaller gesture-vocabulary in this work, the main difference lies in the gesture recording procedure that was strictly controlled in the offline evaluation. Firstly, the participants did gestures in fixed time slots in Chapter 4.3: they prepared the hand position in advance and started gesturing when notified that the application started to record data, and then stopped the hand movement and waited once the gesture finished, until the recording time slot was completed. Second, each gesture was consecutively repeated so that there would be a smaller variability in the gesture performance. In addition, the lighting condition and the position of the gesture with respect to the Kinect camera were suitable for data collection.

The data collection procedure in Chapter 4.3 improved the recognition results thanks to more complete gesture data and less variability between gesture samples for the same gesture. Therefore, the application in a real environment should take into consideration the possible adverse recording conditions and have an effective gesture recording procedure. The gesture recognition algorithm needs to be robust to the variability of the gesture performance.

Grammar A majority of the participants thought the grammar was easy to learn and could be effective to command the environment. In general, the participants would like short gesture commands, and recommended pointing gestures and context information such as location and orientation. This information could help select devices and reduce the number of motion gestures in one command. Pointing and context information could also relieve the conflict of gesture assignments for devices with the same initial letter.

Keep the gestural command simple, short and intuitive. Compared with vocal commands, it took the participants more time to recall the gestural commands. Among the predefined gesture commands, those with fewer gestures and coincident with the gesture proposals of high agreement scores showed better learnability, which was indicated by the grammar errors observed in the user journey test. Keeping the gesture command simple, short and

CHAPTER 5. USER STUDY

associating commands with gestures with a high agreement score will make the system easier to learn, reducing the threshold for technology adoption. Gesture command customization would be necessary for commands with low agreement scores, as also concluded in [85].

Beyond the grammar: accompanying simple gestures with rich contexts. The strategy of having a personalized gesture set might introduce gestures that may be too similar and difficult to differentiate. In addition, the proposed grammar is to organize and control the smart devices but has a dilemma between the description capability and the simplicity. Its analog would be the command line interface, which could be very powerful but not that easy to remember for common users. An alternative strategy could be a more user-adapted system design: making the gesture simple but the interface provides enough contexts to disambiguate different commands even with the same gesture. It would be similar to the touch gestures for smartphones, which are simple and very few but effective with the help of interface contexts, with low physical and mental exertion.

UI *Useful feedback components.* Interface components like the sweet spot, the progress bar on the Kinect and vibration feedback on the Watch were thought to be useful.

Immediate feedback. According to the observation in the user journey test and the comments on user interface in the questionnaire, the participants would prefer immediate feedback (the green dots visualizing the moving hand position in the Kinect application introduced a noticeable delay and the vibration of the Watch was slow sometimes, while the immediate feedback of the Phone was praised).

Clear metaphors. The confusing undo button indicates that the metaphor of the interface components should be clear and coincide with the users' expectation.

Provide options to customize. Although 75% of the participants thought the vibration was long enough, people have different sensibility on these behaviors and an option to personalize the strength and length could be suitable.

Feedback for state changes and multi-mode feedback. To make the user experience immersive it is important to give feedback on every state change. For example, the button the user presses to do gestures with the Phone had no visual and haptic feedback in the design because the designer thought that the users would have little chance to watch the screen when performing gestures and the vibration might mess the gesture's acceleration signal. However, to a beginner it would be important to have a hint about the state of the system. In addition, providing multimodal feedback, as rich as possible (visual, audio and haptic) is another inspiration from the comments of the participants since the users might have different needs (in different social contexts for example) and preferences.

Technology comparison *Selection of enabling technology.* The Phone is the winner when considering most of the indicators. It is efficient, having the shortest gesturing time, training time and task time among the three gesture-based technologies. This is probably due to the low delay of detection of gesture delimiters of the Phone and the familiarity to the Phone interfaces. In the subjective evaluation, it was considered more natural to start and finish a

gesture than the Watch and easier to use in the training application than the other two. A majority of the participants preferred the Phone to do gestures and ranked it as the favorite in terms of gesture recognition and all task types. It had higher SUS (System Usability Scale) score and UEQ (User Experience Questionnaire) score in efficiency factor when compared to the Kinect. When considering locations, the Watch was chosen by the majority as favorite in outdoor social contexts like street, driving, transportation and pub, and the Phone was chosen in home and workplace. Maybe users thought that holding a phone to gesture in public places seemed cumbersome.

Improvements based on the comparison with voice-based interaction. The voice-based application outperformed the gesture-based with less task errors, shorter time, higher SUS score and dominant preference by the participants. As suggested in the evaluation results, especially the comparison with voice-based interaction, the main directions to improve the current gesture-based applications are reducing physical exertion, enhancing naturalness of performance by better design of the gesture input procedure and gesture set, improving the recognition accuracy and exploring a simple but effective grammar.

Other inspirations *Improvements on the gesture input efficiency.* To enhance the efficiency of the gesture command composition, some participants proposed an autocomplete feature, i.e. the system proposes the following possible gestures to select according to the context or the gesture the user has performed. Concatenated gesture performance could also make the command composition faster.

Recovery from wrong recognitions. For repeated wrong gesture detection, a reasoning functionality can be integrated so that the system can deduce that one gesture was detected wrong after several subsequent repetitions and deletions. As a backup option, if after several trials the gesture is still not recognized, the system could propose a dialog for the users to manually select the gesture they intended to do. A quickly accessible retrain would be convenient after several repeated wrong detections during the interaction.

Consideration of handedness. Handedness needs to be considered especially for wearable devices like the Watch since gesturing with the auxiliary hand may cause fatigue and uncomfortableness. Designing simple and effortless gestures for the auxiliary hand or using dominant-hand-friendly wearable devices might relieve this conflict.

5.5 Conclusion

In this Chapter, a user study is carried out to evaluate the performance and user experience of the interaction system designed in Chapter 3, and to conclude some design recommendations for the design of gesture-based interaction systems. The experimental design starts with research questions, which highlight the investigation on a) gesture recognition performance, b) the grammar of gestures and c) the interaction. The user study has a within-group design since it requires a smaller sample size and it is more robust to the impact from individual differences. Twelve participants were recruited to experience a smart space deployed

CHAPTER 5. USER STUDY

in “Experience Lab of Future Spaces” (ExpLab) in UPM. The user study was organized in four stages: Training, Testing of the Recognition System, User Journey and Voice-based Interaction. Counterbalancing, pre-test trials and ample opportunities to take breaks were used to reduce the biases introduced by the learning effect and fatigue. After the user study procedure, data were gathered in form of application logs, recorded videos and questionnaires. Errors occurred in a) the gesture input procedure, b) the gesture recognition, c) user tasks with the gesture-based and voice-based technologies, the time spent on i) training, ii) gesturing, iii) gesture recognition and iv) user tasks are described and analyzed considering the independent variables such as technology, postures, gesture types, and user tasks. In addition to objective evaluations, subjective assessments from the participants have also been gathered, mainly from questionnaires, but also from thinking aloud and observation from video annotation. Many aspects were evaluated ranging from the gesture vocabulary (preference to small or capital letters, predefined or self-defined gestures, gesture agreement), grammar (easiness to learn, effectiveness, alternative grammars and letter conflict problem) to effort (physical exertion), user interface (UI components, general comments, suggestions on improvement), applications (social acceptability, application design, usability, user experience, general comments on the gesture-based and voice-based applications), training (rating, feedback, training order, suggestions on improvement), comparison between technologies (preference on gesture performance, recognition, when different tasks and locations are given, comparison between gesture-based and voice-based interaction, the determining factors to the preference between gestures and voice). Many problems on the design of the gesture input, the recognition approach, the user interface and the interaction itself were exposed in the user study. The gesture recognition does not introduce significant delay in the gesture input workflow and the task duration with gestures is comparable to the one with voice. Finally, several design recommendations on gesture delimiters, gesture recognition, grammars, user interface and application design are proposed in the discussion of the results.

The training with the three technologies by performing 10 gestures (or more if any gesture was repeated because it was not performed well) was relatively short, with average training time as long as 69s (the Kinect) and as short as 34s (the Phone). The Phone had significantly shorter time than the Kinect and the Watch and this is consistent with the results obtained from the comparison of the gesturing time using the three technologies. In the subjective ratings and comments for the training, the Phone interface was considered more natural than the one for the Watch, easier to use than the one for the Kinect and the Watch, being preferred by the majority of the participants. The main activities in the training were gesture performance and data capture so the main difference between applications on the three technologies were the gesture input procedure design. The positive results of the Phone-based application, from both objective and subjective evaluations, indicate that the design of the gesture input procedure is better than the one for the other two. The design recommendations on gesture delimiters in Chapter 5.4 are based on the comparison between the applications of the three technologies.

Compared with the offline gesture recognition test in Chapter 4, the online recognition test in a quasi-real environment has a more user-friendly gesture capture approach and on-

line running environment, recognizes gestures in real-time instead of with Leave-One-Out Cross Validation, and investigates the impact of the posture factor. The recognition approach did not introduce significant delay in the gesture input workflow and worked with multiple platforms and different types of data. The gesture agreement analysis revealed that the participants had diverse knowledge of mapping between tasks and gestures and it would be reasonable to recognize gestures in the user-dependent manner since it allows personalization of the gesture vocabulary. With one training sample, the recognition approach had a moderate accuracy. A detailed analysis on the performance is provided and leveraged in design recommendations. Although the recognition rates were compared between the three technologies with the same recognition algorithm, the technology factor is in fact a compound of factors like the hand tracking system, the sampling rate, data type and the gesture segmentation approach. When we say that the recognition rates are similar between the three technologies, it is based on the particular design and implementation in this work.

There are always potential biases in the user study. The user study in this work reduces biases by counterbalancing and pre-test trials to control learning effect, opportunities of taking break to reduce the impact of fatigue, as well as other control mechanisms in the test procedure such as providing consistent instructions to the participants or reassuring the participants that the focus of the study is to test the interaction system, not their abilities. Some of the recruited participants might have a little stronger technical background than mass-market users. However, they were easier to organize and fitted at least part of the potential user group (i.e. innovators and early majority). In addition, it was a first study on the prototype and the purpose was to reveal unknown problems. In this sense, the recruited participants serve the study purpose.

The participants gave positive comments on the voice-based application and shared their opinions on the determining factors that drive the preference between gesture-based and voice-based interaction. These evaluations indicate the directions of improvement on the current gesture-based interaction system, as discussed in Chapter 5.4. However, the authors in [88] argue that the speech recognition has problems for home automation because of ambient noise such as the constant talking between residents, sounds from music and movies, and the potential to ungracefully interrupt a conversation in a dinner party. The gesture-based interface does not have the problems in those cases although it also has limitations in home automation scenario, and it would be an important complementary interface to the voice-based interaction.

The server in the architecture of the interaction system works in a platform-independent manner. An enabling technology, no matter it is the Kinect, the smartphone, the smartwatch, or emerging technologies like Leap Motion, Ring ZERO, can be integrated into the interaction system as a client only if the application on it implements the interface between the server and the client and provides an intuitive user interface. Multiple clients on the same or different technologies can work simultaneously. The recognition approach in this work has been proved to work with two popular sensing technologies, i.e. vision- and accelerometer-based tracking, although further improvement is necessary, and the gesture recognition module can be adapted to other enabling technologies easily with little changes (mainly in the

CHAPTER 5. USER STUDY

preprocessing step such as feature extraction). The scalability is not only on the client side but also on the server. The server determines what resources the users can control and can easily configure the services it provides. The definition and interpretation of the gesture vocabulary and its grammar, which reside in the server, is scalable on gestures and flexible to the mappings between gestures and their semantic meanings. When it comes to the comparison between the three enabling technologies in this work in terms of platform, the Kinect is only an input device and needs the support of a computer for calculation power and a display for visual feedback. It is stationary and to avoid repeating the training for each instance of the application in different settings, the gestures are recognized in the server for all the Kinect clients by storing templates of gestures. The delay introduced by the wireless communication of gesture data and recognition result is acceptable (see Chapter 5.3). On the other hand, the Phone and the Watch are integrated computing platforms consisting of the input, the output and the computation power (the Watch has limited power), and they are mobile. The gesture recognition is embedded into the clients for these two technologies in this work.

Finally, some limitations in the experimental design have to be mentioned here. First of all, the task order in the User Journey and the voice-based interaction were not counterbalanced to simplify the organization. This obviously introduced a non-manageable learning effect and fatigue. Results have been carefully explained when they are compared along the user tasks. Fortunately, the user tasks are not the main independent variable to investigate.

Secondly, the seated posture was not consistent in the whole test. In the recognition test, to investigate the impact of posture factor, the participants sat in a puf without back while sat in a sofa with back in the User Journey test. The results in the recognition test showed no significant difference in the gesture recognition rates with different postures. But it was observed in the User Journey that the gestures were difficult to recognize when they were performed as some participants leaned back. This observation is not enough to assert that the posture has significant impact on recognition performance but further experiments on the impact of various seated postures might be necessary.

Thirdly, the User Journey and voice-based interaction tested the short-term memorability of the gestural and vocal commands. Study on the learnability of these commands in long term, including grammars and vocabularies, would be recommended to further examine the technology adoption.

Fourth, the User Journey test and the voice-based interaction test were not equivalent since they had different number of user tasks (3 in common) and the participants went through different rooms in User Journey test but stayed in the same room in the voice-based interaction. This might have slight influence on the objective and subjective evaluations.

Apart from the limitations in the experimental design, the biases cannot be eliminated completely. The instructions of the experiment were explained by the facilitator personally and it was inevitable that there existed variations of the received messages of the participants although the facilitator followed a fixed routine to explain the instructions. For example, a situation where some participants ask particular questions on the user interface and the facilitator explain their doubts, probably makes them have different understandings of the

CONCLUSION

system from the understandings of those participants who did not ask. The external environment (lighting, noise level) could not be exactly the same. The thinking aloud technique helped to identify the participants' immediate feedback during the tests and interaction but it also has the drawback that the more the participants talk, the more their task or time performance may be influenced. In addition, the qualitative data analysis like video annotation, analysis on open questions in the questionnaire was done by one coder and it is subjective to biases.

Chapter 6

Conclusion

Smart space technology has a large potential to benefit our daily lives in the areas like energy efficiency, healthcare, entertainment, accessibility, etc., but there is room for improvement to effectively integrate it in the real operation workflow. In particular, natural, intuitive user interfaces providing high usability and immersive user experience might facilitate the adoption of smart services. Gesture-based interaction may be suitable to deliver intuitive user interfaces and presents some advantages when compared to other interaction manners. Additionally, it may be used even by people with loss of vision, motor skills and mobility. The recent development of ubiquitous computing platforms (like smartphones and smartwatches or similar wearables) and the commercial availability of powerful tracking systems (like the Kinect or Leap Motion) makes it much easier than before to develop engaging interaction systems based on gestures. This work is an attempt on exploring the design of gesture-based interaction systems in smart spaces/home-like environments.

After a review of the literature on the application scenarios and on the available sensors for gesture interaction, and also considering the available methods in the topics of gesture set design, device selection, the investigation on usability of gesture-based interaction and social acceptability, and the approaches of gesture recognition and gesture-based identification, the thesis introduces the design space and constraints of an interaction system to facilitate the control of resources in smart spaces with motion gestures. The instrumental interaction model is used to design and describe the interaction system. In particular, a grammar of gestures with a metaphor of natural language-based imperative sentences, along with localization context information to enable gestures portability is proposed to organize the resource entities and operations into a hierarchical structure. With the design goals set up, the thesis provides a wide description and analysis of the issues such as the client-server architecture, the on-line gesture sequence analyzer, the interaction workflow, and the user-end interface including the gesture input procedure for the three enabling technologies, control components and feedback of the user interface.

In particular, the Dynamic Time Warping (DTW) based gesture recognition approach was chosen to meet the requirements of the interaction system in this work, since it supports personalized gesture vocabularies, requires few training repetitions to function well,

CHAPTER 6. CONCLUSION

works with different types of data, also being easy to implement. After introducing the DTW classifier and the architecture of the recognition workflow, the DTW based recognition approach was evaluated offline and the results on a 30-gesture predefined vocabulary show that this approach works with both position and acceleration data, delivering high recognition accuracy. The relationship between the recognition accuracy and the number of training repetitions was tested and according to the results, three repetitions are enough to nearly reach a convergent recognition accuracy. User identification is essential to provide personalized services in smart spaces. Gesture-based identification, which is less invasive than the traditional approaches based on e.g. fingerprint and iris, is also explored based on the DTW classifier. The offline test with the Kinect indicated that the identification with concatenated gesture pairs significantly improved the accuracy when compared with the use of a single gesture.

A user study was carried out in a quasi-real environment, motivated by the research questions on three topics: the gesture recognition performance, the proposed grammar and the interaction. The offline test scenario of the recognition approach in Chapter 4 is highly restricted and the test conditions in the user study of Chapter 5 are closer to a real environment in which users perform gestures with the designed gesture input procedure for daily usage. In the experimentation setting, the training was light, the recognition was online and the factor of postures was considered. The user study includes open questions to analyze the gestural grammar in terms of acceptance, usability, limitations or alternatives. Additionally, the user study facilitates the analysis of user preferences on gesture vocabularies, as well as the convenience of using localization. At the level of the interaction system, the usability, user experience, comparison between enabling technologies, social acceptability and physical exertion are discussed. The user study was conceived following a within-group design, which requires a smaller sample size, being more robust to the impact of individual differences than the between-group design. Then the participants, the apparatus and the procedure composed of four phases including training, gesture recognition test, user journey and voice-based interactions are described in detail. The results collected from application logs, recorded videos and questionnaires were analyzed considering the errors in: the gesture input procedure, the gesture recognition and the user tasks, time performance and subjective assessments on: the gesture vocabulary, the grammar, physical efforts, user interface, training, and technology preferences. The online recognition provides a moderate accuracy, below the obtained in the offline test. No significant delay was introduced by the recognition approach in the gesture input workflow. The task duration with gestures is comparable to the one with voice, while more task errors have been made with gestures than with voice. Most of the participants thought that the grammar is easy to learn and can be effective to control the smart resources. A set of design recommendations to facilitate the design of interactive gesture-based systems is derived from the analysis of results. Specifically, design principles on gesture delimiters are inspired by the smartphone interface. The recommendations include issues such as the need for: a gesture recognition approach robust to the variability of the gesture performance, a simple grammar with rich interface contexts, a user interface with immediate feedback, clear metaphors to facilitate learning, options to person-

alize, or multimodal rich feedback. Additionally, some suggestions on the enhancement of the gesture input efficiency like the autocomplete feature, concatenated gesture performance or wrong input detection are also considered.

6.1 Discussion

With a gesture set of 10 gestures, the average training time with the Phone (34s) is significantly shorter than the time obtained with the Kinect and the Watch and this comparison result is consistent with the result that the gesturing time with the Phone is the shortest. The participants also gave positive feedback on the training with the Phone. The study analyzes and compares the designs of the gesture performance procedure through the three enabling technologies.

The offline gesture recognition approach worked on both position and acceleration data, and addressed some of the design challenges mentioned at the beginning of Chapter 4. As a user-dependent based technique, it avoids the influence of performance variations between different users. The DTW classifier was designed to deal with the nonlinear fluctuation of the signal in the time axis, then being robust to the speed variations of the gesture performance. The offline tests showed that some gestures had more “personal” information and facilitated identification procedures while others were good for user-independent recognition scenarios. The online recognition test as well as the user study was partly guided by the results of the offline evaluation that showed good recognition accuracy and time performance, and slight dependence on the training repetitions. The gesture agreement analysis in the user study revealed that the participants had diverse choices of gestures to associate a task command, and promoted the necessity of the user-dependent working mode of the gesture recognition approach and the option to personalize the vocabulary.

On top of the single gesture input and recognition, a gestural grammar is proposed to facilitate the configuration of the commands by using a structure of the imperative sentence to organize the control of target resources and their operation. The objective is to reuse the operation gestures among different smart resources, thus making the gesture vocabulary compact and intuitive, and to improve the learnability of the gesture-based interaction.

The gesture recognition approach is learning-based and enables the customization of the gesture vocabulary. The client-server architecture of the interaction system allows the easy scaling of the number and type of smart resources integrated in the interaction system. Enabling technologies of gesture-based interaction on the client side can be easily incorporated by implementing the interface between the client and the server and an end-user interface. The association between gestures and commands is configurable with the help of a database. This flexible configuration of the architecture design would make it possible to port the system to other smart spaces.

6.2 Limitations

Since the proposed grammar utilizes the initial letters or acronyms to specify the target resources, it is inevitable the existence of conflicts when multiple resources have the same initial letter or acronym. In the user study, the letter conflict problem was inquired to the participants who recommended appending letters or digits, combining additional dimension of specification such as pointing, localization and body orientation. In addition, detailed parameters that may be necessary to control a specific resource are difficult to specify in the framework of the grammar, such as the target temperature of air conditioning, the name of the file to open, etc. The localization context restricts the control resolution into room level and the devices of the same type in the same room are not distinguishable.

The DTW-based gesture recognition approach does not consider the view variations due to the different orientations of the users' body relative to the Kinect camera (although the interface of the sweet spot gives feedback to guide the users to have their body facing the camera), and the orientation variations of the device frame when users perform gestures with smartphones and smartwatches. In addition, the approach to avoid false positive errors was implemented by setting an empirical threshold on the DTW distance. Better strategies are to be developed as future work. The evaluation of the approach was in one day and did not test the impact of the change of performance habit in a long term.

As mentioned in Chapter 5.5, there are some limitations in the experimental design. The task order in the user journey test was not counterbalanced for easier organization and the results related to the user tasks were explained carefully. Inconsistencies were found in the seated postures in the gesture recognition test and user journey phase, the task set for user journey and voice-based interaction. The user study tested the short-term memorability of the gestural and vocal commands but did not evaluate the long-term learnability of the designed interaction system. There are other biases caused by the differences on the instructions provided to the participants, the varying characteristics of the external environment, the interference of the thinking aloud technique and the way in which video annotation was completed.

6.3 Future work

On the gesture input procedure

The gesture input procedure, as one of the foundational modules of the gesture-based interaction system, has significant influence on the performance of gesture recognition, user experience and the overall effectiveness and efficiency of the interaction. The Imprecise Gesture Boundaries (IGB) errors make the captured gesture data be appended extra data at the ends of the interested gesture data or be cut off part of the gesture data, so that they introduce potential inconsistencies in gesture capture and degenerate the recognition accuracy. In addition, the Unintended Gesture Input (UGI) errors, Fail to Start Gesture (FSG) errors and Instability of Hand Tracking (IHT) errors made the users feel frustrated

and without control. The gesture capture process and recognition errors prolonged the task duration due to the need of repeating the gestures not well recognized. The improvement of the gesture input procedure in terms of reducing the input errors by following the design recommendations in Chapter 5.4 will benefit the performance of the gesture recognition and the global interaction system. The Kinect v2 has better hand tracking performance, which might reduce the Instability of Hand Tracking (IHT) errors. In addition, it can immediately recognize two-finger pointing pose apart from hand open and hand closed poses. These hand poses, for example the two-finger point pose and hand closed pose to indicate gesturing or not, can be used as delimiters and hopefully this design will relieve Unintended Gesture Input (UGI), Imprecise Gesture Boundaries (IGB) and Fail to Start Gesture (FSG) errors since it gives the users more control than the velocity threshold-based delimiter does. What is more, this design follows the proposed recommendations in Chapter 5.4. The wrist rotation to enable gesture input for the Watch-based application seemed cumbersome and the rotation might have influence on the collected acceleration data. Deus Ex Aria [26] is a watchband that detects finger movement from activity in the tendons of the wrist with sensors inside. Similar to the proposal of the Kinect v2, finger movements such as a pinch gesture can be adopted as delimiters. This will enhance the user experience by reducing errors like Unintended Gesture Input (UGI) and Fail to Start Gesture (FSG) errors, and at the same time captures gesture data of higher quality when compared to those gathered when using the wrist rotation. The proposal with the Kinect v2 and Deus Ex Aria would be checked to improve the gesture input procedure and validate the design recommendations discussed in Chapter 5.4.

Training

Although each application of the interaction system requires one training sample for each gesture, which is a minimum amount of the training that can be required by a user-dependent recognition approach, the users have to train the same gesture set for each technology. Imagine a future in which the devices supporting gesture input in home automation environment are widespread and there are, say 10 different devices; the users would then have to train all the 10 devices to have the possibility of using them all, although the gesture sets are exactly the same, which is very annoying. One solution of course is to adopt the user-independent manner, completely eliminating the need of training for new users. If the user-dependent manner is to be used, it would be interesting to create a global training approach so that the training data on one technology can be used by the others, by for example, describing the gesture data with a set of universal features. And it would be better to be a one-shot training, by extracting as much related features as possible from the single training sample like [96], looking for relationships between new users' gesture performance with those of existing users [52], or selecting representative templates for DTW-based approaches [37].

Gesture recognition

In the user journey test, the many gesture recognition errors forced the participants to repeat gestures, enlarging the task duration, degenerating the user experience and demanding higher physical efforts. Some participants thought that the recognition rate was one of the factors to determine the technology preference between gestures and voice. It would be one of the most urgent features to improve in the next design iteration. From the discussion in Chapter 5.4, the fundamental reason leading to wrong gesture recognition is the mismatch between the gesture templates and the samples. The challenges the current recognition approach faces and the possible improvements to address these issues are discussed in the following.

First of all, the design of gesture input procedure of the Kinect had many Imprecise Gesture Boundaries (IGB) errors that make the gesture data capture inconsistent. In the Watch the wrist rotation distorts the acceleration data at the beginning when the wrist recovers to the normal posture. As what has discussed above, the Kinect v2 and Deus Ex Aria watchband could improve the quality of captured gesture data and will be given a try.

Second, it was observed that the participants performed same gestures with different orientations when using the Phone and the Watch, and the recognition approach, which does not address the orientation variation, is sensitive to this variation. There have been several recent works on this topic. Extracting device orientation-independent features [18], transforming gesture data to a reference coordinates system [37] or to the templates' coordinates system [53] may be examined in the future. For the Kinect-based application, the recognition is position-invariant but subjects to scale and rotation variations. It will have problems when users train the gestures with the body facing the device but do gestures with the body facing perpendicular to the device orientation. The shape features, which are proposed in [107] and describe the gesture trajectory in a way invariant to rotation, scale and translation, could be able to enhance the recognition approach to be view-invariant.

Thirdly, the users can alternate their writing manners. For example, it was observed in the User Journey test that the same participant began the letter 't' with horizontal bar in one time and with the vertical stroke in another. Multiple templates for one gesture can cover a certain amount of the performance variation but they also increase the computational complexity, since more templates have to be compared. Statistical methods would be better to model this performance variation, for example the Gaussian Mixture Model (GMM) [10] [31]. In a long term of view, the performance manner of gestures will evolve so that the difference between the templates and actual performance increases over time. The templates would have to be updated constantly according to certain criteria. Other machine learning algorithms can also be considered for long term adaption to the change of the users' behavior, such as the Hidden Markov Models (HMMs). Here there is a tradeoff between the availability of training data and the capability to adapt the performance variance. While some approaches are better than the template-based approaches to model and learn the variations, they normally need more data to obtain the model. It would be interesting to develop a hybrid technique that recognizes gestures with a template-based approach at the beginning with little training and feeds the users' gesture data to the training of statistical methods at

the same time when the system is being used, to finally switch to the statistical methods. The scope of the above discussion is limited in user-dependent recognition, learning new users' performance from scratch. It is a different story if the user-independent recognition is to be used, since the statistical models can be trained offline with a large amount of data before its use. Although there is no standard gesture sets as the natural language, there are some widely accepted gestures, considering more constraints like the application scenario and the culture background. No matter whether in user-dependent or user-independent mode, the recognition approach has limitations to distinguish gestures so the gesture vocabulary should be carefully designed and controlled to avoid similar gestures. A tool to quantify the "resolution" of the gesture set might help a lot for the gesture set design and also to remind the users when their personalized gesture vocabulary has extremely similar gestures.

Fourth, the current recognition approach sets a large distance threshold between the gesture template and the samples to reject false positives, which happen when a gesture motion out of the defined gesture set is recognized as one of the define gestures. In the future, more accurate and reliable measurement should be thought out to detect false positives.

Design

Feedforward. The design of the system did not consider self-revealing of gesture commands so that novice users will not know what gestures are available and how to trigger them. The work [25] argues that a gesture guiding system is critical to the adoption of gesture commands and gesture-based interaction. It provides a design space that has a unified framework and organizes a set of design issues along axes. This work can be a first step to the study on guiding systems to enhance the adoption of the system.

Physical exertion. Fatigue is a widely known drawback of gesture-based interaction. Recently the sensors are getting more and more accurate and powerful. Many of them can stably and accurately track finger movements such as the Kinect v2, Leap Motion, Deus Ex Aria, just name a few. It would become feasible to develop finger-gesture-based applications. Since the user study showed medium physical demand in the arm using hand gestures, the interaction with finger gestures will hopefully reduce the exertion. In addition, the usability and user experience can also be compared with the hand-gesture-based interaction.

UI and system

The interface and functionality are to be refined. The interface of the Watch should be more independent and provide feedback itself without the help of the paired Phone. Although the architecture and backend support gesture personalization and configuration on the association between gestures and commands, this functionality is not available in the actual interface and will be added in the future. As discussed before, a tool to estimate the distinguishability of gestures in a personalized vocabulary is necessary to remind users if they propose too similar gestures, in order to keep a decent recognition rate. The feasibil-

CHAPTER 6. CONCLUSION

ity to enlarge the coverage of the Kinect camera by deploying multiple devices will also be considered to reduce the constraints of the interaction area.

User study

The survey on the social acceptability of the gesture-based interaction was done in a lab environment and it did not distinguish the gesture types such as device-based gestures and hand-free gestures. A test in the wild considering the impact of gesture types should be carried out to elicit the constraint of social acceptability on gesture design and social scenarios of the applications. In the user study, the gestures were predefined and some of them had low agreement score. It would be interesting to check whether the grammar error rate could be reduced if the users are allowed to personalize the gesture vocabulary and associate gestures with commands. A user study in a more realistic environment would be extremely valuable. For example, involving normal people to use the interaction system in their home for a long period, say several weeks, to evaluate the usability, user experience and performance will reveal more problems and design recommendations.

Appendix A

Offline evaluation of the gesture recognition approach

A.1 Confusion matrices

This part lists the confusion matrices of gestures in the three groups of our vocabulary in Table 4.1 of Chapter 4: directional movements, small letters and capital letters. The first three tables are test results based on Kinect data and the latter three are based on phone data.

Table A.1: Confusion matrix about detection of directional movements based on the Kinect

Directional	Recognized as on the Kinect (%)									
	Up	Down	Left	Right	For.	Back.	C. C.	CC. C.	S. L.	S. R.
Up	99.1	0.9	0	0	0	0	0	0	0	0
Down	0	100	0	0	0	0	0	0	0	0
Left	0	0	99.1	0.9	0	0	0	0	0	0
Right	0	0	0	99.1	0.9	0	0	0	0	0
For.	0	0	0	0	100	0	0	0	0	0
Back.	1.8	0	0	0	0	98.2	0	0	0	0
C. C.	0	0.9	0	1.8	0	0	97.3	0	0	0
CC. C.	0	0	0	0	0	0	0	100	0	0
S. L.	0	0	0	0	0	0	0	0	100	0
S. R.	0	0	0	0	0	0.9	0	0	0	99.1

APPENDIX A. OFFLINE EVALUATION OF THE GESTURE RECOGNITION APPROACH

Table A.2: Confusion matrix about detection of small letters based on the Kinect

Small letters	Recognized as on the Kinect (%)									
	d	w	tv	p	l	ac	r	f	m	s
d	100	0	0	0	0	0	0	0	0	0
w	0	98.2	1.8	0	0	0	0	0	0	0
tv	0	0	99.1	0	0	0	0	0	0	0.9
p	0	0	0	99.1	0.9	0	0	0	0	0
l	0	0	0	0	100	0	0	0	0	0
ac	0	0	0.9	0	0	98.2	0	0	0	0.9
r	0	0	0	1.8	0	0	98.2	0	0	0
f	0	0	0	0	1.8	0	0	97.3	0	0.9
m	0	0.9	0	0	0	0	0	0	99.1	0
s	0	0	0	0	0	0	0	0	0	100

Table A.3: Confusion matrix about detection of capital letters based on the Kinect

Capital letters	Recognized as on the Kinect (%)									
	D	W	TV	P	L	AC	R	F	M	S
D	95.5	0	0	2.7	0	0	0	1.8	0	0
W	0	99.1	0	0	0	0	0	0	0	0.9
TV	0	0	99.1	0	0	0	0	0	0	0.9
P	6.4	0	0	92.7	0	0	0.9	0	0	0
L	0	0	0	0	100	0	0	0	0	0
AC	0.9	0	0.9	0	0	98.2	0	0	0	0
R	0	0.9	0.9	0	0	0	97.3	0.9	0	0
F	0	0	0.9	0	0	0	0.9	97.3	0.9	0
M	0	0	0	0	0	0	0	0	100	0
S	0	0	0	0	0	0	0	0.9	0	99.1

A.2 Handwriting of letters

Figure A.1 is a scanned copy of handwritings of the capital letters and small letters of the gesture vocabulary in Table 4.1 of Chapter 4, written by several colleagues in the laboratory. It is observed that different persons wrote the same letter in various manners and one person may write different letters in similar, confusing way, such as the case of ‘D’ and ‘P’. It can be seen that the forms of capital letter handwriting vary a little while the forms of small letters change much more from one user to another (e.g. ‘a’, ‘r’, ‘f’). The order as well as

Table A.4: Confusion matrix about detection of directional movements based on the Phone

Directional	Recognized as on the Phone (%)									
	Up	Down	Left	Right	For.	Back.	C. C.	CC. C.	S. L.	S. R.
Up	78.2	3.6	3.6	3.6	0	5.5	3.6	1.8	0	0
Down	0	92.7	1.8	1.8	3.6	0	0	0	0	0
Left	3.6	0	90.9	3.6	0	1.8	0	0	0	0
Right	0	0	3.6	94.5	0	1.8	0	0	0	0
For.	0	0	0	0	98.2	1.8	0	0	0	0
Back.	0	0	0	1.8	1.8	96.4	0	0	0	0
C. C.	0	0	0	0	0	0	96.4	0	0	3.6
CC. C.	0	0	0	0	0	1.8	0	96.4	1.8	0
S. L.	0	0	0	0	0	0	0	1.8	98.2	0
S. R.	0	0	0	0	0	0	0	0	0	100

Table A.5: Confusion matrix about detection of small letters based on the Phone

Small letters	Recognized as on the Phone (%)									
	d	w	tv	p	l	ac	r	f	m	s
d	98.2	1.8	0	0	0	0	0	0	0	0
w	1.8	94.5	3.6	0	0	0	0	0	0	0
tv	0	1.8	94.5	0	0	3.6	0	0	0	0
p	0	0	1.8	96.4	0	1.8	0	0	0	0
l	1.8	3.6	0	0	94.5	0	0	0	0	0
ac	0	0	1.8	0	0	96.4	0	0	0	1.8
r	0	0	0	0	0	0	100	0	0	0
f	0	7.3	0	0	0	3.6	0	89.1	0	0
m	0	1.8	1.8	1.8	0	0	0	1.8	92.7	0
s	0	0	1.8	0	1.8	0	0	0	0	96.4

the direction of strokes of some letters also change among the users, like ‘T’, ‘R’, ‘M’ and ‘d’.

APPENDIX A. OFFLINE EVALUATION OF THE GESTURE RECOGNITION APPROACH

Table A.6: Confusion matrix about detection of capital letters based on the Phone

Capital letters	Recognized as on the Phone (%)									
	D	W	TV	P	L	AC	R	F	M	S
D	78.2	3.6	0	14.5	0	0	0	0	3.6	0
W	3.6	96.4	0	0	0	0	0	0	0	0
TV	0	3.6	92.7	0	0	0	0	1.8	1.8	0
P	9.1	0	0	87.3	0	0	3.6	0	0	0
L	3.6	0	0	0	92.7	0	0	0	0	3.6
AC	0	0	1.8	3.6	0	89.1	1.8	1.8	1.8	0
R	3.6	0	0	5.5	1.8	0	89.1	0	0	0
F	0	0	0	3.6	0	0	1.8	94.5	0	0
M	1.8	1.8	0	0	0	0	0	1.8	94.5	0
S	0	0	0	0	0	0	0	0	0	100

Figure A.1: Survey of handwriting of letters

Appendix B

Brief introduction to the statistical methods used in data analysis

The statistical methods used in this article include one-way and two-way repeated measures ANOVA (Analysis of Variance), paired-samples t-test and Friedman test, all of which are used to detect differences between groups and were carried out using SPSS software package in this work. Selection of statistical methods depends on design of experiments and types of variables (continuous or categorical). In addition, their assumptions must be met before applying the methods. Use scenarios, assumptions examination, and explanation and further analysis of results for the three methods are introduced in the following. For more explanation to these methods and how to do the tests with SPSS, please refer to this tutorial [59].

B.1 Repeated measures ANOVA

Compared with standard ANOVA, repeated measures ANOVA is used when all members of a random sample go through a number of different conditions. It is appropriate to the within-group design of the user study in this work. One-way repeated measures ANOVA is for experiments with only one independent variable (e.g. the comparison of mean training time between the three technologies in Chapter 5.3.2). Two-way (or two-factor) repeated measures ANOVA, on the other hand, is used when there are two independent variables (e.g. the investigation of the impact of both technologies and postures to recognition accuracy in Chapter 5.3.2).

Repeated measures ANOVA assumes that the dependent variable is continuous and that the independent variable consists of two or more categorical groups. There should be no significant outliers, which can be detected by boxplots. Then the distribution of the dependent variable should be approximately normally distributed. Normality can be tested with Shapiro-Wilk test. Finally, there is a requirement of sphericity, which refers to the condition where the variance of the differences between all possible pairs of groups (levels/conditions of the independent variable) are equal. Mauchly's test detects whether sphericity assumption

APPENDIX B. BRIEF INTRODUCTION TO THE STATISTICAL METHODS USED IN DATA ANALYSIS

is met.

The existence of statistically significant difference between groups is detected in this article at a 5% significance level, i.e. the difference is significant if the p value is lower than 0.05 (the significance level is the probability of rejecting the null hypothesis when it is true). Repeated measures ANOVA is an omnibus test and cannot tell which specific group within each factor are significantly different from each other. It only tells that at least two of the groups are different. After significant difference is detected, post hoc test provides pair-wise comparison results between groups and determines which of these groups differ from each other.

B.2 Paired-samples t-test

The paired-samples t-test (dependent t-test) compares the means between only two related groups, instead of two or more as in the case of repeated measures ANOVA. It assumes the same subjects are present in both groups (appropriate for within-group design), the dependent variable is measured on a continuous scale, and there is only one independent variable. Before using this method, other assumptions to be checked include no significant outliers and approximately normal distribution of the differences in the dependent variable between the two related groups.

The statistically significant difference is reported at 5% significance level as well in this article with this method, which is used to explore the influence of gesture type (directional and letter-drawing gestures) on recognition rate, and posture (standing or seated) on gesture performance time in Chapter 5.3.2. Since there are only two groups, we can determine from the direction of the t-value which group has higher or better values.

B.3 Friedman test

The Friedman test is a non-parametric alternative to the one-way repeated measures ANOVA. It is used in this article due to violation of the assumptions of one-way repeated measures ANOVA when comparing ratings on social acceptability in Chapter 5.3.3, ratings on training, scores of System Usability Scale (SUS) and User Experience Questionnaire (UEQ) in Chapter 5.3.4.

It assumes that the independent variable has three or more levels/conditions and that the dependent variable is measured at an ordinal or continuous level. Different from the one-way repeated measures ANOVA, the Friedman test does not require the samples to be normally distributed.

Friedman test is an omnibus test as its parametric alternative (one-way repeated measures ANOVA) i.e. it only tells whether there is difference between groups but does not indicate specifically which groups differ from others. If a significant difference is detected by Friedman test, post hoc tests with Wilcoxon signed-rank tests on different combinations of related

groups should be run to determine which groups are different. A Bonferroni adjustment on the results from Wilcoxon tests is necessary to avoid Type I error, which is made more likely due to multiple comparisons. The Bonferroni adjustment is easy to calculate: dividing the significance level (5% in this article) by the number of tests run (e.g. 3 in the case of the comparison of ratings on training since there are 3 groups), and the new significance level is $0.05/3 = 0.0167$. Alternatively, new versions of SPSS give adjusted p value directly using Bonferroni's correction at the significance level of 5%.

Appendix C

Questionnaire of the user study

C.1 User profile

Name

Gender ☐Male ☐Female

Age

Handedness ☐Left-handed ☐Right-handed

Do you wear watch and/or bracelets normally? ☐Yes ☐No

How often do you use these devices?

	Never	Seldom	Sometimes	Often	Daily
Microsoft Kinect					
Leap Motion					
Wii					
Smartphone					
Smartwatch					
Other gesture enabled devices					

Do you have the experience of developing applications of these devices?

	Yes	No
Microsoft Kinect		
Leap Motion		
Wii		
Smartphone		
Smartwatch		
Other gesture enabled devices		

C.2 Training section

C.2.1 Kinect

	Strongly disagree				Strongly agree
I feel natural to start and finish the gestures using Kinect	1	2	3	4	5
The training procedure with Kinect is boring	1	2	3	4	5
The training application of Kinect is easy to use	1	2	3	4	5
The training took a lot of efforts	1	2	3	4	5
I felt lost during the interaction	1	2	3	4	5

Was the feedback provided by the training interface as expected? Please explain your answer.

C.2.2 Phone

	Strongly disagree				Strongly agree
I feel natural to start and finish the gestures using Phone	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
The training procedure with Phone is boring	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
The training application of Phone is easy to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
The training took a lot of efforts	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
I felt lost during the interaction	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

Was the feedback provided by the training interface as expected? Please explain your answer.

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

C.2.3 Watch

	Strongly disagree				Strongly agree
I feel natural to start and finish the gestures using Watch	1	2	3	4	5
The training procedure with Watch is boring	1	2	3	4	5
The training application of Watch is easy to use	1	2	3	4	5
The training took a lot of efforts	1	2	3	4	5
I felt lost during the interaction	1	2	3	4	5

Was the feedback provided by the training interface as expected? Please explain your answer.

--

C.2.4 Summary

How do you prefer doing gestures with these devices? Rank these devices by preference (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Do you find more convenient to train gestures interlaced or to finish the repetitions for every single gesture sequentially?

☐ Sequential ☐ Interlaced

Please provide your suggestions to improve the training.

C.3 User journey section

C.3.1 Gesture preference

What gesture would you choose to replace the question mark in the gesture chain in the following questions for each task? Please feel free to suggest one or more, if you feel like.

1. For task “Raise temperature”, the gesture chain would be “t → ?”.

2. For task “Check weather forecast”, the gesture chain would be “w → ?”.

3. For task “Pull down blind”, the gesture chain would be “b → ?”.

4. For task “Set color of HUE to red”, the gesture chain would be “h → ? → r”.

5. For task “Set lights in TV mode”, the gesture chain would be “l → ? → tv”.

6. For task “Make robot approach to yourself”, the gesture chain would be “r → ?”.

7. For task “Play music”, the gesture chain would be “m → ?”.

8. For task “Close movie”, the gesture chain would be “tv → ?”.

9. For task “Play on a tablet the movie in TV”, the gesture chain would be “tv → ? → t”.

Would you prefer capital or small letters to specify resource and parameter if the choice has no influence on the application performance?

☐Small letter ☐Capital letter

Would you prefer predefined gesture set or defining your own gestures to interact

☐Predefined gestures ☐Your own defined gestures

C.3.2 Kinect

System Usability Scale

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	☐	☒	☐	☐	☐	☐	☐	unattractive
------------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	--------------

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by marking one circle per line.

USER EXPERIENCE QUESTIONNAIRE

	1	2	3	4	5	6	7		
annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable	1
not understandable	☐	☐	☐	☐	☐	☐	☐	understandable	2
creative	☐	☐	☐	☐	☐	☐	☐	dull	3
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn	4
valuable	☐	☐	☐	☐	☐	☐	☐	inferior	5
boring	☐	☐	☐	☐	☐	☐	☐	exciting	6
not interesting	☐	☐	☐	☐	☐	☐	☐	interesting	7
unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable	8
fast	☐	☐	☐	☐	☐	☐	☐	slow	9
inventive	☐	☐	☐	☐	☐	☐	☐	conventional	10
obstructive	☐	☐	☐	☐	☐	☐	☐	supportive	11
good	☐	☐	☐	☐	☐	☐	☐	bad	12
complicated	☐	☐	☐	☐	☐	☐	☐	easy	13
unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing	14
usual	☐	☐	☐	☐	☐	☐	☐	leading edge	15
unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant	16
secure	☐	☐	☐	☐	☐	☐	☐	not secure	17
motivating	☐	☐	☐	☐	☐	☐	☐	demotivating	18
meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations	19
inefficient	☐	☐	☐	☐	☐	☐	☐	efficient	20
clear	☐	☐	☐	☐	☐	☐	☐	confusing	21
impractical	☐	☐	☐	☐	☐	☐	☐	practical	22
organized	☐	☐	☐	☐	☐	☐	☐	cluttered	23
attractive	☐	☐	☐	☐	☐	☐	☐	unattractive	24
friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly	25
conservative	☐	☐	☐	☐	☐	☐	☐	innovative	26

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

Other questions

Rate the usefulness of the following elements in the interface (1 no useful at all; 7 extremely useful):

	1	2	3	4	5	6	7
Sweet spot							
Progress bar							
Semaphore							

What would be your suggestions to improve the feedback of the interface?

And any other general comments?

C.3.3 Phone

System Usability Scale

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	☐	☒	☐	☐	☐	☐	☐	unattractive
------------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	--------------

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by marking one circle per line.

USER EXPERIENCE QUESTIONNAIRE

	1	2	3	4	5	6	7		
annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable	1
not understandable	☐	☐	☐	☐	☐	☐	☐	understandable	2
creative	☐	☐	☐	☐	☐	☐	☐	dull	3
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn	4
valuable	☐	☐	☐	☐	☐	☐	☐	inferior	5
boring	☐	☐	☐	☐	☐	☐	☐	exciting	6
not interesting	☐	☐	☐	☐	☐	☐	☐	interesting	7
unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable	8
fast	☐	☐	☐	☐	☐	☐	☐	slow	9
inventive	☐	☐	☐	☐	☐	☐	☐	conventional	10
obstructive	☐	☐	☐	☐	☐	☐	☐	supportive	11
good	☐	☐	☐	☐	☐	☐	☐	bad	12
complicated	☐	☐	☐	☐	☐	☐	☐	easy	13
unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing	14
usual	☐	☐	☐	☐	☐	☐	☐	leading edge	15
unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant	16
secure	☐	☐	☐	☐	☐	☐	☐	not secure	17
motivating	☐	☐	☐	☐	☐	☐	☐	demotivating	18
meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations	19
inefficient	☐	☐	☐	☐	☐	☐	☐	efficient	20
clear	☐	☐	☐	☐	☐	☐	☐	confusing	21
impractical	☐	☐	☐	☐	☐	☐	☐	practical	22
organized	☐	☐	☐	☐	☐	☐	☐	cluttered	23
attractive	☐	☐	☐	☐	☐	☐	☐	unattractive	24
friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly	25
conservative	☐	☐	☐	☐	☐	☐	☐	innovative	26

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

Other questions

What would be your suggestions to improve the feedback of the interface?

And any other general comments?

C.3.4 Watch

System Usability Scale

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

attractive	☐	☒	☐	☐	☐	☐	☐	unattractive
------------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	--------------

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Please assess the product now by marking one circle per line.

USER EXPERIENCE QUESTIONNAIRE

	1	2	3	4	5	6	7		
annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable	1
not understandable	☐	☐	☐	☐	☐	☐	☐	understandable	2
creative	☐	☐	☐	☐	☐	☐	☐	dull	3
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn	4
valuable	☐	☐	☐	☐	☐	☐	☐	inferior	5
boring	☐	☐	☐	☐	☐	☐	☐	exciting	6
not interesting	☐	☐	☐	☐	☐	☐	☐	interesting	7
unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable	8
fast	☐	☐	☐	☐	☐	☐	☐	slow	9
inventive	☐	☐	☐	☐	☐	☐	☐	conventional	10
obstructive	☐	☐	☐	☐	☐	☐	☐	supportive	11
good	☐	☐	☐	☐	☐	☐	☐	bad	12
complicated	☐	☐	☐	☐	☐	☐	☐	easy	13
unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing	14
usual	☐	☐	☐	☐	☐	☐	☐	leading edge	15
unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant	16
secure	☐	☐	☐	☐	☐	☐	☐	not secure	17
motivating	☐	☐	☐	☐	☐	☐	☐	demotivating	18
meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations	19
inefficient	☐	☐	☐	☐	☐	☐	☐	efficient	20
clear	☐	☐	☐	☐	☐	☐	☐	confusing	21
impractical	☐	☐	☐	☐	☐	☐	☐	practical	22
organized	☐	☐	☐	☐	☐	☐	☐	cluttered	23
attractive	☐	☐	☐	☐	☐	☐	☐	unattractive	24
friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly	25
conservative	☐	☐	☐	☐	☐	☐	☐	innovative	26

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

Other questions

How do you feel about using vibration as feedback tool for the service?

	1	2	3	4	5	6	7	
I don't like it at all								I think it's extremely useful

What would be your suggestions to improve the feedback of the interface?

And any other general comments?

C.3.5 Summary

Esfuerzo (Escala de Borg para esfuerzos del brazo, del hombro y del cuello (fuente: ISO-9241))

Esfuerzo			Esfuerzo
Brazo	Hombro	Cuello	
() 10	() 10	() 10	Extremadamente importante (casi máximo)
() 9	() 9	() 9	
() 8	() 8	() 8	
() 7	() 7	() 7	Muy importante
() 6	() 6	() 6	
() 5	() 5	() 5	Importante (pesado)
() 4	() 4	() 4	Bastante importante
() 3	() 3	() 3	Moderado
() 2	() 2	() 2	Débil (ligero)
() 1	() 1	() 1	Muy débil
() 0,5	() 0,5	() 0,5	Extremadamente débil (apenas perceptible)
() 0	() 0	() 0	Inexistente

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

	Strongly disagree				Strongly agree
It is cumbersome to gesture in the air	☐	☐	☐	☐	☐
	1	2	3	4	5
The grammar is easy to learn	☐	☐	☐	☐	☐
	1	2	3	4	5
The grammar can be effective to command the environment	☐	☐	☐	☐	☐
	1	2	3	4	5
I would like to perform gestures of a gesture chain in one go instead of doing one gesture, waiting feedback and then doing another gesture	☐	☐	☐	☐	☐
	1	2	3	4	5
I like that the service is location-aware, being effective over the objects in the space where the user is	☐	☐	☐	☐	☐
	1	2	3	4	5
I would like an additional step to confirm or negate the command after performing the gesture chain	☐	☐	☐	☐	☐
	1	2	3	4	5

SUMMARY

Can you think out other format of grammar to command the environment except the grammar in this test?

Suppose that you could configure the association between gestures and devices. How would you resolve the conflicts of letters (For example, there are both robot and radio in the smart space and they would both be assigned ‘r’)?

I would be willing to perform the gestures in this test in (1 - Strongly disagree, 10 - Strongly agree)

	1	2	3	4	5	6	7	8	9	10
Home										
Pavement or Sidewalk										
Driving										
Passenger on a Bus or Train										
Pub or Restaurant										
Workplace										

I would be willing to perform the gestures in front of (1 - Strongly disagree, 10 - Strongly agree)

	1	2	3	4	5	6	7	8	9	10
Alone										
Partner										
Friends										
Colleagues										
Strangers										
Family										

Rank the devices by preference in the task “Check weather forecast” (1 - favorite, 3 - least favorite)

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

	1	2	3
Kinect			
Phone			
Watch			

Rank the devices by preference in the task “Pull down blind” (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Rank the devices by preference in the task “Set color of HUE to red” (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Rank the devices by preference in the task “Set lights in TV mode” (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Rank the devices by preference in the task “Make robot approach to yourself” (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Rank the devices by preference in the task “Play music” (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

SUMMARY

Rank by preference the three technologies that have been used to support gesture recognition (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Rank by preference the three technologies in the following situations if all of them are available: Home (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Street (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Driving (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Passenger on public transportation means (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

Pub or restaurant (1 - favorite, 3 - least favorite)

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

	1	2	3
Kinect			
Phone			
Watch			

Workplace (1 - favorite, 3 - least favorite)

	1	2	3
Kinect			
Phone			
Watch			

C.4 Voice-based services section

C.4.1 System Usability Scale

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

APPENDIX C. QUESTIONNAIRE OF THE USER STUDY

C.4.2 Summary

	Strongly disagree				Strongly agree
I prefer gestures to voice to interact with the objects in the environment	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
I prefer voice to watch-based gestures	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
I prefer smartphone-based gestures to voice	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
I prefer voice to Kinect-based gestures	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

What would be the factors that determine your preference between gesture- and voice-based interaction?

- ☐ noise level of the environment
- ☐ level of fatigue of these two techniques
- ☐ Other:

Please include any additional comment you may consider

Bibliography

- [1] Yomna Abdelrahman, Alireza Sahami Shirazi, Niels Henze, and Albrecht Schmidt. “Investigation of Material Properties for Thermal Imaging-Based Interaction”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 15–18. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702290. URL: <http://doi.acm.org/10.1145/2702123.2702290>.
- [2] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos. “A Survey on Ambient Intelligence in Healthcare”. In: *Proceedings of the IEEE* 101.12 (2013), pp. 2470–2494. ISSN: 0018-9219. DOI: 10.1109/JPROC.2013.2262913.
- [3] A. Akl, Chen Feng, and S. Valaee. “A Novel Accelerometer-Based Gesture Recognition System”. In: *Signal Processing, IEEE Transactions on* 59.12 (2011), pp. 6197–6205. ISSN: 1053-587X. DOI: 10.1109/TSP.2011.2165707.
- [4] Florian Alt, Andreas Bulling, Gino Gravanis, and Daniel Buschek. “GravitySpot: Guiding Users in Front of Public Displays Using On-Screen Visual Cues”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. UIST '15. Daegu, Kyungpook, Republic of Korea: ACM, 2015, pp. 47–56. ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807490. URL: <http://doi.acm.org/10.1145/2807442.2807490>.
- [5] Christoph Amma and Tanja Schultz. “Airwriting: Demonstrating Mobile Text Input by 3D-space Handwriting”. In: *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. IUI '12. Lisbon, Portugal: ACM, 2012, pp. 319–320. ISBN: 978-1-4503-1048-2. DOI: 10.1145/2166966.2167032. URL: <http://doi.acm.org/10.1145/2166966.2167032>.
- [6] Gilles Bailly, Dong-Bach Vo, Eric Lecolinet, and Yves Guiard. “Gesture-aware Remote Controls: Guidelines and Interaction Technique”. In: *Proceedings of the 13th International Conference on Multimodal Interfaces*. ICMI '11. Alicante, Spain: ACM, 2011, pp. 263–270. ISBN: 978-1-4503-0641-6. DOI: 10.1145/2070481.2070530. URL: <http://doi.acm.org/10.1145/2070481.2070530>.

BIBLIOGRAPHY

- [7] Gilles Bailly, Jörg Müller, Michael Rohs, Daniel Wigdor, and Sven Kratz. “ShoeSense: A New Perspective on Gestural Interaction and Wearable Applications”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 1239–1248. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208576. URL: <http://doi.acm.org/10.1145/2207676.2208576>.
- [8] Sara Bartolini, Bojan Milosevic, Alfredo D’Elia, Elisabetta Farella, Luca Benini, and Tullio Salmon Cinotti. “Reconfigurable Natural Interaction in Smart Environments: Approach and Prototype Implementation”. In: *Personal Ubiquitous Comput.* 16.7 (Oct. 2012), pp. 943–956. ISSN: 1617-4909. DOI: 10.1007/s00779-011-0454-5. URL: <http://dx.doi.org/10.1007/s00779-011-0454-5>.
- [9] M. Á Bautista, A. Hernández-Vela, S. Escalera, L. Igual, O. Pujol, J. Moya, V. Violant, and M. T. Anguera. “A Gesture Recognition System for Detecting Behavioral Patterns of ADHD”. In: *IEEE Transactions on Cybernetics* 46.1 (2016), pp. 136–147. ISSN: 2168-2267. DOI: 10.1109/TCYB.2015.2396635.
- [10] Miguel Ángel Bautista, Antonio Hernández-Vela, Victor Ponce, Xavier Perez-Sala, Xavier Baró, Oriol Pujol, Cecilio Angulo, and Sergio Escalera. “Advances in Depth Image Analysis and Applications: International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers”. In: ed. by Xiaoyi Jiang, Olga Regina Pereira Bellon, Dmitry Goldgof, and Takeshi Oishi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Probability-Based Dynamic Time Warping for Gesture Recognition on RGB-D Data, pp. 126–135. ISBN: 978-3-642-40303-3. DOI: 10.1007/978-3-642-40303-3_14. URL: http://dx.doi.org/10.1007/978-3-642-40303-3_14.
- [11] Michel Beaudouin-Lafon. “Instrumental Interaction: An Interaction Model for Designing post-WIMP User Interfaces”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: ACM, 2000, pp. 446–453. ISBN: 1-58113-216-6. DOI: 10.1145/332040.332473. URL: <http://doi.acm.org/10.1145/332040.332473>.
- [12] Ana M. Bernardos, Xian Wang, Enrique García, Javier Portillo, and José R. Casar. “Using a Platform for Mobile Gesture-based Interaction to Control Smart Objects”. In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. SenSys '13. Roma, Italy: ACM, 2013, 39:1–39:2. ISBN: 978-1-4503-2027-6. DOI: 10.1145/2517351.2517411. URL: <http://doi.acm.org/10.1145/2517351.2517411>.
- [13] Ali Bigdelou, Loren Schwarz, and Nassir Navab. “An Adaptive Solution for Intra-operative Gesture-based Human-machine Interaction”. In: *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. IUI '12. Lisbon, Portugal: ACM, 2012, pp. 75–84. ISBN: 978-1-4503-1048-2. DOI: 10.1145/2166966.2166981. URL: <http://doi.acm.org/10.1145/2166966.2166981>.

- [14] S. Bodiroža, G. Doisy, and V. V. Hafner. “Position-invariant, real-time gesture recognition based on dynamic time warping”. In: *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*. 2013, pp. 87–88. DOI: 10.1109/HRI.2013.6483514.
- [15] Richard A. Bolt. ““Put-that-there”: Voice and Gesture at the Graphics Interface”. In: *SIGGRAPH Comput. Graph.* 14.3 (July 1980), pp. 262–270. ISSN: 0097-8930. DOI: 10.1145/965105.807503. URL: <http://doi.acm.org/10.1145/965105.807503>.
- [16] John Brooke et al. “SUS-A quick and dirty usability scale”. In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.
- [17] Matthias Budde, Matthias Berning, Christopher Baumgärtner, Florian Kinn, Timo Kopf, Sven Ochs, Frederik Reiche, Till Riedel, and Michael Beigl. “Point & Control – Interaction in Smart Environments: You Only Click Twice”. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. UbiComp ’13 Adjunct. Zurich, Switzerland: ACM, 2013, pp. 303–306. ISBN: 978-1-4503-2215-7. DOI: 10.1145/2494091.2494184. URL: <http://doi.acm.org/10.1145/2494091.2494184>.
- [18] A. Cenedese, G. A. Susto, G. Belgioioso, G. I. Cirillo, and F. Fraccaroli. “Home Automation Oriented Gesture Classification From Inertial Measurements”. In: *IEEE Transactions on Automation Science and Engineering* 12.4 (2015), pp. 1200–1210. ISSN: 1545-5955. DOI: 10.1109/TASE.2015.2473659.
- [19] Debaleena Chattopadhyay and Davide Bolchini. “Touchless Circular Menus: Toward an Intuitive UI for Touchless Interactions with Large Displays”. In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. AVI ’14. Como, Italy: ACM, 2014, pp. 33–40. ISBN: 978-1-4503-2775-6. DOI: 10.1145/2598153.2598181. URL: <http://doi.acm.org/10.1145/2598153.2598181>.
- [20] M. Chen, G. AlRegib, and B. H. Juang. “Feature Processing and Modeling for 6D Motion Gesture Recognition”. In: *IEEE Transactions on Multimedia* 15.3 (2013), pp. 561–571. ISSN: 1520-9210. DOI: 10.1109/TMM.2012.2237024.
- [21] Jun Cheng, Can Xie, Wei Bian, and Dacheng Tao. “Feature Fusion for 3D Hand Gesture Recognition by Learning a Shared Hidden Space”. In: *Pattern Recogn. Lett.* 33.4 (Mar. 2012), pp. 476–484. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2010.12.009. URL: <http://dx.doi.org/10.1016/j.patrec.2010.12.009>.
- [22] CADOZ Claude. “Les réalités virtuelles”. In: *Dominos/Flammarion* (1994).
- [23] Enrico Costanza, Joel E. Fischer, James A. Colley, Tom Rodden, Sarvapali D. Ramchurn, and Nicholas R. Jennings. “Doing the Laundry with Agents: A Field Trial of a Future Smart Energy System in the Home”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 813–822. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557167. URL: <http://doi.acm.org/10.1145/2556288.2557167>.

BIBLIOGRAPHY

- [24] H. Cuevas, A. M. Bernardos, X. Wang, L. Bergesio, and J. R. Casar. “Integrating gesture-based identification in context-aware applications: a system approach”. In: *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2014, pp. 257–264. DOI: 10.1109/WiMOB.2014.6962180.
- [25] William Delamare, Céline Coutrix, and Laurence Nigay. “Designing Guiding Systems for Gesture-based Interaction”. In: *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. EICS ’15. Duisburg, Germany: ACM, 2015, pp. 44–53. ISBN: 978-1-4503-3646-8. DOI: 10.1145/2774225.2774847. URL: <http://doi.acm.org/10.1145/2774225.2774847>.
- [26] *Deus Ex Aria smart strap*. URL: <https://www.ariawearable.com/>.
- [27] Tilman Dingler, Markus Funk, and Florian Alt. “Interaction Proxemics: Combining Physical Spaces for Seamless Gesture Interaction”. In: *Proceedings of the 4th International Symposium on Pervasive Displays*. PerDis ’15. Saarbruecken, Germany: ACM, 2015, pp. 107–114. ISBN: 978-1-4503-3608-6. DOI: 10.1145/2757710.2757722. URL: <http://doi.acm.org/10.1145/2757710.2757722>.
- [28] Jun-Hyeong Do, Jin-Woo Jung, Sung Hoon Jung, Hyoyoung Jang, and Zeungnam Bien. “MICA1 2006: Advances in Artificial Intelligence: 5th Mexican International Conference on Artificial Intelligence, Apizaco, Mexico, November 13-17, 2006. Proceedings”. In: ed. by Alexander Gelbukh and Carlos Alberto Reyes-Garcia. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. Advanced Soft Remote Control System Using Hand Gesture, pp. 745–755. ISBN: 978-3-540-49058-6. DOI: 10.1007/11925231_71. URL: http://dx.doi.org/10.1007/11925231_71.
- [29] *Exploride website*. <https://exploride.com>. Accessed: 2015-09-28.
- [30] *Eyesight - Gesture Control with Smartphone-based VR*. <http://www.techtimes.com/articles/159204/20160518/eyesight-showcases-gesture-control-with-smartphone-based-vr-video.htm>. Accessed: 2016-06-27.
- [31] D. Frolova, H. Stern, and S. Berman. “Most Probable Longest Common Subsequence for Recognition of Gesture Character Input”. In: *IEEE Transactions on Cybernetics* 43.3 (2013), pp. 871–880. ISSN: 2168-2267. DOI: 10.1109/TSMCB.2012.2217324.
- [32] Jože Guna, Emilija Stojmenova, Artur Lugmayr, Iztok Humar, and Matevž Pogačnik. “User identification approach based on simple gestures”. In: *Multimedia Tools and Applications* 71.1 (2013), pp. 179–194. ISSN: 1573-7721. DOI: 10.1007/s11042-013-1635-1. URL: <http://dx.doi.org/10.1007/s11042-013-1635-1>.
- [33] Edward T Hall. “The silent language.” In: (1973).

- [34] Eiji Hayashi, Manuel Maas, and Jason I. Hong. “Wave to Me: User Identification Using Body Lengths and Natural Gestures”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 3453–3462. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557043. URL: <http://doi.acm.org/10.1145/2556288.2557043>.
- [35] S. Helal and S. Tarkoma. “Smart Spaces [Guest editors’ introduction]”. In: *IEEE Pervasive Computing* 14.2 (2015), pp. 22–23. ISSN: 1536-1268. DOI: 10.1109/MPRV.2015.40.
- [36] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O’Hara, Gavin Smyth, and William Buxton. “Pre-Touch Sensing for Mobile Interaction”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. Santa Clara, California, USA: ACM, 2016, pp. 2869–2881. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858095. URL: <http://doi.acm.org/10.1145/2858036.2858095>.
- [37] Y. L. Hsu, C. L. Chu, Y. J. Tsai, and J. S. Wang. “An Inertial Pen With Dynamic Time Warping Recognizer for Handwriting and Gesture Recognition”. In: *IEEE Sensors Journal* 15.1 (2015), pp. 154–163. ISSN: 1530-437X. DOI: 10.1109/JSEN.2014.2339843.
- [38] Anil K. Jain, Patrick Flynn, and Arun A. Ross. *Handbook of Biometrics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN: 038771040X.
- [39] Soonmook Jeong, Jungdong Jin, Taehoun Song, Keyho Kwon, and Jae Wook Jeon. “Single-camera dedicated television control system using gesture drawing”. In: *Consumer Electronics, IEEE Transactions on* 58.4 (2012), pp. 1129–1137. ISSN: 0098-3063. DOI: 10.1109/TCE.2012.6414977.
- [40] Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. “GesText: Accelerometer-based Gestural Text-entry Systems”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’10. Atlanta, Georgia, USA: ACM, 2010, pp. 2173–2182. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753655. URL: <http://doi.acm.org/10.1145/1753326.1753655>.
- [41] Alvin Jude, G. Michael Poor, and Darren Guinness. “Personal Space: User Defined Gesture Space for GUI Interaction”. In: *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI EA ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 1615–1620. ISBN: 978-1-4503-2474-8. DOI: 10.1145/2559206.2581242. URL: <http://doi.acm.org/10.1145/2559206.2581242>.
- [42] Maria Karam. “PhD Thesis: A framework for research and design of gesture-based human-computer interactions”. PhD thesis. University of Southampton, 2006.

BIBLIOGRAPHY

- [43] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and SergioDi Marca. “Accelerometer-based gesture control for a design environment”. English. In: *Personal and Ubiquitous Computing* 10.5 (2006), pp. 285–299. ISSN: 1617-4909. DOI: 10.1007/s00779-005-0033-8. URL: <http://dx.doi.org/10.1007/s00779-005-0033-8>.
- [44] Adam Kendon. “Some relationships between body motion and speech”. In: *Studies in dyadic communication* 7.177 (1972), p. 90.
- [45] Mi Jeong Kim, Myoung Won Oh, Myung Eun Cho, Hyungsoo Lee, and Jeong Tai Kim. “A critical review of user studies on healthy smart homes”. In: *Indoor and Built Environment* (2012), p. 1420326X12469733.
- [46] *Kinect for windows*. <https://dev.windows.com/en-us/kinect>. Accessed: 2015-09-28.
- [47] Michael Kipp. *ANVIL video annotation tool*. URL: <http://www.anvil-software.org/>.
- [48] Ming Hsiao Ko, Geoff West, Svetha Venkatesh, and Mohan Kumar. “Using Dynamic Time Warping for Online Temporal Fusion in Multisensor Systems”. In: *Inf. Fusion* 9.3 (July 2008), pp. 370–388. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2006.08.002. URL: <http://dx.doi.org/10.1016/j.inffus.2006.08.002>.
- [49] Kazuki Kobayashi and Tomoaki Ohtsuki. “User Identification Based on Toothbrushing Information Using Three-axis Accelerometer”. In: *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. UbiComp/ISWC’15 Adjunct. Osaka, Japan: ACM, 2015, pp. 129–132. ISBN: 978-1-4503-3575-1. DOI: 10.1145/2800835.2800889. URL: <http://doi.acm.org/10.1145/2800835.2800889>.
- [50] Barry Kollee, Sven Kratz, and Anthony Dunnigan. “Exploring Gestural Interaction in Smart Spaces Using Head Mounted Devices with Ego-centric Sensing”. In: *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*. SUI ’14. Honolulu, Hawaii, USA: ACM, 2014, pp. 40–49. ISBN: 978-1-4503-2820-3. DOI: 10.1145/2659766.2659781. URL: <http://doi.acm.org/10.1145/2659766.2659781>.
- [51] Kishore Reddy Konda, Achim Königs, Hannes Schulz, and Dirk Schulz. “Real Time Interaction with Mobile Robots Using Hand Gestures”. In: *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’12. Boston, Massachusetts, USA: ACM, 2012, pp. 177–178. ISBN: 978-1-4503-1063-5. DOI: 10.1145/2157689.2157743. URL: <http://doi.acm.org/10.1145/2157689.2157743>.

- [52] Louis Kratz, Daniel Morris, and T. Scott Saponas. “Making Gestural Input from Arm-worn Inertial Sensors More Practical”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. Austin, Texas, USA: ACM, 2012, pp. 1747–1750. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208304. URL: <http://doi.acm.org/10.1145/2207676.2208304>.
- [53] Sven Kratz and Michael Rohs. “Protractor3D: A Closed-form Solution to Rotation-invariant 3D Gestures”. In: *Proceedings of the 16th International Conference on Intelligent User Interfaces*. IUI ’11. Palo Alto, CA, USA: ACM, 2011, pp. 371–374. ISBN: 978-1-4503-0419-1. DOI: 10.1145/1943403.1943468. URL: <http://doi.acm.org/10.1145/1943403.1943468>.
- [54] Sven Kratz, Michael Rohs, and Georg Essl. “Combining Acceleration and Gyroscope Data for Motion Gesture Recognition Using Classifiers with Dimensionality Constraints”. In: *Proceedings of the 2013 International Conference on Intelligent User Interfaces*. IUI ’13. Santa Monica, California, USA: ACM, 2013, pp. 173–178. ISBN: 978-1-4503-1965-2. DOI: 10.1145/2449396.2449419. URL: <http://doi.acm.org/10.1145/2449396.2449419>.
- [55] Christian Kray, Daniel Nesbitt, John Dawson, and Michael Rohs. “User-defined Gestures for Connecting Mobile Phones, Public Displays, and Tabletops”. In: *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI ’10. Lisbon, Portugal: ACM, 2010, pp. 239–248. ISBN: 978-1-60558-835-3. DOI: 10.1145/1851600.1851640. URL: <http://doi.acm.org/10.1145/1851600.1851640>.
- [56] Bastian Kriesten, Christian Mertes, René Tünnermann, and Thomas Hermann. “Unobtrusively Controlling and Linking Information and Services in Smart Environments”. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. NordiCHI ’10. Reykjavik, Iceland: ACM, 2010, pp. 276–285. ISBN: 978-1-60558-934-3. DOI: 10.1145/1868914.1868948. URL: <http://doi.acm.org/10.1145/1868914.1868948>.
- [57] Christine Kühnel, Tilo Westermann, Benjamin Weiss, and Sebastian Möller. “Evaluating Multimodal Systems: A Comparison of Established Questionnaires and Interaction Parameters”. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. NordiCHI ’10. Reykjavik, Iceland: ACM, 2010, pp. 286–294. ISBN: 978-1-60558-934-3. DOI: 10.1145/1868914.1868949. URL: <http://doi.acm.org/10.1145/1868914.1868949>.
- [58] Christine Kühnel, Tilo Westermann, Fabian Hemmert, Sven Kratz, Alexander Müller, and Sebastian Möller. “I’m home: Defining and evaluating a gesture set for smart-home control”. In: *International Journal of Human-Computer Studies* 69.11 (2011), pp. 693–704.
- [59] *Laerd Statistics: SPSS Statistics Tutorials and Statistical Guides*. <https://statistics.laerd.com/>. Accessed: 2016-08-11.

BIBLIOGRAPHY

- [60] Bettina Laugwitz, Theo Held, and Martin Schrepp. “HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings”. In: ed. by Andreas Holzinger. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Chap. Construction and Evaluation of a User Experience Questionnaire, pp. 63–76. ISBN: 978-3-540-89350-9. DOI: 10.1007/978-3-540-89350-9_6. URL: http://dx.doi.org/10.1007/978-3-540-89350-9_6.
- [61] *LeapMotion - API Overview*. https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html?proglang=current. Accessed: 2015-09-28.
- [62] Sang-Su Lee, Jeonghun Chae, Hyunjeong Kim, Youn-kyung Lim, and Kun-pyo Lee. “Towards More Natural Digital Content Manipulation via User Freehand Gestural Interaction in a Living Room”. In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’13. Zurich, Switzerland: ACM, 2013, pp. 617–626. ISBN: 978-1-4503-1770-2. DOI: 10.1145/2493432.2493480. URL: <http://doi.acm.org/10.1145/2493432.2493480>.
- [63] Yun Li, Xiang Chen, Xu Zhang, Kongqiao Wang, and Z.J. Wang. “A Sign-Component-Based Framework for Chinese Sign Language Recognition Using Accelerometer and sEMG Data”. In: *Biomedical Engineering, IEEE Transactions on* 59.10 (2012), pp. 2695–2704. ISSN: 0018-9294. DOI: 10.1109/TBME.2012.2190734.
- [64] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. “uWave: Accelerometer based personalized gesture recognition and its applications”. In: *Pervasive Mob. Comput.* 5.6 (Dec. 2009), pp. 657–675. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2009.07.007. URL: <http://dx.doi.org/10.1016/j.pmcj.2009.07.007>.
- [65] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz. “Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition”. In: *IEEE Sensors Journal* 14.6 (2014), pp. 1898–1903. ISSN: 1530-437X. DOI: 10.1109/JSEN.2014.2306094.
- [66] Dennis Lupiana, Zanifa Omary, Fredrick Mtenzi, and Ciaran O’Driscoll. “Smart Spaces in Ubiquitous Computing”. In: *4th International Conference on Information Technology*. 2009.
- [67] Christian Mayer, Patrick Pogscheba, Dionysios Marinos, Björn Wöldecke, and Christian Geiger. “An Audio-visual Music Installation with Dichotomous User Interactions”. In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*. ACE ’14. Funchal, Portugal: ACM, 2014, 55:1–55:6. ISBN: 978-1-4503-2945-3. DOI: 10.1145/2663806.2663842. URL: <http://doi.acm.org/10.1145/2663806.2663842>.
- [68] S. Melzi, S. Napoli, A. Pollastro, and S. Fowler. “An Heterogeneous wireless networks testbed for Smart Environment scenarios”. In: *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2014, pp. 173–177. DOI: 10.1109/CAMAD.2014.7033229.

- [69] O. Mendels, H. Stern, and S. Berman. “User Identification for Home Entertainment Based on Free-Air Hand Motion Signatures”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.11 (2014), pp. 1461–1473. ISSN: 2168-2216. DOI: 10.1109/TSMC.2014.2329652.
- [70] S. Mitra and T. Acharya. “Gesture Recognition: A Survey”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.3 (2007), pp. 311–324. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.893280.
- [71] Meredith Ringel Morris. “Web on the Wall: Insights from a Multimodal Interaction Elicitation Study”. In: *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’12. Cambridge, Massachusetts, USA: ACM, 2012, pp. 95–104. ISBN: 978-1-4503-1209-7. DOI: 10.1145/2396636.2396651. URL: <http://doi.acm.org/10.1145/2396636.2396651>.
- [72] C. Myers, L. Rabiner, and A. Rosenberg. “Performance tradeoffs in dynamic time warping algorithms for isolated word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.6 (1980), pp. 623–635. ISSN: 0096-3518. DOI: 10.1109/TASSP.1980.1163491.
- [73] *Myo armband*. <https://www.myo.com/>. Accessed: 2016-08-31.
- [74] Michael Nebeling, Alexander Huber, David Ott, and Moira C. Norrie. “Web on the Wall Reloaded: Implementation, Replication and Refinement of User-Defined Interaction Sets”. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’14. Dresden, Germany: ACM, 2014, pp. 15–24. ISBN: 978-1-4503-2587-5. DOI: 10.1145/2669485.2669497. URL: <http://doi.acm.org/10.1145/2669485.2669497>.
- [75] Michael Nielsen, Moritz Störring, ThomasB. Moeslund, and Erik Granum. “A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI”. English. In: *Gesture-Based Communication in Human-Computer Interaction*. Ed. by Antonio Camurri and Gualtiero Volpe. Vol. 2915. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 409–420. ISBN: 978-3-540-21072-6. DOI: 10.1007/978-3-540-24598-8_38. URL: http://dx.doi.org/10.1007/978-3-540-24598-8_38.
- [76] C. D. Nugent, D. D. Finlay, P. Fiorini, Y. Tsumaki, and E. Prassler. “Editorial Home Automation as a Means of Independent Living”. In: *IEEE Transactions on Automation Science and Engineering* 5.1 (2008), pp. 1–9. ISSN: 1545-5955. DOI: 10.1109/TASE.2007.912247.
- [77] Alexandra-Gwyn Paetz, Elisabeth Dütschke, and Wolf Fichtner. “Smart Homes as a Means to Sustainable Energy Consumption: A Study of Consumer Perceptions”. In: *Journal of Consumer Policy* 35.1 (2012), pp. 23–41. ISSN: 1573-0700. DOI: 10.1007/s10603-011-9177-2. URL: <http://dx.doi.org/10.1007/s10603-011-9177-2>.
- [78] *Project Soli - Google*. <https://atap.google.com/soli/>. Accessed: 2016-06-27.

BIBLIOGRAPHY

- [79] ASM Mahfujur Rahman, M. Anwar Hossain, Jorge Parra, and Abdulmotaleb El Sadik. “Motion-path Based Gesture Interaction with Smart Home Services”. In: *Proceedings of the 17th ACM International Conference on Multimedia*. MM ’09. Beijing, China: ACM, 2009, pp. 761–764. ISBN: 978-1-60558-608-3. DOI: 10.1145/1631272.1631407. URL: <http://doi.acm.org/10.1145/1631272.1631407>.
- [80] Siddharth S. Rautaray and Anupam Agrawal. “Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey”. In: *Artif. Intell. Rev.* 43.1 (Jan. 2015), pp. 1–54. ISSN: 0269-2821. DOI: 10.1007/s10462-012-9356-9. URL: <http://dx.doi.org/10.1007/s10462-012-9356-9>.
- [81] *RealSense camera*. <http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html>. Accessed: 2016-05-05.
- [82] Julie Rico and Stephen Brewster. “Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’10. Atlanta, Georgia, USA: ACM, 2010, pp. 887–896. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753458. URL: <http://doi.acm.org/10.1145/1753326.1753458>.
- [83] *Ring ZERO*. <http://logbar.jp/ring/en/zerowhite>. Accessed: 2016-05-05.
- [84] Gustavo Alberto Rovelo Ruiz, Davy Vanacken, Kris Luyten, Francisco Abad, and Emilio Camahort. “Multi-viewer Gesture-based Interaction for Omni-directional Video”. In: *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 4077–4086. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557113. URL: <http://doi.acm.org/10.1145/2556288.2557113>.
- [85] Jaime Ruiz, Yang Li, and Edward Lank. “User-defined Motion Gestures for Mobile Interaction”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. Vancouver, BC, Canada: ACM, 2011, pp. 197–206. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1978971. URL: <http://doi.acm.org/10.1145/1978942.1978971>.
- [86] Hiroaki Sakoe and Seibi Chiba. “Readings in Speech Recognition”. In: ed. by Alex Waibel and Kai-Fu Lee. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990. Chap. Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pp. 159–165. ISBN: 1-55860-124-4. URL: <http://dl.acm.org/citation.cfm?id=108235.108244>.
- [87] Xiaohui Shen, Gang Hua, Lance Williams, and Ying Wu. “Dynamic Hand Gesture Recognition: An Exemplar-based Approach from Motion Divergence Fields”. In: *Image Vision Comput.* 30.3 (Mar. 2012), pp. 227–235. ISSN: 0262-8856. DOI: 10.1016/j.imavis.2011.11.003. URL: <http://dx.doi.org/10.1016/j.imavis.2011.11.003>.

- [88] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. “The gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring”. In: *Wearable Computers, The Fourth International Symposium on*. 2000, pp. 87–94. DOI: 10.1109/ISWC.2000.888469.
- [89] S. Suresh and P. V. Sruthi. “A review on smart home technology”. In: *2015 Online International Conference on Green Engineering and Technologies (IC-GET)*. 2015, pp. 1–3. DOI: 10.1109/GET.2015.7453832.
- [90] *Template for the interaction graphs and interaction table*. <https://www.lri.fr/~mbl/Stanford/CS477/handouts/course7-interaction-table.pdf>. Accessed: 2016-08-12.
- [91] Cuong Tran and M.M. Trivedi. “3-D Posture and Gesture Recognition for Interactivity in Smart Spaces”. In: *Industrial Informatics, IEEE Transactions on* 8.1 (2012), pp. 178–187. ISSN: 1551-3203. DOI: 10.1109/TII.2011.2172450.
- [92] Matthew Turk. “Gesture recognition”. In: *Computer Vision: A Reference Guide* (2014), pp. 346–349.
- [93] Radu-Daniel Vatavu. “Nomadic Gestures: A Technique for Reusing Gesture Commands for Frequent Ambient Interactions”. In: *J. Ambient Intell. Smart Environ.* 4.2 (Apr. 2012), pp. 79–93. ISSN: 1876-1364. URL: <http://dl.acm.org/citation.cfm?id=2350758.2350765>.
- [94] Radu-Daniel Vatavu. “User-defined Gestures for Free-hand TV Control”. In: *Proceedings of the 10th European Conference on Interactive Tv and Video*. EuroITV ’12. Berlin, Germany: ACM, 2012, pp. 45–48. ISBN: 978-1-4503-1107-6. DOI: 10.1145/2325616.2325626. URL: <http://doi.acm.org/10.1145/2325616.2325626>.
- [95] Robert Walter, Gilles Bailly, and Jörg Müller. “StrikeAPose: Revealing Mid-air Gestures on Public Displays”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’13. Paris, France: ACM, 2013, pp. 841–850. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2470774. URL: <http://doi.acm.org/10.1145/2470654.2470774>.
- [96] Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng. “One-shot Learning Gesture Recognition from RGB-D Data Using Bag of Features”. In: *J. Mach. Learn. Res.* 14.1 (Jan. 2013), pp. 2549–2582. ISSN: 1532-4435. URL: <http://dl.acm.org/citation.cfm?id=2567709.2567743>.
- [97] Jeen-Shing Wang and Fang-Chen Chuang. “An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition”. In: *Industrial Electronics, IEEE Transactions on* 59.7 (2012), pp. 2998–3007. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2167895.
- [98] Xian Wang and Ana M. Bernardos. *Questionnaire of the user study*. 2015. URL: <http://goo.gl/forms/4kCi6gTXUc>.

BIBLIOGRAPHY

- [99] Xian Wang, Ana M. Bernardos, Juan A. Besada, Eduardo Metola, and José R. Casar. “A gesture-based method for natural interaction in smart spaces”. In: *JAISE* 7.4 (2015), pp. 535–562. DOI: 10.3233/AIS-150325. URL: <http://dx.doi.org/10.3233/AIS-150325>.
- [100] Xian Wang, Paula Tarrío, Eduardo Metola, Ana M. Bernardos, and José R. Casar. “Gesture Recognition Using Mobile Phone’s Inertial Sensors”. In: *Distributed Computing and Artificial Intelligence*. Ed. by Sigeru Omatu, Juan F. De Paz Santana, Sara Rodríguez González, Jose M. Molina, Ana M. Bernardos, and Juan M. Corchado Rodríguez. Vol. 151. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, 2012, pp. 173–184. ISBN: 978-3-642-28764-0. DOI: 10.1007/978-3-642-28765-7_21. URL: http://dx.doi.org/10.1007/978-3-642-28765-7_21.
- [101] Y. Wang, C. Yang, X. Wu, S. Xu, and H. Li. “Kinect Based Dynamic Hand Gesture Recognition Algorithm Research”. In: *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*. Vol. 1. 2012, pp. 274–279. DOI: 10.1109/IHMSC.2012.76.
- [102] M. Weiser. “Hot topics-ubiquitous computing”. In: *Computer* 26.10 (1993), pp. 71–72. ISSN: 0018-9162. DOI: 10.1109/2.237456.
- [103] Julie R. Williamson, Stephen Brewster, and Rama Vennelakanti. “Mo!Games: Evaluating Mobile Gestures in the Wild”. In: *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*. ICMI ’13. Sydney, Australia: ACM, 2013, pp. 173–180. ISBN: 978-1-4503-2129-7. DOI: 10.1145/2522848.2522874. URL: <http://doi.acm.org/10.1145/2522848.2522874>.
- [104] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. “User-defined Gestures for Surface Computing”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA: ACM, 2009, pp. 1083–1092. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518866. URL: <http://doi.acm.org/10.1145/1518701.1518866>.
- [105] D. Wu, F. Zhu, and L. Shao. “One shot learning gesture recognition from RGBD images”. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. 2012, pp. 7–12. DOI: 10.1109/CVPRW.2012.6239179.
- [106] J. Wu, J. Konrad, and P. Ishwar. “Dynamic time warping for gesture-based user identification and authentication with Kinect”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 2371–2375. DOI: 10.1109/ICASSP.2013.6638079.
- [107] Xingyu Wu, Xia Mao, Lijiang Chen, and Yuli Xue. “Trajectory-based view-invariant hand gesture recognition by fusing shape and orientation”. In: *Computer Vision, IET* 9.6 (2015), pp. 797–805.

- [108] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. “Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch”. In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. HotMobile '15. Santa Fe, New Mexico, USA: ACM, 2015, pp. 9–14. ISBN: 978-1-4503-3391-7. DOI: 10.1145/2699343.2699350. URL: <http://doi.acm.org/10.1145/2699343.2699350>.
- [109] R. Xu, S. Zhou, and W. J. Li. “MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition”. In: *IEEE Sensors Journal* 12.5 (2012), pp. 1166–1173. ISSN: 1530-437X. DOI: 10.1109/JSEN.2011.2166953.
- [110] Ying Yin and Randall Davis. “Gesture Spotting and Recognition Using Saliency Detection and Concatenated Hidden Markov Models”. In: *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*. ICMI '13. Sydney, Australia: ACM, 2013, pp. 489–494. ISBN: 978-1-4503-2129-7. DOI: 10.1145/2522848.2532588. URL: <http://doi.acm.org/10.1145/2522848.2532588>.
- [111] Haikuo Zhang, Wenjun Wu, and Yihua Lou. “PRICAI 2014: Trends in Artificial Intelligence: 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, December 1-5, 2014. Proceedings”. In: ed. by Duc-Nghia Pham and Seong-Bae Park. Cham: Springer International Publishing, 2014. Chap. A Personalized Gesture Interaction System with User Identification Using Kinect, pp. 614–626. ISBN: 978-3-319-13560-1. DOI: 10.1007/978-3-319-13560-1_49. URL: http://dx.doi.org/10.1007/978-3-319-13560-1_49.
- [112] Xu Zhang, Xiang Chen, Yun Li, V. Lantz, Kongqiao Wang, and Jihai Yang. “A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors”. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 41.6 (2011), pp. 1064–1076. ISSN: 1083-4427. DOI: 10.1109/TSMCA.2011.2116004.