Sensor Fusion with Gaussian Processes

Shimin Feng

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

SCHOOL OF COMPUTING SCIENCE

COLLEGE OF SCIENCE AND ENGINEERING
UNIVERSITY OF GLASGOW

October 2014

Abstract

This thesis presents a new approach to multi-rate sensor fusion for (1) user matching and (2) position stabilisation and lag reduction. The Microsoft Kinect sensor and the inertial sensors in a mobile device are fused with a Gaussian Process (GP) prior method. We present a Gaussian Process prior model-based framework for multisensor data fusion and explore the use of this model for fusing mobile inertial sensors and an external position sensing device.

The Gaussian Process prior model provides a principled mechanism for incorporating the low-sampling-rate position measurements and the high-sampling-rate derivatives in multi-rate sensor fusion, which takes account of the uncertainty of each sensor type. We explore the complementary properties of the Kinect sensor and the built-in inertial sensors in a mobile device and apply the GP framework for sensor fusion in the mobile human-computer interaction area.

The Gaussian Process prior model-based sensor fusion is presented as a principled probabilistic approach to dealing with position uncertainty and the lag of the system, which are critical for indoor augmented reality (AR) and other location-aware sensing applications. The sensor fusion helps increase the stability of the position and reduce the lag. This is of great benefit for improving the usability of a human-computer interaction system.

We develop two applications using the novel and improved GP prior model. (1) User matching and identification. We apply the GP model to identify individual users, by matching the observed Kinect skeletons with the sensed inertial data from their mobile devices. (2) Position stabilisation and lag reduction in a spatially aware display application for user performance improvement. We conduct a user study. Experimental results show the improved accuracy of target selection, and reduced delay from the sensor fusion system, allowing the users to acquire the target more rapidly, and with fewer errors in comparison with the Kinect filtered system. They also reported improved performance in subjective questions. The two applications can be combined seamlessly in a proxemic interaction system as identification of people and their positions in a room-sized environment plays a key role in proxemic interactions.

Acknowledgements

I am grateful to my supervisor Prof. Roderick Murray-Smith. He has given me this opportunity to work in this area. I would like to express my deep and sincere gratitude for his guidance. His expertise, patience and inspirational ideas made possible any progress that was made. He reviewed my work carefully and provided many hints that helped to improve the quality of my thesis. I also want to thank my second supervisor Dr. Alessandro Vinciarelli for his support and fruitful discussions.

I would like to thank the entire Inference, Dynamics and Interaction group for enabling me to work in such a pleasant atmosphere. I gratefully acknowledge the contributions of Andrew Ramsay with whom I had an opportunity to work with. He is always ready to help and has given me a lot of support during my study. Thank Dr. John Williamson and Dr. Andy Crossan for their helpful discussions. Thank Dr. Simon Rogers for his support on machine learning. The machine learning class taught me a lot. Many people helped me during my PhD study. I also want to thank Melissa Quek, Lauren Norrie, Daniel Boland, Daryl Weir, and some other people, to whom I apologize that I forgot to name. The life and study here is fun!

This research has been jointly funded by University of Glasgow and China Scholarship Council. These are hereby gratefully acknowledged. I sincerely appreciate the help of the administration staff in the School of Computing Science and College of Science and Engineering office during my PhD application and the study process. I would like to express my gratitude to Prof. Jonathan Cooper for his kind assistance. I also want to express my deep thankfulness towards Associate Prof. Qing Guan and Prof. Qicong Peng for their support during my graduate study and the PhD application process.

Finally, I am grateful to my parents and want to express my deep gratitude for your love, support and encouragement!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This introductory chapter gives an introduction to context-aware sensing by proposing scenarios of two people using a proxemic interaction system in a room, and then presents the research problems and motivations. We briefly discuss the problems of position sensing for indoor mobile Augmented Reality (AR) and other location-aware sensing applications. We argue the need for dealing with the uncertainty of different sensor measurements and the latency in the conventional Kinect system. We discuss the complementary properties of the Kinect sensor and mobile inertial sensors, and summarise the sensor fusion theme that will run through this thesis. Meanwhile, we highlight the role of Gaussian Processes (GPs) in dynamical system modelling, and finally present the contributions and the outline of the thesis.

1.1 Introduction

In recent years, advanced sensors have become ubiquitous. The human-computer interaction systems are composed of a variety of sensors. These sensors work at a range of sampling rates and often have very different noise characteristics. They may measure different derivatives of measurands (e.g. position, velocity, acceleration) in the world. If we can fuse information from such systems in an efficient and principled manner, we can potentially improve the context sensing capability of the system without adding extra sensing hardware. A concrete example of this is integration of inertial sensor data from mobile devices such as phones or tablets with position sensing from an embedded Microsoft Kinect sensor (Wikipedia, 2014; Livingston et al., 2012), but the same principle can be found in many systems. The Microsoft Kinect is a human motion sensing device that can be used for human body tracking, and is low-cost, portable and unobtrusive in a room. If the Kinect can sense multiple people in the room and each has a device in the hand or pocket, which person car-

ries which device? If we successfully associate a person with a device, can the inertial sensor data sensed by this device be used to improve the person's skeleton position tracking?

The identification and tracking of people in an indoor environment plays an important role in human-computer interaction systems. When there are multiple persons in the room, the identification of people allows the system to provide personalized services to each of them. The tracking of a person using a handheld device is critical to the effective use of a mobile augmented reality (AR) or a spatially aware display application.

Identification of people and their positions in a room-sized environment plays a key role in proxemic interactions. *Proxemics* is the theory proposed by Edward Hall about people's understanding and use of interpersonal distances to mediate their interactions with others (Hall & Hall, 1969). Greenberg et al. operationalized the concept of proxemic within ubiquitous computing and proposed five proxemic dimensions including: distance, orientation, identity, movement and location for proxemic interaction (Ballendat et al., 2010; Marquardt et al., 2011; Greenberg et al., 2011). Knowledge of the identity of a person, or a device is critical in proxemic-aware applications (Ballendat et al., 2010).

When several users are in a sensor-augmented room (e.g. using a Microsoft Kinect depth sensor) and each of them carries a sensor-enhanced mobile device (e.g. with accelerometers), it is possible to find the matching relationship between individual users and the mobile devices. A personal device can then provide the means to associate an identity with a tracked user (Ackad et al., 2012), implicitly providing a way for user identification through user matching, i.e. finding the correlation between the multiple skeletons (users) and the mobile devices. In practice, this can be challenging because the different types of sensors have different noise and sampling properties, as well as measuring different physical quantities. In this work, we apply a novel and improved Gaussian Process prior model to fuse the low-sampling-rate position measurements sensed by the Kinect and the higher frequency acceleration measured by the mobile inertial sensors. Firstly, the sensor fusion combines data from multiple sensors (Hall & Llinas, 1997), and can be applied to improve the accuracy and speed of measuring the match between a set of users' skeletons and a set of candidate mobile devices. This is the first application, i.e. user matching and identification. Secondly, the Kinect sensor data and the mobile inertial sensor data can be fused to improve the accuracy of the Kinect skeleton joint position tracking and to reduce the lag of the system. This enables the user to better interact in a spatially aware display or augmented reality (AR) application in a room. This is the second application.

User Matching Scenario

To illustrate this, we propose a scenario of two people using a proxemic interaction system in a room, as shown in Figure 1.1. The system can display the users' favorite books and

also make personalized recommendations for them (Funk et al., 2010). The Kinect and the interactive vertical display surfaces are fixed on the wall. Two people (Jim and Tom) walk into the room. Each carries a mobile device in the trousers pocket or in the hand. Jim likes classic literature and Tom likes contemporary books. The Kinect starts tracking and assigns a user ID to each person. Jim is user 1 and Tom is user 2. As a personal device can provide the means to associate an identity with a tracked user (Ackad et al., 2012) and the system can detect the identities of the personal devices, we know who the user is if we can link a particular skeleton with one of the mobile devices. This enables the system to provide a personalized service when a user approaches a display surface through proximity interaction.

Designing technologies that are embedded in people's everyday lives plays an important role in context-aware applications (Bilandzic & Foth, 2012). The process mentioned above may involve a variety of people's everyday movements, including moving with a device in the trousers pocket, the subtle hand movements or walking with a device held in the hand (Barnard et al., 2005). Vogel & Balakrishnan (2004) proposed an interaction framework for ambient displays that support the transition from implicit to explicit interaction by identifying individual users through registered marker sets, and argued the need for marker-free tracking systems and user identification techniques.

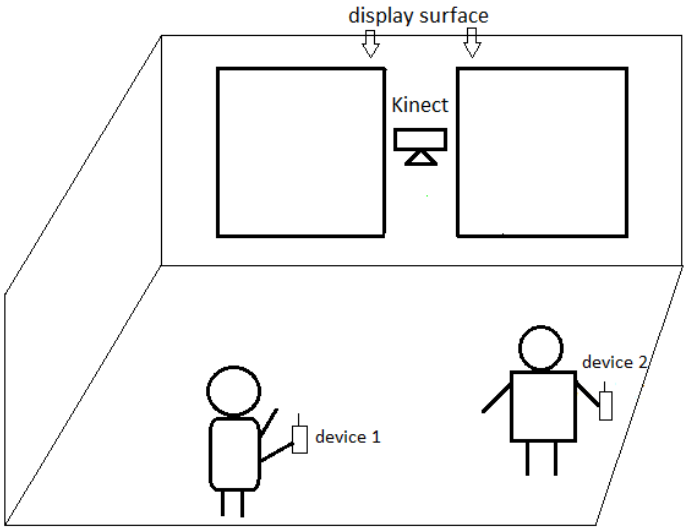


Figure 1.1: A scenario of two people using a proxemic interaction system in a room. Proxemic interaction relates the two users to their personal devices by matching the motion sensed by the Kinect with the motion sensed by the devices when they carry the devices and move in the field of the Kinect's view. The personalized content will be displayed when the user approaches the surface as the system knows the identity of the user through matching the user with the personal device. The device can be held in the hand, as shown in the figure, or in a trouser pocket. The user matching application will be presented in Chapter 5.

Location-Aware Sensing Application Scenario

In the above scenario, the system can achieve user matching and identification implicitly, and customise services appropriately for them. A spatially aware display or an augmented reality (AR) application in the room is an example of a proxemic-aware application, which enables the user to use explicit hand motion-based interaction to acquire information in this room. This is illustrated in Figure 1.2. Jim walks a few steps forward with the device held in the hand. When he approaches the vertical screen, more contents, e.g. book category labels, become visible to him as it zooms out. At certain spatial locations near the surface, we can design a spatially aware display application that links the digital books with the spatial locations. This enables Jim to browse the detailed content of a book by placing his device there.

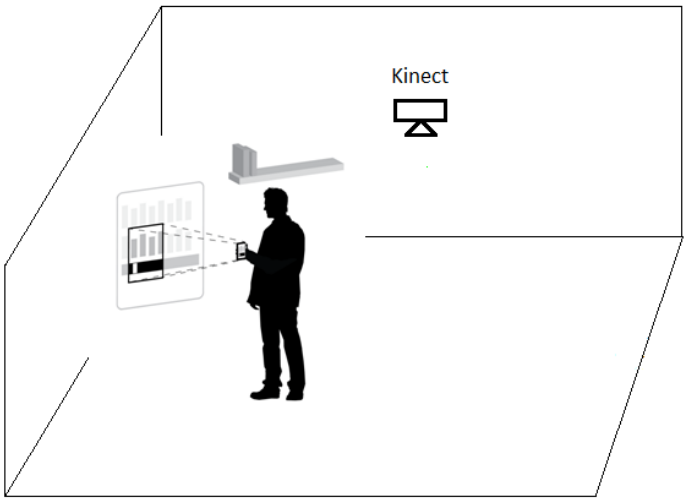


Figure 1.2: A scenario of a person (e.g. Jim) using a proxemic interaction system in a room. After user matching and identification in Figure 1.1, we can use the mobile device as an aiding sensor to augment the Kinect, stabilising the user's skeleton joint (e.g. hand) positions and reducing the latency of the conventional Kinect system in an augmented reality (AR) or a spatially aware display application, which can be a part of this proxemic interaction system.

An important issue in this proxemic interaction system is the accuracy of position tracking. In order to reduce the joint position uncertainty and improve the interaction performance and experience of the users (Jim and Tom), we proposed a sensor fusion approach to stabilising the hand position and reducing the lag of the system in the Kinect space by fusing the Kinect sensor and the mobile inertial sensors (Feng et al., 2014). After user matching, we can apply the acceleration sensed by Jim's device to compensate for the effects of position uncertainty and lag in Jim's skeleton tracking sensed by the conventional Kinect system, giving a smoother, more responsive experience.

1.2 Research Problems and Motivations

1.2.1 Research Problems

The identity and position of the user in an indoor environment is critical to the effective use of a proxemic-aware interaction system. The accuracy of position tracking and the responsiveness of an interaction system play a key role in a Kinect-based spatially aware display or mobile augmented reality (AR) application.

When there are multiple users in a room, we cannot determine the identity of each user with only a Kinect sensor. Besides, the two problems with the Microsoft Kinect skeleton tracking (Azimi, 2012) include:

1. The joint position uncertainty
2. The latency of the Kinect system

To address these problems, we need to apply sensor fusion techniques as the filtering techniques will induce lags. Multisensor data fusion requires interdisciplinary knowledge and techniques. We focus on building a Gaussian Process (GP) prior model to fuse the Kinect sensor and the built-in inertial sensors in a mobile device. This Gaussian Process prior model-based probabilistic approach helps improve the usability of a proxemic-aware system by improving the accuracy of state estimation and reducing the lag, i.e. the latency. Moreover, this model can be used to compute the joint log-likelihood of the low-sampling-rate position and the high-sampling-rate acceleration. The highest log-likelihood indicates the best match of the skeleton and the device. Thus, this is beneficial for user matching and identification.

The main applications include:

- Fusion of the Microsoft Kinect sensor and mobile inertial sensors for user matching and identification
- Fusion of the Microsoft Kinect sensor and mobile inertial sensors to improve the joint (e.g. hand) position estimation and reduce the lag of the system in a location-aware sensing application (spatially aware display)

In this thesis, we apply a novel and improved Gaussian Process prior model to fuse the low-sampling-rate position measurements sensed by the Kinect and the higher frequency acceleration measured by the mobile inertial sensors. Sensor fusion combines data from multiple sensors (Hall & Llinas, 1997), and can be applied for matching a particular user's skeleton with a mobile device. The first application of the sensor fusion system is user

matching, i.e. finding the correlation between the multiple skeletons and the mobile devices, presented in Chapter 5. The second application is to stabilise the joint (hand) position and reduce the lag in a spatially aware display application for user performance improvement, described in Chapter 6.

1.2.2 Research Motivations

In order to solve the accuracy and latency problems of the conventional Kinect system, we need additional sensors to augment the Kinect sensor. Location-aware sensing applications require the researchers to combine indoor position tracking devices and aiding sensors, and to fuse multiple sensor data. Firstly, we discuss the complementary sensing in a proxemic interaction system composed of a Kinect and mobile devices. In order to fuse multiple motion sensors, we need a multisensor data fusion method. We highlight the two key advantages of sensor fusion with Gaussian Processes (GPs), and discuss the two applications of the GP prior model-based sensor fusion.

The Kinect-augmented system can enhance a user's interaction through context-aware sensing, e.g. identify the user implicitly through the user's everyday movements and provide a personalized service on the screen. In addition, the Kinect-based sensor fusion system can improve the user's spatial interaction experience by stabilising the user's hand position and reducing the lag of the tracking system in a spatially aware display application.

Complementary Sensing in Proxemic Interaction

Sensors provide a way to capture proxemic data in a proxemic-aware system. The Microsoft Kinect is a successful sensor for sensing human skeleton joints positions (Greenberg et al., 2011). The Kinect skeleton tracking opens a rich design space for Human-Computer Interaction (HCI) researchers. However, for human motion tracking with a Kinect, the uncertainty in position measurement limits the styles of interactions that are possible (Casiez et al., 2012). Besides, the latency is also a problem for the Kinect system. In order to use it for location-aware sensing, we need to augment the Kinect with additional sensors, e.g. the built-in inertial sensors in a mobile device.

The combination of the Kinect and a mobile device has been studied in the literature and this will be reviewed in section 2.2.2. In this thesis, the fusion of the Kinect sensor and mobile inertial sensors focuses on data-level fusion. The mobile inertial sensor data can compensate for the effects of position uncertainty and latency in the conventional Kinect skeleton tracking.

Inertial sensors are becoming ubiquitous in a smartphone, which has become an essential part of our everyday life. Nowadays, a smartphone is usually equipped with a wide range

of sensors, such as an accelerometer, a gyroscope, a magnetometer, camera and GPS. These sensors measure people's everyday motion, for instance, walking, running, answering the phone etc. Thus, the sensors can be used to monitor the daily activities of a person and profile their preferences and behaviour, making personalized recommendations for services, products, or points of interest possible (Lane et al., 2010). If we want to augment the Kinect system with such a mobile device, we need to find the connection between these sensors.

The Kinect sensor and the inertial sensors have *complementary* properties. The Kinect senses human pose and can be used for human skeleton tracking. However, the inferred joint positions are subject to significant uncertainty (Casiez et al., 2012). Inertial sensors, which have been widely used for sensing human movement (Luinge, 2002), can be used to measure the skeleton joint acceleration. The higher frequency acceleration can augment the noisy, low-sampling-rate positions sensed by the Kinect. Thus, the inertial sensors can be used to compensate for the shortcomings of the Kinect sensor. Meanwhile, the Kinect sensor can provide the absolute position information in 3D space, where the inertial sensors suffer from integration drift problem for position changes estimation. In this thesis, our focus is to augment the Kinect with mobile inertial sensors.

Firstly, we can apply the proposed novel and improved Gaussian Process (GP) prior model for computing the joint log-likelihood of the low-sampling-rate position and the high-sampling-rate acceleration for user matching. Secondly, we can fuse the Kinect position and the acceleration measured by mobile inertial sensors for position prediction with the GP prior model. The sensor fusion helps increase the stability of the skeleton joint position and reduce the lag. Responsiveness is a critical factor for a real-time interaction system (Wachs et al., 2011). The sensor fusion helps improve the position tracking and reduce the overall lag of the system, improving the usability of the system.

Probabilistic Approach

In order to explore the complementary properties of the Kinect sensor and mobile inertial sensors, we need a sensor fusion approach. In multisensor data fusion area, Hall & Llinas (1997) proposed a data fusion process model, which uses a variety of data processing levels to extract data from sources, and provides information for Human-Computer Interaction (HCI). The first level processing combines multisensor data to determine the position, velocity, attributes, and identity of individual objects or entities (Hall & Llinas, 1997). To apply this concept for human motion tracking and analysis in human-computer interaction area, the human body tracking and the identity of the user are two important aspects that we need to deal with using multisensor data fusion approaches. The researchers in robotics and HCI area prefer Bayesian probabilistic approaches, among which the Kalman filters (KF), Hidden Markov Models, Dynamic Bayesian Network and particle filters are popular methods.

In order to fuse the Kinect sensor and the inertial sensors for state estimation, we need dynamical system modelling techniques. Bayesian filtering is a general framework for recursively estimating the state of a dynamic system (Ko & Fox, 2009). The basic idea of Bayesian filtering is that we estimate the state of the system with probabilistic models, including the state transition model and the observation model. For instance, the Kalman filter and its variants (EKF and UKF) have been widely used for filtering and sensor fusion (Welch & Bishop, 1995, 1997).

Although Bayesian parametric filters, e.g. the Kalman filter, are efficient, the data flexibility and the predictive capabilities are limited (Ko et al., 2007). In recent years, Bayesian nonparametric models have become popular. Gaussian Process (GP) priors are examples of nonparametric models and have been applied for classification and regression problems, such as robotics and human motion analysis (Wang et al., 2008; Ko & Fox, 2009).

Considering the complementary properties, the different sampling rates and different noise characteristics of the Kinect sensor and mobile inertial sensors, we present a novel and improved Gaussian Process prior model that provides a principled mechanism for incorporating the low-sampling-rate position measurements and the high-sampling-rate derivatives in multi-rate sensor fusion, which takes account of the uncertainty of each sensor type. We chose a Gaussian Process (GP) prior model-based sensor fusion approach as this model satisfies the requirements for (1) user matching and identification (2) position stabilisation and lag reduction in a location-aware sensing application. The proposed GP prior model has two beneficial aspects that correspond with the two applications. On one hand, the model can be applied for computing the joint log-likelihoods of matching a particular user's skeleton with multiple time-series of acceleration signals sensed by the mobile devices. The highest log-likelihood indicates the best match of a user and a device. On the other hand, we can fuse the low-sampling-rate positions sensed by the Kinect and the higher frequency accelerations measured by the mobile devices with the proposed GP prior model for improving the skeleton joint position estimation. This satisfies our second requirement.

1.3 Thesis Aims and Contributions

This research aims to present a multi-rate sensor fusion system for (1) user matching and identification and (2) position stabilisation and lag reduction in a spatially aware display application. The approach we adopt is to apply a Gaussian Process (GP) prior model-based sensor fusion approach to fusing the Microsoft Kinect sensor and the built-in inertial sensors in a mobile device.

The main contributions of this research include:

1. We describe the use of transformations of Gaussian Process (GP) priors to improve the context sensing capability of a system composed of a Kinect sensor and mobile inertial sensors. We propose a variation of a Gaussian Process prior model (a type of Bayesian nonparametric model) (Rasmussen & Williams, 2005) that provides a principled mechanism for incorporating the low-sampling-rate position measurements and the high-sampling-rate derivatives in multi-rate sensor fusion, which takes account of the uncertainty of each sensor type. This is of great benefit for implementing a multi-rate sensor fusion system for novel interaction techniques.

   This will be presented in Chapter 4 *The Sensor Fusion System.*

2. We propose the use of Gaussian Processes prior model-based sensor fusion approach for user matching and identification. We apply the GP model to identify individual users, by matching the observed Kinect skeletons with the sensed inertial data from their mobile devices using the GP prior model-based sensor fusion algorithm. We apply the proposed GP model for calculating the joint log-likelihood of the low-sampling-rate sensor measurements and the high-sampling-rate derivatives. This is beneficial for associating the motion sensed by the measurement sensor (e.g. a position sensor) with the motion sensed by the derivative sensor (e.g. a velocity sensor or an acceleration sensor).

   This will be introduced in Chapter 5 *Transformations of Gaussian Process Priors for User Matching.*

3. The novel and improved GP prior model-based sensor fusion helps stabilise the skeleton joint position, and reduce the lag of the system, thus improve the usability of an interaction system composed of a position sensing device (Kinect) and the mobile inertial sensors in a spatially aware display application.

   This will be described in Chapter 6 *Experiment – User Performance Improvement in Sensor Fusion System*

4. Coordinate system transformation. We propose a method for converting the coordinates from the body frame to the Kinect frame. Experimental results in section 3.4.2 show that the hand accelerations estimated with the Kinect sensor and the inertial sensors are comparable. In this way, the high-sampling-rate movement acceleration estimated with the mobile inertial sensors can be used to augment the noisy, low-sampling-rate Kinect position measurements.

   This will be introduced in Chapter 3 *Sensor Fusion with Multi-rate Sensors-based Kalman Filter.*

5. Fusing the low-sampling-rate position measurements sensed by the Kinect sensor and the high-sampling-rate accelerations measured by the mobile inertial sensors with a

multi-rate sensors-based Kalman filter. The sensor fusion helps improve the accuracy of the system state estimation, including the position, the velocity and the acceleration.

This will be introduced in Chapter 3 *Sensor Fusion with Multi-rate Sensors-based Kalman Filter.*

1.4 Thesis Outline

The remainder of the thesis is organised as follows:

Chapter 2 *Context-Aware Sensing and Multisensor Data Fusion*

This chapter presents a literature review. We introduce the context-aware sensing systems, the indoor positioning technologies that can be used for human motion tracking. We discuss the Kinect sensor and the inertial sensing of human movement, and describe the multisensor data fusion and the Gaussian Process framework for sensor fusion.

Chapter 3 *Sensor Fusion with Multi-rate Sensors-based Kalman filter*

In this chapter, we present a coordinate system transformation method for converting the acceleration estimated with inertial sensors from the body frame to the Kinect coordinate system, and design a multi-rate sensors-based Kalman filter for fusing the low-sampling-rate positions and the high-sampling-rate accelerations.

Chapter 4 *The Sensor Fusion system*

This chapter presents the novel GP prior model-based sensor fusion system composed of a Kinect sensor and mobile inertial sensors. We give a detailed description of the GP prior model-based sensor fusion approach and apply it for fusing the Kinect sensor and the built-in inertial sensors in a mobile device.

Chapter 5 *Transformations of Gaussian Process Priors for User Matching*

This chapter presents the first application of the proposed sensor fusion system. In this chapter, we apply the novel and improved GP prior model for user matching application. We conducted three experiments and investigated the performance of the proposed GP prior model in these situations: (1) subtle hand movement (2) with a mobile device in the user's trouser pocket (3) walking with a mobile device held in the hand. We compared our work with the state-of-the-art work presented in the literature and demonstrated that our method achieves successful matches in all 3 contexts, including when there are only subtle hand movements, where the direct acceleration comparison method fails to find a match.

Chapter 6 *Experiment – User Performance Improvement in Sensor Fusion System*

This chapter presents a user study on the sensor fusion system in a spatially aware display application, where the user performed the trajectory-based target acquisition tasks. Experimental results show that the improved accuracy of target selection, and reduced delay from the sensor fusion system, compared to the filtered system means that users can acquire the target more rapidly, and with fewer errors. They also reported improved performance in subjective questions.

Chapter 7 *Conclusions* drawn from the thesis, and discussions of the benefits of the proposed sensor fusion system. We propose a coordinate system transformation method to estimate the skeleton joint acceleration in the Kinect frame, and use a multi-rate sensors-based Kalman filter approach to fusing the Kinect and mobile inertial sensors. We design a novel and improved GP prior model-based sensor fusion approach for user matching and identification, and position stabilisation and lag reduction.

Chapter 2

Context-Aware Sensing and Multisensor Data Fusion

In this chapter, we present a brief survey on the context-aware sensing and multisensor data fusion. We highlight the importance of identification of people and their positions in an indoor environment. Following this, we introduce the context-aware systems dealing with location information, i.e. the location-aware sensing applications. We discuss the challenges, including the position uncertainty and the lag problem, and emphasize the importance of accurate position tracking and fast system response. Following this, we present the position sensing technologies. After that, we give an introduction to mobile interaction in space. As the indoor human motion tracking plays a key role in a proxemic interaction system, we discuss the human motion tracking techniques. We focus on the inertial sensing and the Kinect skeleton tracking, the fusion of which will run through the thesis. After this, we give a brief introduction of the multisensor data fusion and its applications. Following this, we discuss the probabilistic approaches for sensor fusion. We introduce the Bayesian filters, including the Kalman filter and its variants. Moreover, the Gaussian Processes (GPs) framework is described. We emphasize the benefits of GPs, including the GP log-likelihood and the GP prediction.

2.1 Context-Aware Sensing

Context-aware sensing plays a key role in Ubiquitous Computing (UbiComp), where information processing has been thoroughly integrated into everyday objects, activities, and computing is everywhere. The applications in UbiComp are based on the context, which can include a person's location, goals, resources, activity and state of people, and nearby people and objects (Salber et al., 1999; Krumm, 2009).

Context is very important in sensing-based interactions and interest in context-aware computing is high (Abowd et al., 2002). Context plays a crucial role in understanding of human behavioural signals, since they are easily misinterpreted if the information about the situation in which the shown behavioural cues have been displayed is not taken into account (Pantic & Rothkrantz, 2003). In (Dey, 2001), context was defined as any information that can be used to characterise the situation related to the interaction between users, applications and the surrounding environments. Dey et al. (2001) introduced four essential categories of context information, including identity, location, status (or activity) and time. Context is often inferred with sensors (Fraden, 2004), which include wearable sensors and environment sensors. Micromachined sensors such as accelerometers and gyroscopes are small enough to be attached to human body, and have thus been widely used for measuring human movement (Luinge, 2002). Context inferencing is the act of making sense of the data from sensors and other sources, to determine or infer the user's situation (Krumm, 2009). For example, to determine who the user is, or what he is doing. Based on this information, the appropriate action could be taken by the system.

The sensor-based and context-aware interaction system could use the information gathered from sensors and adjust to a user's behaviour. In a location-aware sensing application, e.g. a digital book library application (Norrie et al., 2013), the system could detect the user's location in a room and enable the user to browse the virtual information, i.e. the different digital books embedded in the physical space.

In context-aware computing, human-computer interaction is more implicit than ordinary interface use (Dix, 2004). Schmidt (2000) proposed that implicit human-computer interaction is an action, performed by the user that is not primarily aimed to interact with a system but which the system understands and takes as input. Thus, implicit interactions are based not on explicit action by the user, but more commonly on the user's existing patterns of behaviour. For example, the user identification in smart home (Kadouche et al., 2010). Vogel & Balakrishnan (2004) proposed an interaction framework for ambient displays that support the transition from implicit to explicit interaction by identifying individual users through registered marker sets, and argued the need for marker-free tracking system and user identification technique. The concept of implicit and explicit interaction has been regulated by proxemics in proxemic interaction (Ballendat et al., 2010).

In context-aware computing, an important type of interaction system is the proxemic interaction system. As discussed in section 1.1, Greenberg et al. proposed that *proxemic interactions* relate people to devices, devices to devices, and also relate the objects in the room-sized environment to people and devices (Ballendat et al., 2010). Knowledge of the identity of a person, or a device is critical in proxemic-aware applications (Ballendat et al., 2010).

The user identification is beneficial for service personalization, e.g. how the system responds

to that particular user. The context-aware applications are built to facilitate people's usage. In order to make computer technology more usable by people, we need to build a system that can understand who the user is, who interacts with it (Jaimes & Sebe, 2007). In this way, the system can provide personalized services or make personalized recommendations to the user. For example, in a family environment, the system can help family members personalize their own TV programs and multimedia services.

Another essential part of proxemic interaction is indoor position tracking. Ballendat et al. (2010) proposed that the tracking system should return the four dimensions in order to determine the basic proxemic relationships between entities, including position, orientation, movement and identity.

Therefore, identification of people and their positions in a room-sized environment plays a key role in a proxemic interaction system. Identifying the user implicitly and tracking the user for location-aware sensing applications in an indoor environment are the crucial parts of context sensing in context-aware applications.

2.1.1 Location-Aware Sensing

Context-aware systems dealing with location information, i.e. location-aware sensing systems, have widespread applications, e.g. mobile tour guides (Salber et al., 1999), augmented reality (Azuma et al., 2001), mobile spatial interaction (Strachan & Murray-Smith, 2009) and spatially aware display (Fitzmaurice, 1993). Hightower & Borriello (2001) presented a survey of the basic techniques used for location-sensing and described a taxonomy of location systems for ubiquitous computing. The rapidly developing sensing techniques and pervasive computing applications provide people access to information everywhere and anywhere.

Mobile devices equipped with GPS, digital camera and multiple sensors are becoming ubiquitous, enabling researchers in HCI to explore the use of mobile devices to access and augment information related to the user's surroundings. The combination of GPS and mobile devices can be used for outdoor applications, e.g. navigation (Robinson et al., 2012) and bearing-based target selection (Strachan & Murray-Smith, 2009).

In this thesis, our work focuses on indoor position sensing. In particular, we study the human skeleton joints position tracking and the indoor joint location-aware applications. We explore the use of mobile inertial sensors to improve the Kinect skeleton tracking. Now we give a brief introduction to the location-aware sensing applications.

Location-Aware Sensing Applications

Nowadays, augmented reality (AR) is a popular location-aware sensing application, especially the mobile AR. Augmented reality (AR) supplements the real world with computer generated graphics to create a seamless environment for enhancing a user's interaction with the real world (Azuma et al., 1997, 2001). With the development of advanced sensors and powerful computing devices, the mobile phone is becoming a tool for accessing ubiquitous information. For instance, a mobile device can be used as a handheld display for mobile augmented reality system, which exploits the person's surrounding context and provides a powerful user interface to context-aware computing environments (Höllerer & Feiner, 2004).

Mobile spatial interaction is an emerging field in location-aware applications (Fröhlich et al., 2007; Strachan & Murray-Smith, 2009). The three main categories of mobile spatial interaction include orientation and wayfinding, access and creation of spatial data and augmented reality (Froehlich et al., 2008). Strachan et al. (2007) proposed BodySpace, where positions on the body were assigned to specific functions. Virtual Shelves (Li et al., 2009) allowed a user to trigger programmable shortcuts by orienting a spatially-aware mobile device within the circular hemisphere in front of the user.

Spatially aware displays provide access to more information by mapping the physical movement of the device to the movement in virtual space. In this way, the screen of the handheld device is like a window, through which the user can see the virtual information stored in the physical space. Fitzmaurice proposed this idea in (Fitzmaurice, 1993). In such a spatially aware display application, people would browse and interact with electronic information within the context with a small, portable, high-fidelity display and spatially aware palmtop computer, which could act as a window onto the 3D-situated information space. This kind of spatially aware display application allows the user to access, modify and interact with the information in a matter of seconds.

Challenges

A central problem in mobile augmented reality (AR) and other location-aware computing applications is location sensing. For outdoor applications, GPS is a popular location sensing technique. In this thesis, the location sensing refers to the indoor position tracking. In particular, we study the human skeleton joints position tracking. For any location-aware system, position uncertainty and inaccuracy is critical to the effective use and acceptance of the system (Strachan & Murray-Smith, 2009; Azuma et al., 1997). For example, in an augmented reality application, accurately tracking the user's position is crucial for AR registration. Accurate registration and positioning of virtual objects in the real environment requires accurate position tracking (Azuma et al., 1997). However, the static and dynamic

errors exist and seriously influence the user's interaction and experience in an AR system (Azuma et al., 1997).

Besides the position uncertainty, another key problem in location-aware sensing applications is the latency. For instance, the temporal mismatch of real and virtual view in AR will cause problems due to the system delay, which is often the largest source of registration errors in AR systems (Azuma et al., 2001).

Therefore, accurate position tracking and fast system response play key roles in augmented reality (AR) and other location-aware sensing applications. For indoor location-aware sensing applications, we need position sensors and tracking devices. Although advanced position sensing devices are being developed and used for tracking, uncertainty always exists. In order to improve the accuracy of the position tracking and reduce the lag of the system, we need additional sensors to augment the position tracking device.

We need multisensor data fusion techniques to fuse the data from different sources. Different sensors often have different sampling rates and different noise characteristics. A major challenge in determining the location is to make sense of a large amount of sensor data. The sensor fusion techniques provide support for location-aware applications (Hazas et al., 2004). Two important issues in sensor fusion are *uncertainty* and *lag*.

Uncertainty *Uncertainty* is a well-known topic in robotics and human-computer interaction (HCI) area. Sensors have limited perceiving capabilities and are subject to noise, which perturbs sensor measurements. Uncertainty should be handled appropriately for robust interaction in the human-computer interaction area (Strachan & Murray-Smith, 2009; Schssel et al., 2013).

The Microsoft Kinect is a motion sensing input device, which provides 3D human body tracking that enables whole-body input (Shotton et al., 2013). It contains a RGB camera, 3D depth sensors and multi-array microphones. It is low-cost, portable and has enabled new styles of human-computer interaction. The Kinect has attracted much interest since its release. In 2010, Microsoft released the Kinect as a gaming platform. Researchers in HCI started to use it for Natural User Interface (NUI) and have explored the use of the Kinect sensor for novel interaction applications, e.g. dancing evaluation (Alexiadis et al., 2011), sports science and physical rehabilitation (Chang et al., 2011; Velloso et al., 2013), and convenience improvement for everyday life (Panger, 2012; Oh et al., 2012). In addition to putting the Kinect in a fixed location in a room, the researchers also used the Kinect as a wearable device for hand gesture recognition. Bailly et al. (2012) developed the ShoeSense system, a wearable system that used the Kinect as a depth sensor and aimed to recognize relaxed and discreet as well as large and demonstrative hand gestures.

For human motion tracking with the Kinect, the position uncertainty is a common problem

(Casiez et al., 2012). Thus, we need to apply filtering or sensor fusion techniques. However, filtering will induce lag, which reduces the system responsiveness (Casiez et al., 2012), potentially causing lower satisfaction and poor productivity among users (Shneiderman & Plaisant, 2005). For instance, in Virtual Reality (VR), a high latency can induce motion sickness and unpleasant user experience (Preece et al., 1994; Conner & Holden, 1997).

The inertial sensors equipped in a mobile device can be used to compensate for the position uncertainty. In recent years, inertial sensors have become ubiquitous and have been equipped in consumer devices, e.g. smartphones and tablets. The inertial sensors have been widely used in inertial navigation systems. However, drift happens for position estimation with inertial sensors by double-integrating acceleration. The additional position sensing device can be used to compensate for the effect of drift that the inertial sensors suffer from in an inertial navigation system. In this work, we focus on using the built-in inertial sensors in a mobile device to estimate the acceleration, which can augment the noisy, low-sampling-rate position measurements sensed by the Kinect.

Uncertainty in interaction arises for many reasons, including the inherent limitations of a particular model of the world, the noise in sensor measurements and perceptual limitations of the sensors , and the approximate nature of many algorithmic solutions (Thrun et al., 2005). In (Strachan & Murray-Smith, 2009), uncertainty was divided into two main categories including sensor sources and human sources. For handheld display applications, hand tremor will also induce uncertainty.

The uncertainty needs to be handled appropriately in multisensor data fusion. Due to the complexity of human motion and the difficulty of efficiently fusing information from different sensors, human motion analysis based on sensor data is challenging.

Lag The *lag*, which is the delay between input action and output response, can be attributed to properties of input devices, software and output devices (MacKenzie & Ware, 1993). In this thesis, the lag refers to the delay lag. The lag, latency and delay are used interchangeably. Latency is the end-to-end measure of the time elapsed between the moment a physical action is performed by the user, versus the moment the system responds to it with feedback that the user can perceive (Hinckley & Wigdor, 2002). Source of latency may include the hardware sampling rate; the time it takes to report samples to the operating system as well as report events to applications; the processing time required by software; the time to refresh the frame buffer; and the physical screen refresh rate.

The lag reduces the system responsiveness. The system response time is a topic of interest in computer science (Dabrowski & Munson, 2011). The general conclusion is that faster is better. For human motion sensing device, there are delays between the user's input and the output of the computer system, e.g. the Kinect. It is well-known that users dislike delay,

which may cause dissatisfaction and frustration. The user will not get a good user experience when computer systems do not response quickly enough to their input.

Delays in computer systems have a great impact on a user's performance, e.g. the user's accuracy and error rates, and the performance (Dabrowski & Munson, 2011). The delay may lead directly to decrease in satisfaction with computer systems and increase in frustration, annoyance and irritation with the system.

Lag is inevitable and is a problem for all interactive systems. For instance, the system delay is often the largest source of registration errors in augmented reality (AR) systems (Azuma et al., 2001). The lag is negligible in some traditional computing systems, e.g. text entry or cursor movement. With the development of sensor techniques and computing devices, smartphones and tablets are augmented with accelerometers, gyroscopes and other sensors, which allow novel styles of interaction. Although the Microsoft Kinect (version 1) has many advantages, e.g. low-cost and portable, it still has some fundamental limitations with the latency (0.1s) and frame rate (30Hz) (Azimi, 2012; Livingston et al., 2012).

To reduce the position uncertainty and minimize the lag with a filter in the Kinect system is challenging. However, with additional, aiding sensors sampled at higher rates, e.g. inertial sensors, we can improve the usability of the system by increasing the stability of the position and reducing the overall lag of the system.

2.1.2 Positioning Technologies

A key issue in location-aware sensing applications is position tracking. For outdoor applications, Global Positioning System (GPS) is a well-known outdoor positioning technique, but usually not suited for indoor positioning. GPS technology has been widely used for providing location information for the navigation system. However, these applications are limited to outdoor conditions. Reliable positioning of a user in a room plays a key role in indoor location-aware applications. In this thesis, the Microsoft Kinect is used for indoor positioning tracking, which will be introduced in section 2.2.2.

Indoor Positioning

The positioning systems have two main application areas including the outdoor and indoor applications. In this thesis, we focus on indoor position tracking and location-aware sensing applications. The indoor positioning techniques include the InfraRed (IR) radiation, Radio-Frequency IDentification (RFID), ultrasound and ultra-wideband radio, Wireless LAN (WLAN), mobile cellular network and computer vision techniques (Liu et al., 2007; Woodman & Harle, 2008). For indoor mobile interactions, the conventional position tracking

technologies require instrumented environment, e.g. markers and expensive cameras fixed in a room. An alternative option is to use an inertial navigation system.

Inertial Navigation

Navigation is essentially about travel and finding the way from one place to another (Titterton et al., 2004). Inertial navigation has a wide range of applications, including the military applications, e.g. the navigation of aircraft, missiles and ships, and the civilian applications, e.g. the pedestrian tracking (Foxlin, 2005).

Inertial navigation is the process of determining the position and orientation of an object relative to a known starting point using the measurements provided by accelerometers and gyroscopes (Titterton et al., 2004). By combining the two sets of measurements, it is possible to define the translational motion of the vehicle within the inertial reference frame and to calculate its position within it. The inertial sensors are mounted rigidly onto the device in a strapdown system.

The Inertial Measurement Unit (IMU) is typically composed of $3-$axis gyroscopes and $3-$axis accelerometers, sometimes also $3-$axis magnetometers. The 3-axis accelerometer measures the acceleration of the body, and the 3-axis gyroscope measures the changing rate of the body's orientation. The linear velocity, position, and angular position can be obtained by integration. This is the principle behind inertial navigation system (INS), which is widely used in aerospace and naval applications (Corke et al., 2007). By integrating these sensor data, it is possible to track the position, the velocity, the acceleration and the orientation of a device. The availability of accurate knowledge of vehicle position at the start of navigation is a pre-requirement for the inertial navigation systems. An Inertial Navigation System (INS) employs these sensors to calculate the state (position, velocity and orientation) of the moving object without the need for external references.

Orientation estimation plays a key role in inertial navigation. In order to compute the changes of position, velocity and acceleration in a real-world coordinate system, we need orientation information to convert the coordinates from one frame to another. The popular ways of representing orientation include *direct cosine matrix*, *Euler angles (Roll, Pitch and Yaw)* and *quaternion* (Titterton et al., 2004). In order to determine a complete orientation with respect to Earth frame, we need magnetometers. The Attitude and Heading Reference System (AHRS) fuses the accelerometer data, gyroscope data and magnetometer data to provide the object's orientation including the attitude (Roll and Pitch) and azimuth information (Madgwick et al., 2011).

Inertial Sensors Accelerometers and gyroscopes are known as inertial sensors since they exploit the properties of inertia, i.e. resistance to a change in momentum. The accelerometer

senses changes in linear motion and the gyroscope senses the angular motion (Corke et al., 2007). Now we introduce the accelerometer, the gyroscope and the magnetometer.

Accelerometer Accelerometers have a wide range of applications, e.g. in inertial navigation systems, automotive industry and consumer devices (Wilson, 2007). The accelerometers are widely used in automotive air bag systems. The smartphones and tablets equipped with accelerometers can facilitate and enhance a user's interaction through automatically rotating the phone screen to the landscape or portrait mode (Tuck, 2007). Moreover, the built-in hard disks in laptops are usually equipped with accelerometers to detect the external force and protect the disks. These are all example applications in our everyday lives.

The accelerometer measures the total external specific force acting on the sensor. This force includes the movement force plus a force due to the earth's gravitational field. Thus, the accelerometer measures the acceleration due to motion, i.e. the linear acceleration, plus the acceleration due to gravity. In an inertial navigation system, the accelerometer is combined with the gyroscope to provide position changes and orientation information.

Gyroscope The gyroscope is also called an angular rate sensor, which measures angular velocities resolved in the body frame. Gyroscopes have been used in stabilizing handheld cameras and in the Gyromouse product (Wilson, 2007). A gyroscopic mouse uses a gyroscope to sense the movement of the mouse as it moves through the air.

Magnetometer The magnetometer detects the strength of the earth's magnetic field. It is useful for determining the absolute orientation of an object. The fusion of the magnetometer and the accelerometer can provide pose information. In an AHRS system, the magnetometer is used to compute the azimuth (compass heading) information.

Fusion of Inertial Sensors and Aiding Positioning System One disadvantage of inertial navigation systems is drift error. The drift due to the bias and errors is a common problem for inertial sensors. The errors in the accelerometers propagate through the double integration and the errors in the gyroscopes also cause drift.

One way to overcome the shortcoming of inertial navigation, i.e. the drift problem, is to use an aiding position sensing system, which can provide absolute position data. For example, the GPS data can be fused with an INS in outdoor applications. However, the fusion with GPS is unsuitable for indoor applications. The fusion of inertial sensors and visual sensors has been investigated and this will be introduced in section 2.2.2.

2.1.3 Spatial Interaction

Recent progress in sensor technology and computing devices has introduced novel and natural styles of human-computer interaction. The technology embedded in a modern smartphone enables the user to interact with the surroundings and acquire the context information. Moreover, the Microsoft Kinect, which is a motion sensing input device that can be used for skeleton tracking, has received interest in HCI. It can be used as a hand tracking system, which can be combined with a mobile device for augmented reality (AR) and other location-aware sensing applications. Hand tracking systems have been widely used in HCI, e.g. virtual reality and athletic performance measurement (Rehg & Kanade, 1994).

Situating Interaction in Space

The researchers in HCI have much interest in situating interactions in space in order to overcome the limitations of screen size display. The interaction space of a mobile device is not limited to the touchscreen. It can be expanded beyond the physical boundary of the device to the 3D space around the device through aiding sensors.

One type of the expanded interactions is the around-device interaction (Kratz & Rohs, 2009a; Kratz et al., 2012a). The mobile devices with proximity sensors or augmented with a depth sensor enables them to sense the proximity space. Kratz & Rohs (2009a) presented an around-device interaction interface that allowed mobile devices to track coarse hand gestures performed above the device's screen by using infrared proximity sensors to track the hand. Kratz et al. (2012a) proposed PalmSpace, the 3D space by the reach of the user's arm and around the device that allowed manipulating 3D virtual objects via hand gestures. This style of mobile interaction increased the number of degrees of freedom and alleviated the limitations of touch interaction with mobile devices through mid-air gestures in proximity of the device. The interaction space was further expanded in later work. Bailly et al. (2012) proposed ShoeSense, a wearable system that used a Kinect as a shoe-mounted depth sensor pointing upward at the wearer to sense gesture input.

Besides the around-device interaction, the interactions can be situated on the body or around the body. We discussed BodySpace (Strachan et al., 2007) and Virtual Shelves (Li et al., 2009) in section 2.1.1. A body-centric design space that reflects how different body parts enhance or restrict movement within particular interaction techniques was proposed in (Wagner et al., 2013). Kratz et al. (2012b) proposed Attjector, an attention following wearable microprojector, which can be put on the user's shoulder. It is a Kinect-based prototype of the wearable and steerable projector system composed of a Kinect sensor and inertial sensors. The Kinect sensor is used to track the hand position. Meanwhile, the mobile inertial sensors, including an accelerometer and a gyroscope, are fused to maintain level orientation. The

combination of these sensors provides a stabilized mobile projector that allows the projected image to follow the user's locus of attention. This system can be used for peephole pointing applications in a Kinect-augmented environment.

The interaction space can be further expanded to include 3D space beyond the reach of the user's arm. Exploring the use of a handheld device to provide enhanced interaction and information in space has been thoroughly researched in the literature, such as spatially aware display (Fitzmaurice, 1993) and mobile augmented reality (Höllerer & Feiner, 2004). As a handheld device has a limited display size, it is beneficial to improve a user's information navigation with a handheld device.

Spatially aware displays allow the user to access the virtual information embedded in a physical environment through a window, such as a handheld display. Spatially aware handheld devices can serve as bridges between the real and virtual information space (Fröhlich et al., 2007). For outdoor augmented reality applications, a spatially aware display application can serve as a window to the virtual information, augmenting the user's interaction with the real world, e.g. a place of interest (Froehlich et al., 2008). A mobile context-aware tour guide for indoor and outdoor applications was proposed in (Abowd et al., 1997). Peephole displays (Yee, 2003) show a movable window on the large 2D virtual space and augment the physical space around a user with digital information. Dynamic and static peephole navigation on handheld displays were compared in (Mehra et al., 2006). Olwal & Feiner (2009) proposed a method for using a tracked mobile device for direct interaction on large digital displays. Magic lens, which acts as a see-through tool, is a type of mobile augmented reality applications, which improve a user's information navigation (Bier et al., 1993; Rohs & Oulasvirta, 2008).

Peephole interaction allows users to treat their handheld devices as a window (peephole) into a larger information space. In order to display a larger virtual information space on a small screen interface, Rohs & Essl (2006) investigated and compared information navigation techniques, including pan, halo, zoom, and halo & zoom for small-screen interfaces in spatially aware handheld display applications. In recent years, peephole pointing has been studied in the literature (Cao & Balakrishnan, 2006; Cao et al., 2008; Kaufmann & Ahlström, 2012). Cao & Balakrishnan (2006) explored the dynamically defined information spaces using a handheld projector and a pen. Kaufmann & Ahlström (2012) presented a study of target acquisition with a handheld projector in a peephole pointing application, and studied spatial memory and map navigation performance on projector phones with peephole interaction.

Mobile augmented reality becomes increasingly feasible and popular nowadays because of the mobile devices, which grow in power, capabilities and features (de Sá & Churchill, 2012). Mobile augmented reality integrates virtual information into a person's surrounding environment without constraining a person's whereabouts to a specially equipped area (Höllerer &

Feiner, 2004). Mobile handheld devices are popular displays that present the information in physical space to the user. Nowadays, the smartphones equipped with multiple sensors (e.g. camera and inertial sensors) can be combined with location positioning service, enabling the user to gain easy access to the information about their surroundings.

The Kinect-based spatial interaction has received some recent interest in HCI. The Kinect has an interaction space, which is the area that is located in the Kinect field of view. The Kinect-based spatially aware display application explores the use of a handheld mobile device for situated interaction in this space.

2.2 Human Motion Capture and Analysis

Automatic motion capture and analysis is an active research area and has a variety of applications. Moeslund et al. (2006) roughly grouped these applications to three categories, including surveillance, control and analysis. In control applications, the human motion estimation is to enable the user to control something, e.g. mobile augmented reality in human-computer interaction (HCI).

Interest in human motion goes back very far in human history, and human motion capture and analysis have been developing and have widespread applications. The inherent curiosity, needs and methods motivate humans to explore and understand (Klette & Tee, 2008). Human motion capture goes back to at least nineteen century (Moeslund & Granum, 2001). Human motion analysis plays an important role in many fields, such as athletic performance analysis, video surveillance, video conferencing and human-computer interaction (Aggarwal & Cai, 1997).

The human motion analysis has attracted much interest and the standard functional taxonomy for human motion analysis has been established. In (Moeslund & Granum, 2001) and (Moeslund et al., 2006), the human motion analysis includes four parts, including initialization, tracking, pose estimation and recognition. Human motion recognition is a high level of analysis. It covers the recognition of individuals' identities, actions, activities and behaviors performed by one or more people (Moeslund et al., 2006). Thus, *user identification* is an important issue in human motion analysis area.

Human motion analysis is still challenging due to the high dimensionality of human pose data and the complexity of the motion. Automatic tracking and recognition of human behavior is a common requirement of potential applications of human motion analysis (Moeslund et al., 2006).

2.2.1 Human Motion

Human motion consists of a variety of motion levels. Bobick (1997) used a different taxonomy of human motion: *movement, activity, action*. *Movements* are atomic primitives, requiring no contextual or sequence knowledge to be recognized. *Activity* refers to a sequence of movements or states, where the only real knowledge is the statistics of the sequence. *Actions* are larger scale events which typically include interaction with the environment and causal relations.

Human motion (e.g. body movement, gesture and gaze) plays an important role in HCI. With the development of advanced sensors and computing devices, the human motion capture becomes feasible in people's everyday lives. This could benefit the researchers to develop novel interaction techniques. In contrast to the traditional input devices, such as the keyboard and the mouse, the novel sensing devices allow the user to use the hand or the whole body as the input, e.g. the Kinect sensor. The availability of new inputs and outputs devices provide us more information about how the user moves. These devices open a rich design space for HCI researchers to develop novel interaction techniques and applications.

The combination of position tracking and human motion brings us a human motion tracking system. In the above section, we discussed the position tracking techniques. Now we introduce the human motion capture systems.

2.2.2 Human Motion Capture Systems

In order to analyse the human motion, we need the equipment that can be used to capture human motion. Human motion tracking systems play an important role in sport sciences (Velloso et al., 2013), film industry and consumer-level motion tracking applications, e.g. Nike+Kinect Training.

Human motion tracking systems can be divided into two categories: (1) optical motion capture systems, including marker-based optical motion capture systems and markerless motion capture systems; (2) non-optical motion capture systems.

Optical Motion Capture Systems

Optical motion capture system uses computer vision techniques for human motion tracking. (1) The marker-based system uses the markers attached on the body. For example, the image-based systems use multiple cameras to track the markers on the subject's body segments. The infrared (IR) LED is used in reflective systems. The conventional marker-based optical motion capture systems are expensive and obstructive (Poppe, 2007). (2) The markerless

systems track human motion using advanced computer vision algorithms without the aid of markers.

There are some challenges with motion tracking with optical systems. Computer vision-based tracking systems often suffer from sensitivity to illumination and occlusion problems. The lighting conditions often influence the tracking results and the tracking reliability.

Commercial tracking technologies have been used for human motion tracking and applications in the literature. A Vicon motion tracking system was applied for human body location and orientation tracking in (Vogel & Balakrishnan, 2004). Ballendat et al. (2010) used a Vicon infrared camera tracking system to sense a room-sized environment, including people, objects and digital devices moving around an interactive wall display. However, such a camera tracking system is expensive, and it requires the user to attach markers on the body for tracking. Ballendat et al. (2010) proposed that proxemic interaction requires cheaper tracking technology for sensing proximity and orientation. Vogel & Balakrishnan (2004) discussed the two challenges involved in proxemic interaction design, that is, the marker-free tracking and user identification techniques.

Non-optical Motion Capture Systems

An alternative to vision-based tracking is sensor-based wearable computing technology. The use of sensors enables us to capture human behavioral signals including facial expressions, body gestures, non-linguistic vocalizations, and vocal intonations (Pantic et al., 2007). With the development of computing devices, such as a mobile device equipped with inertial sensors, a revolution has been happening in sensor and measurement technologies, enabling measurement devices to be deployed comfortably without encumbering daily activity (Picard, 2010).

The recent progress in sensor technology and computing devices could benefit human motion analysis and its applications in HCI by providing intuitive human motion data. The rapid development of micro-machined electromechanical system (MEMS) technology has led to smaller and cheaper inertial sensors. A lot of wearable sensors and devices are available on the market. For example, the electronic badges, mobile phones, wrist-mounted devices, head-mounted devices and electronic textiles (Olguín-Olguín & Pentland, 2010). These wearable devices could function as self-contained monitoring devices. For instance, the built-in inertial sensors in a mobile device can be used in an inertial navigation system to detect the changes of position and orientation. With a known starting point, the sensors can detect the location and orientation of a body part. Moreover, these sensors may also communicate with each other or radio base stations in a wireless sensor network. The wearable sensing devices should have a small form factor, be comfortable to wear over long periods

of time, and have a long battery life. The motion detection sensors may include accelerometers, gyroscopes, magnetometers and inclinometers. With these sensors, we could get a lot of measurements, such as the body movement detection, the body position and orientation, body postures (e.g. sitting, standing and lying down) and physical activities (e.g. walking and running) (Olguín-Olguín & Pentland, 2010). The wrist-mounted inertial sensors can be used for forearm and hand gesture recognition (Morganti et al., 2012). The recent development in wearable computing has been enabling people's digital lives. Park et al. (2014) gave an introduction to the fundamentals of wearables and the recent advancements, and discussed the future of wearables.

In addition to the wearable sensors, the environment sensors (e.g. temperature, light, sound, movement and activity), which capture the current conditions in an office environment, can be placed in fixed locations inside a building in order to detect and track the location of interaction events and subjects (Olguín-Olguín & Pentland, 2010).

Inertial Sensing Micro-machined inertial sensors have been widely used for human motion analysis. The most common approach is to attach multiple inertial sensors on the subject's body segments. The complementary inertial sensors are fused to estimate the orientation and position of each body segment, and provide six Degree-Of-Freedom (DOF) tracking of the human body.

The use of inertial sensors for human motion tracking is a common practice and has been studied in the literature (Luinge, 2002; Zhu & Zhou, 2004; Roetenberg, 2006; Roetenberg et al., 2009). Zhu & Zhou (2004) used tri-axis microelectromechanical inertial sensors and presented a Kalman-based fusion method to track the orientations and positions of human body segments. Roetenberg (2006) combined inertial sensors with an optical tracking system for improving motion tracking performance, and inertial sensors could also be combined with magnetic sensors for position and orientation tracking. However, due to the drift, the inertial sensing systems for human body tracking cannot provide accurate and complete positions of body segments without the extra aiding sensors. The additional position sensing device is needed for a reliable full body tracking.

Inertial sensing has many advantages for human motion capture. The inertial sensors are small enough to be attached on the human body. Moreover, the built-in inertial sensors in consumer devices are becoming ubiquitous, making human motion sensing implicitly available in people's everyday lives. The inertial sensors are sampled at a higher rate in comparison with the Kinect (sampling rate 30Hz). The inertial sensor data are accurate for analyzing the rapid changing of hand motion, e.g. the hand pose estimation, which cannot be sensed by the Kinect. Also, inertial sensing has the potential to be sampled more frequently, leading to much lower lags in comparison with the latency (0.1s) of a Kinect. Moreover,

the acceleration can be estimated through inertial sensor fusion, which will be described in section 3.4.

Nowadays, the mobile device equipped with advanced sensors can adjust the sampling rates of the inertial sensors to maximize the battery life. For example, the sampling rate of inertial sensors will be high when the phone is moving fast while the sampling rate will be low when the phone is stationary. The sampling rate of mobile inertial sensors influences the battery life of the phone. The automatic adjustment of sampling rates increases the usage of the phone by maximizing the battery life.

Microsoft Kinect Skeleton Tracking Figure 2.1 illustrates that the Kinect skeleton tracking provides human skeleton joints positions in 3D space.

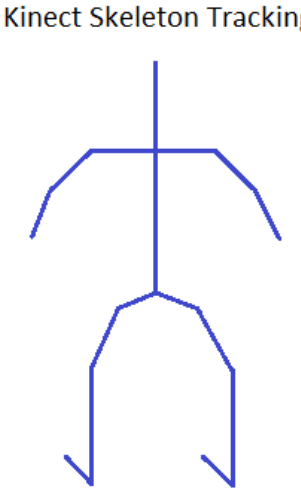


Figure 2.1: The Microsoft Kinect sensor can be applied for human skeleton tracking, which provides a stick figure in 3D space.

The Kinect skeleton tracking provides a way of representing the human pose in 3D space. The stick figure is shown in Figure 2.1. In a skeleton tracking, a human body is represented by a number of joints representing body parts, such as head, shoulders and hands. The skeleton tracking gives the 3D coordinates of each joint. By connecting these joints in 3D space, we get a "stick" figure. The movement of the human body is represented with the moving joints connected with lines. This is one of the conventional methods used to analyse the human body. Other methods include 2D contours, or volumetric models (Aggarwal & Cai, 1997). The human body can be represented at various levels of detail, involving bounding boxes, stick figures, 2D contours, or 3D volumes, based on the complexity of model required in an application.

Fusing Position Sensing and Inertial Sensors Previous work on fusing an inertial navigation system with other systems, such as position sensing systems, is well established

in traditional navigation applications (Brown et al., 1992).

Sensor fusion, combining position sensor and inertial sensors has been applied in inertial navigation system (INS) and the motion control of robots (Jeon et al., 2009). For inertial navigation applications, an INS-GPS integration system combines INS measurements with GPS, providing greater precision than any single system alone (Titterton et al., 2004). For motion control of robots, the combination of vision sensors and inertial sensors has been investigated in the literature (Corke et al., 2007; Hol et al., 2007; Armesto et al., 2007; Gemeiner et al., 2007; Grewal et al., 2007). Corke et al. (2007) gave an introduction to inertial and visual sensing, where they showed the complementary properties of inertial and vision sensors and integrated information to provide a robust and non-ambiguous representation of robotic motion. Hol et al. (2007) proposed a method for estimating the position and orientation (pose) of a camera by fusing measurements from inertial and vision sensors. Integration of visual sensing and inertial sensors opens a rich design space for robotics and HCI. The fusion of the Kinect sensor and inertial sensors enables the HCI researchers to explore the use of mobile devices for enhanced spatial interaction in a Kinect-augmented environment.

Combination of Microsoft Kinect and Mobile Device The advanced sensing techniques bring novel and natural styles of human-computer interaction. The Kinect has been used as a popular platform for developing NUI. Besides, the modern smartphones are being equipped with advanced sensors, which can improve the context sensing capabilities of a system, e.g. the accelerometer-based user identification, and provide rich feedback information through screen display, e.g. the visual feedback in peephole interaction enables the user to control the device.

In the literature, the combination of the Kinect and other mobile devices has attracted some recent interest. In (Vera et al., 2011), the Kinect was combined with a Gyroscope and WiiMote for an augmented mirror application. However, each component was used separately without a sensor fusion algorithm. Rofouei et al. (2012) combined the Kinect and mobile devices for user matching application. They proposed the ShakeID method, which is a technique for associating multi-touch interactions with individual users and their mobile devices. Kratz et al. (2012b) proposed a Kinect-based prototype of a wearable and steerable projector system composed of a Kinect sensor and inertial sensors. The Kinect was used to track the user's hand position and the inertial sensor data were fused to maintain level orientation. Bailly et al. (2012) proposed the ShoeSense system that can enhance the capabilities of the mobile device by serving as an input device and providing more degrees of freedom. Norrie & Murray-Smith (2011) proposed that the Kinect can be combined with a modern mobile phone to rapidly create digitally augmented environments. In (Norrie et al., 2013), situated interactions with digital book collections on a smartphone were studied. The prototype uses the Kinect depth sensor to detect a user's position and the mobile application allows users to

browse and organise the digital books.

Although the Kinect sensor and the mobile device can be combined for enhanced interaction, the Microsoft Kinect, a commercial motion tracking system, is essentially a Black Box to developers (Ashby, 1957). The user moves in the field of the Kinect's view and the system gives the user's skeleton tracking. Not all of the system is accessible to direct observation. In order to provide access to more information about the user's motion, we need additional, aiding devices and data fusion techniques.

We have discussed the complementary properties of the Kinect sensor and mobile inertial sensors in section 1.2.2. In order to adapt this Black Box for use in other services, we explore the use of a mobile device to augment the Kinect sensor. We can develop mobile interaction applications by fusing multiple sensor data.

For an indoor location-aware sensing application, the system can make use of the sensor data and provide useful information on the device, such as a smartphone or a tablet, allowing the user to get access to more information in the Kinect space. The combination of the Kinect sensor and mobile inertial sensors allows the user to get feedback on their movement with a mobile device in a Kinect-augmented system. The visual feedback on a phone screen enables the user to see information on the movement and to control, bringing an enhanced interaction experience. By providing real-time feedback to the user, the system facilitates the user's action when the user wants to explore the digital information space embedded in the physical environment. This kind of real-time feedback enables the user to better control the system, to improve the user's experience in the mobile spatial interaction. The work in this thesis aims to enhance the user's interaction through improved sensing accuracy and reduced delay with a multisensor data fusion approach.

In general, the commercial sensing system in HCI is a Black Box to the developers and the users. In this thesis, we present a view that this kind of Black Box system can be adapted for interaction development.

2.2.3 Human Motion Analysis

Human motion analysis requires sensor data processing techniques. The multiple sensors in a human motion tracking system bring a large amount of data. Although more and more advanced sensors are being developed, the data acquired from the sensors may be influenced by several factors, such as environment noise, sensor noise and the limitation of the sensor's transducer. Hence, we need data processing techniques.

The sequential sensor data processing techniques can be used to reduce noise and infer context beyond what the sensor actually measures (Krumm, 2009). Mean and median filters,

the Kalman filter, the particle filter, and the Hidden Markov Model are commonly used techniques for processing sequential sensor data. Among these methods, the mean and median filters do not have a dynamic model of the measured process. For the Kinect skeleton joint position tracking, a dynamic model can be applied for state estimation. The state variables of interest may include those that are not sensed by the sensor. For instance, a Kalman filter can be used for filtering and prediction. In the meantime, it can be used to estimate the position, the velocity and the acceleration of the target based only on the position measurements.

Multiple sensors combined with a mathematical model can be viewed as a "virtual sensor" that gives information about variables of interest, which cannot be observed or measured directly. The process of reconciling signals from many sensors with mathematical models is also called sensor fusion (Aström & Murray, 2010).

The sensor data from multiple sources need to be fused using multisensor data fusion method. The multiple sensors, e.g. the Kinect sensor and the inertial sensors, make human motion tracking feasible in people's everyday lives and provide a large amount of data. The fusion of the Kinect and inertial sensors is a multisensor data fusion problem. Sensor fusion combines different sensor sources and provides significant advantages over a single sensor source (Hall & Llinas, 1997). Following this, we introduce the multisensor data fusion techniques.

2.3 Multisensor Data Fusion

2.3.1 Introduction

Multisensor data fusion exists in many areas, e.g. the human and the robot. Humans and animals have evolved the capability to use multiple senses to improve their ability to survive (Hall & Llinas, 1997). With the development of advanced sensors and processing techniques, the researchers in HCI are developing multisensor data fusion approaches for novel styles of human-computer interaction.

Multisensor data fusion combines data from multiple sensors, and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone (Hall & Llinas, 1997). It requires interdisciplinary knowledge, and techniques to fuse data are drawn from digital signal processing, statistical estimation and probability, control theory and artificial intelligence (Hall & Llinas, 1997; Luo et al., 2011). It has widespread applications including military applications, e.g. multitarget tracking (Smith & Singh, 2006), and civilian applications, e.g. robotics (Thrun, 2002; Siciliano & Khatib, 2008).

The role of sensor fusion is to minimize the user's uncertainty of information (Llinas et al., 2009). The uncertainty problem in sensor fusion needs to be handled appropriately. Strachan

& Murray-Smith (2009) proposed that the uncertainty could reduce the usability of a system if it is not handled properly, and it could also benefit the user's interaction if the uncertainty is taken into account. The position uncertainty in a tracking system composed of multiple sensors needs to be handled with sensor fusion techniques.

Fusion can be adopted in several ways. The sensor data can be combined at the data level, the feature level and the decision level (Hall & Llinas, 1997). The data level fusion means that the sensor data can be directly integrated. The feature level fusion extracts and uses representative features from sensor data for sensor fusion. The decision level fusion fuses the decision information that each sensor gives, i.e. combines the decision each sensor makes.

Multisensor data could be processed separately and only combined at the end. A better way is to use different sensors in a complementary way. In Chapter 1, we discussed the complementary properties of the Kinect sensor and inertial sensors in section 1.2.2. The sensor fusion process could be considered as an inference problem, and could be dealt with the probabilistic approaches. Now we give a review of the popular probabilistic approaches in sensor fusion area.

2.3.2 Probabilistic Approaches

The probabilistic approaches, which represent uncertainty using probability theory, have been widely used to process sensor data. Probabilistic data fusion methods, e.g. the Kalman filter and its variants, the Monte Carlo and the Sequential Monte Carlo, are widely used in robotics. Alternatives to the probabilistic methods include interval calculus, fuzzy logic and evidential reasoning (Dempster-Shafer theory) (Siciliano & Khatib, 2008). Although many sensor fusion algorithms exist in the literature, there is no standard and well-established evaluation framework to assess the performance of data fusion algorithms (Khaleghi et al., 2011).

Now we give a detailed description on the widely used sensor fusion techniques, i.e. the Bayesian filters, including the Kalman filter and its variants. Following this, we discuss the Gaussian Processes framework and emphasize the sensor fusion application.

2.3.3 Bayesian Filters and Sensor Fusion

The Kalman Filter

In a dynamic system, the state of the system cannot be observed directly. We need to infer the system state based on the measurements sensed by the sensors. The problem is that there is always uncertainty, which may be due to noise, biases, and device inaccuracies. Different

sensors may have different sampling rates and different noise characteristics. A Kalman filter is a technique that has been widely used for fusing all available measurement data, with prior information about the system and measuring devices, to produce a statistically optimal estimate of the underlying system state.

The classical Kalman filter was first introduced by Kalman in his famous paper on the discrete-data linear filtering problem (Kalman et al., 1960). Since then, the Kalman filter has played an important role in numerous applications, particularly in the area of autonomous or assisted navigation.

The Kalman filter can be seen as a special case of a Gaussian Process (GP) (Roweis & Ghahramani, 1999; Leith et al., 2004; Bishop, 2006). However, they differ in the way the models need to be thought about (i.e. physical state based versus covariance function) that describing the underlying process (Reece & Roberts, 2010). The Gaussian Process provides a useful unifying framework which encompasses a variety of popular models (Rasmussen & Williams, 2005).

The Kalman filter is a technique for filtering and prediction in linear Gaussian systems (Thrun et al., 2005). It consists of a process model and a measurement model. The process model, as defined in equation (2.1), describes the transition of the process state, assuming that the next state is a linear function of the previous state. The measurement model, which is also linear and defined in equation (2.2), assumes that observations are linear functions of the state.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.1}$$

where x_k is the system state at time instant k. A is the state transition matrix. Matrix B relates the optional control input u_k to the state and w_k represents the process noise. $p(w) \sim \mathcal{N}(0, Q)$, where Q represents the process noise covariance matrix.

The measurement model is

$$z_k = Hx_k + v_k, \tag{2.2}$$

where z_k is the measurement at time instant k. H relates the state to the measurement. v_k represents the measurement noise. $p(v) \sim \mathcal{N}(0, R)$, where R represents the measurement noise covariance matrix. w_k and v_k are independent Gaussian white noise.

Thus, the Kalman filter estimates the state of a discrete-time system based on the time update equations and the measurement update equations (Welch & Bishop, 1995). Besides the mean estimation of system state, Kalman filter also gives the uncertainty of the estimation, which can be found in the posteriori estimate error covariance P_k, as defined in equation (2.7).

The discrete Kalman filter time update equations include:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.3}$$

$$P_k{}^- = AP_{k-1}A^T + Q \tag{2.4}$$

and the measurement update equations include:

$$K_k = P_k{}^- H^T (HP_k{}^- H^T + R)^{-1} \tag{2.5}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.6}$$

$$P_k = (I - K_k H)P_k{}^-, \tag{2.7}$$

where K_k is the gain matrix. $\hat{x}_k^-$ is the priori state estimate at time step k and $\hat{x}_k$ is the posteriori state estimate. $P_k{}^-$ is the priori estimate error covariance and P_k is the posteriori estimate error covariance.

The Extended Kalman Filter and Unscented Kalman Filter

The classic Kalman filter is a Bayesian technique for linear Gaussian systems. Unfortunately, the state transitions and the measurement model are rarely linear in practice. In order to solve the state estimation problem in nonlinear Gaussian systems, we need to apply variants of the Kalman filter. The Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF) have been widely used to solve the state estimation problem in nonlinear Gaussian systems (Thrun et al., 2005).

The Extended Kalman filter (EKF) is a nonlinear version of the Kalman filter that linearizes about the mean and covariance (Welch & Bishop, 1995). It is probably the most widely used estimation algorithm for nonlinear systems. However, it is difficult to implement and difficult to tune (Julier & Uhlmann, 2004). An alternative filter is the Unscented Kalman filter (UKF) (Julier et al., 1995; Julier & Uhlmann, 2004), which represents a derivative-free alternative to the EKF and provides superior performance at an equivalent computational complexity (Wan & Van Der Merwe, 2001).

The key idea underlying the EKF and the UKF is linearization (Thrun et al., 2005). The difference between EKF and UKF is the linearization method. EKF utilizes the Taylor series expansion method for linearizing the nonlinear functions while UKF applies the unscented transform. The UKF performs a stochastic linearization through the use of a weighted statistical linear regression process.

2.4 Gaussian Processes and Sensor Fusion

The Kalman filter and other variations are all Bayesian parametric filters. However, in many practical applications, parametric models may be efficient, but they have limited data flexi-

bility and predictive capabilities (Ko et al., 2007). In recent years, Bayesian nonparametric models have become popular. Gaussian Process (GP) prior is such a nonparametric model and has been applied for regression and classification problems (Rasmussen & Williams, 2005).

GPs are flexible nonparametric models that are capable of modelling complex nonlinear systems. The use of Gaussian Process models for dynamic system modelling has been investigated in the literature (Williams & Rasmussen, 1996; Kocijan et al., 2005). Kocijan et al. (2005) described the identification of nonlinear dynamic systems with a Gaussian Process (GP) prior model. The idea of Gaussian Process modelling is to place a prior directly on the space of functions. A Gaussian Process is defined as a set of random variables, any finite number of which have a joint Gaussian distribution. A GP can be used as a prior probability distribution over functions in the function space. When we place a prior on the function space, this prior is taken to represent our prior beliefs. That is, we incorporate prior knowledge on the latent function. The combination of the prior and the observation data leads to the posterior distribution over functions. This is the Gaussian Process prior method (MacKay, 1998; Rasmussen & Williams, 2005). It is an example of a probabilistic nonparametric model that provides not only the mean prediction, but also the uncertainty information, which are difficult to evaluate appropriately in nonlinear parametric models. The Gaussian Process prior method for curve fitting was first introduced in (O'Hagan & Kingman, 1978). The prediction and regression using GP priors was studied in (Williams, 1998).

Gaussian Processes (GPs) are useful for human motion data analysis. Lawrence (2003) proposed an algorithm called Gaussian Process Latent Variable Model (GP-LVM) for dimensionality reduction. Lawrence & Moore (2007) extended the GP-LVM through hierarchies and applied the hierarchical GP-LVM to the visualisation of human motion data. Grochow et al. (2004) proposed an inverse kinematics system based on a Scaled GP-LVM of human poses. The GP-LVM was applied for human pose estimation in (Ek et al., 2008). Gaussian Process Dynamical Model (GPDM) (Wang et al., 2008; Urtasun et al., 2006) proved useful for nonlinear time series analysis, with applications to learning models of human pose and motion from high-dimensional motion capture data.

2.4.1 Gaussian Processes

Gaussian Processes (GPs) are useful machine learning techniques that have been applied for regression and classification (Williams & Rasmussen, 1996; Rasmussen & Williams, 2005), and dimensionality reduction (Lawrence, 2005). Over the last decade, Gaussian Processes (GPs) have been widely used as non-parametric regression models in a wide range of applications, e.g. dynamic system identification (Kocijan et al., 2005), model predictive control

(Kocijan et al., 2004), sensor fusion (Murray-Smith & Pearlmutter, 2005) and human motion analysis (Wang et al., 2008). An overview of recent advances in GPs for nonlinear signal processing is presented in (Pérez-Cruz et al., 2013). In our work, we focus on the applications of Gaussian Process Regression (GPR).

Gaussian Processes for Regression

In a dynamic system composed of multiple sensors, we want to estimate the system state from the noisy sensor measurements, and also make predictions in order to predict the future states of the system. Instead of using a specific parametric model for regression, we can apply a nonparametric method. This is a Gaussian Process regression problem. Consider the simplest form of a GPR model,

$$y = f(x) + \varepsilon, \tag{2.8}$$

where ε is a zero-mean Gaussian noise, i.e. $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. x is the input, y is the output and $f(x)$ is the latent function that follows a GP. It can be drawn from a Gaussian Process prior, as defined in equation (2.9). The GP predicts the mean and the variance at any data-point in its input space, conditioned on the training data. It can be completely characterized by its mean function and covariance function.

The Covariance Function

The covariance function plays a key role in GP modelling. It represents the covariance between each pair of data-points in the input space. In order to model a continuous dynamic system, we assume that the covariance function is *stationary*. A stationary covariance function is a function of $x - x'$. A covariance function is *isotropic* if it is a function of $|x - x'|$.

Assuming a relationship of the form $y = f(x)$ between input x and output y,

$$Y \sim \mathcal{N}(0, \Sigma), \tag{2.9}$$

where Σ is the $N \times N$ covariance matrix, the elements of which are functions of inputs X. Y denote the outputs. $\Sigma_{pq} = Cov(y_p, y_q) = C(x_p, x_q)$ gives the covariance between output points corresponding to input points x_p and x_q. A common choice is

$$C(x_p, x_q) = v_0 \exp\left(-\frac{1}{2}\sum_{d=1}^{D} w_d\big(x_p^d - x_q^d\big)^2\right) + v_1, \tag{2.10}$$

where $\Theta = [w_1, \cdots, w_D, v_0, v_1]^T$ are the hyperparameters of the covariance function and D represents the input dimension.

The parameters of the covariance function are optimised using standard optimisation algorithms, to maximize the likelihood of the model.

The Gaussian Process Log-Likelihood

In the Bayesian analysis of regression models, we have

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}. \tag{2.11}$$

The *likelihood* is the probability density of the observations given the parameters. The marginal likelihood is the integral of the likelihood times the prior. Under the Gaussian Process model,

$$p(Y|X) = \int p(Y|\mathrm{f}, X)p(\mathrm{f}|X)\,\mathrm{df}, \tag{2.12}$$

where Y are the noisy outputs. A noisy version of the relationship between the input x and the output y is $y = f(x) + \varepsilon$, where the additive Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$.

The log marginal likelihood

$$\log L = -\frac{1}{2}\log\left|K + \sigma_n^2 I\right| - \frac{1}{2}Y^T\left(K + \sigma_n^2 I\right)^{-1}Y - \frac{1}{2}N\log 2\pi, \tag{2.13}$$

where $K = C(X, X)$, Y represent the observations and $Y \sim \mathcal{N}(0, K + \sigma_n^2 I)$.

The Gaussian Process Prediction

One of the main objectives in time series data analysis is prediction. A Gaussian Process is an example of the use of a flexible, probabilistic, nonparametric model which provides us not only the mean prediction, but also the uncertainty. Now we introduce the Gaussian Process prediction. Prediction with Gaussian Processes has many applications. Dynamic system identification and model predictive control with Gaussian Process models have received much interest (Murray-Smith et al., 1999; Girard et al., 2003a,b). GP prior models provide a probabilistic and non-parametric approach to modelling complex non-linear dynamical systems (Kocijan et al., 2005). In comparison with the conventional parametric models described in the literature, GP prior method has the advantage of being able to provide not only the mean estimation, but also the uncertainty information, which is in general difficult to obtain in a parametric model.

Given a set of N training data-points $\{x_i, y_i, i = 1, \cdots, N\}$, where $X = [x_1, \cdots, x_N]^T$ is a $D-$dimensional vector of inputs. In this work, the time instants are used as the training inputs, thus $D = 1$. $Y = [y_1, \cdots, y_N]^T$ is a vector of output data.

The covariance function is of the form

$$\mathrm{cov}(f(x_i), f(x_j)) = v_0 \exp\left(-\sum_k \omega_k (x_{i,k} - x_{j,k})^2\right) + \sigma_n^2 \delta_{ij}, \tag{2.14}$$

where $\{v_0, \omega_k, \sigma_n^2\}$ are the hyperparameters. v_0 represents the signal variance. As $D = 1$, $k = 1$ here. ω_1 is related with the length scale and σ_n^2 represents the noise variance.

Based on the training inputs X, the covariance matrix C can be determined according to equation (2.14). Given a new input vector x^*, we can find the predictive distribution of the corresponding output y^* according to equations (2.15) and 2.16.

$$\mu(x^*) = C(x^*, X)\left[C(X, X) + \sigma_n^2 I\right]^{-1} Y \tag{2.15}$$

$$\sigma^2(x^*) = C(x^*, x^*) - C(x^*, X)\left[C(X, X) + \sigma_n^2 I\right]^{-1} C(X, x^*), \tag{2.16}$$

where $C(x^*, x^*)$ represents the covariance matrix between the test inputs and themselves. $C(x^*, X)$ represents the covariance matrix between the test inputs and the training inputs. $C(X, X)$ represents the covariance matrix between the training inputs and themselves.

2.4.2 Sensor Fusion with Gaussian Processes

Gaussian Processes (GPs) have proved useful and have been widely used for sensor fusion. In (Solak et al., 2003), Gaussian Processes provide an approach to nonparametric modelling which allows a straightforward combination of function and derivative observations in an empirical model. Murray-Smith & Pearlmutter (2005) generalized arbitrary transformations on GP priors through linear transformations and proposed the use of transformed Gaussian Process priors for estimating the derivatives of noisy sensor measurements, and fusing information from multiple sensors. Recent work (Vasudevan, 2012) addressed the problem of fusing multiple sets of heterogeneous sensor data using Gaussian Processes, which were applied for terrain data fusion.

Data fusion in the context of Gaussian Processes has been investigated in the literature. El-Beltagy & Wright (2001) presented the use of Gaussian Processes for the fusion of results of computational simulations with varying degrees of accuracy and computational loads. Girolami (2006) applied Gaussian Process priors method to large scale bioinformatics problems and illustrated this with a protein fold prediction problem.

2.5 Conclusions

In Chapter 1, we present that the Kinect can be used for indoor context-aware sensing applications. It is a human motion tracking system, where the position uncertainty and latency are two main issues. It can be augmented by mobile inertial sensors through sensor fusion. The probabilistic approaches, i.e. GPs, are well suited for fusing the Kinect sensor and mobile inertial sensors as GPs provide useful results, on the basis of which we develop two applications.

In this chapter, we describe the context-aware sensing systems, among which the location-aware applications play an important role. We briefly discuss the positioning technologies and then present the techniques used for human motion capture and analysis. Considering the advantages and disadvantages of different sensors, and the need for fusing data from multiple sensors, we give a brief introduction to multisensor data fusion. We describe the probabilistic approaches for sensor fusion and introduce the widely used sensor fusion techniques, including the Kalman filter and its variants methods. Considering the limitations of Bayesian parametric models, we present the Gaussian Processes (GPs) framework. We discuss the work on sensor fusion with GPs in the literature and highlight the benefits of sensor fusion with GPs, i.e. the log-likelihood and the prediction.

Chapter 3

Sensor Fusion with Multi-rate Sensors-based Kalman Filter

In this chapter, we propose a new coordinate system transformation method for estimating the skeleton joint acceleration in the Kinect coordinate system through inertial sensor fusion. We present a multi-rate sensors-based Kalman filter approach to fusing the low-sampling-rate positions sensed by the Kinect and the high-sampling-rate accelerations measured by the built-in inertial sensors in a mobile device. We conduct an experiment to test this. By comparing the accelerations measured by different sensors, we conclude that these accelerations are comparable and the coordinate conversion method is effective. We propose that the high-sampling-rate acceleration can augment the low-sampling-rate position measurements and present a multi-rate sensors-based Kalman filter for fusing the Kinect sensor and inertial sensors equipped in a mobile device. This improves the system state estimation, i.e. the position, the velocity and the acceleration of the skeleton joints. We conduct an experiment to test the proposed multi-rate sensors-based Kalman filter sensor fusion approach and draw the conclusions.

3.1 Introduction

As discussed in Chapter 1, a Kinect can be used for position tracking in the location-aware sensing applications, where the accuracy of position tracking is critical. Due to the position jitter in the Kinect skeleton tracking, we need additional sensors to augment the Kinect. We discussed the complementary properties of the Kinect sensor and mobile inertial sensors in section 1.2.2. We need sensor fusion techniques that take account of all the sensor characteristics to deal with the Kinect position uncertainty.

In this chapter, a multi-rate sensors-based Kalman filter is designed and applied for fus-

ing the low-sampling-rate (30Hz) uncertain positions sensed by the Kinect sensor and the high-sampling-rate (90Hz) accelerations measured by a sensor pack composed of inertial sensors. Only one sensor pack is on the body, e.g. held in the hand. These sensors have complementary properties. The Kinect can be applied for skeleton tracking, which gives the joints positions. Meanwhile, the built-in inertial sensors in the mobile device sense the hand motion. The hand acceleration can be estimated through inertial sensor fusion. The high frequency acceleration measurements can compensate for the effect of position uncertainty, thus augment the low-sampling-rate, noisy positions sensed by the Kinect. In addition to the different sampling rates, these sensors have different noise characteristics. A multi-rate Kalman filter is well suited for this sensor fusion.

In this section, we will give an introduction of using a Kinect for location-aware sensing applications and discuss the importance and benefits of robust estimation of position, velocity and acceleration in HCI.

In addition to the position, the velocity and acceleration estimation also play an important role in HCI, e.g. a velocity control for exploring 3D graphical environments (Paton & Ware, 1994), velocity control cursor mechanisms (Crossan et al., 2009) and pointing tasks (Gallo & Minutolo, 2012). Moreover, the benefits of acceleration measurements in velocity estimation were introduced in (Jeon & Tomizuka, 2007).

We need to augment a Kinect sensor with mobile inertial sensors in order to get a robust estimation of position, velocity and acceleration. With only a Kinect sensor, the velocity and acceleration estimation are not robust. With only inertial sensors, drift happens for velocity estimation by integrating acceleration and position estimation by double-integrating acceleration. The Kinect sensor measures the joints position while the inertial sensors in a handheld device can be used for estimating the acceleration. Fusion of these sensors can improve the system state estimation, including the position, the velocity and the acceleration.

In this chapter, we will discuss the problem of fusing the Kinect sensor and the inertial sensors for improving the system state estimation with the proposed multi-rate sensors-based Kalman filter method. As an example, we analyse the hand motion. Our goal is to explore the complementary properties and fuse these sensors for improving the system state estimation. We focus on indoor augmented reality (AR) application by fusing the inertial sensors and external position sensing for the location-aware application.

3.2 The Kalman Filter and Multi-rate Sensors-based Kalman Filter

In this section, we first give an introduction of the Kalman filter framework and its applications. Then we discuss the literature work on using the multi-rate Kalman filter for sensor fusion.

3.2.1 Background

The Kalman filter and its variants have been widely used for sensor fusion. Some variants of the classic Kalman filter have been developed to take into account the different noise characteristics of multiple sensors. The multi-rate Kalman filter is a popular method for fusing data sampled at different sampling rates. The integration of global positioning system (GPS) and inertial navigation system (INS) with a multi-rate Kalman filter has been thoroughly researched in the literature (Caron et al., 2006; Qi & Moore, 2002). In addition to the systems with no delays, several works have been done for coping with the delayed measurements in the Kalman filter framework (Peñarrocha et al., 2012). Position uncertainty and lag have been two common problems for the interactive systems (Foxlin, 1996). In (Friedmann et al., 1992), the delay in computer systems was coped with an optimal linear filter method. The Kalman filter method was presented for predicting the future user position. In the virtual reality application (Liang et al., 1991), a predictive Kalman filter was designed to compensate for the delay in orientation data and a low pass filter was designed to reduce the noise in position data. Lu et al. (2005) presented a reorganization technique of studying the linear minimum variance estimation in discrete-time systems with delayed measurements. In (Gopalakrishnan et al., 2011), the Extended Kalman filter was applied for state estimation in the presence of delayed and infrequent measurements. In recent years, Peñarrocha et al. (2012) addressed the state estimation problem in linear time-varying systems with several sensors with different availability, randomly sampled in time, and whose measurements have a time-varying delay.

The multi-rate Kalman filter has been widely used for sensor fusion. Different sensors may have different sampling rates and different noise characteristics. The multi-rate Kalman filter is well suited for sensor fusion. Niwa et al. (1999) proposed a sensor fusion system that combined the visual sensor and inertial sensors using a modified Kalman filter. The obvious difference between this filter and the standard Kalman filter is the use of time-varying filter gain matrix. Armesto et al. (2007) applied the multi-rate EKF and UKF for fusing vision and inertial measurements. Smyth & Wu (2007) proposed a multi-rate Kalman filter for fusion of displacement and acceleration response measurements in dynamical system monitoring.

Caron et al. (2006) proposed a GPS/IMU multisensor fusion algorithm, taking context into consideration. Lee & Tomizuka (2003) considered optimal state estimation problems with a multi-rate Kalman filter, where the position sensor data and accelerometer measurements were utilized. The multi-rate Kalman filter proved effective.

Integration of visual and inertial sensing modalities opens new application directions for robotics and other fields (Corke et al., 2007). Hol et al. (2007) proposed a method for estimating the position and orientation (pose) of a camera by fusing measurements from inertial and vision sensors. In particular, position measurements are vital for end-effector control in industrial robots. Kinematic Kalman filter based on the kinematic model is called the Kinematic Kalman Filter (KKF) (Jeon & Tomizuka, 2007). Jeon et al. (2009) applied the KKF for fusing vision sensor and inertial sensors. Wang et al. (2012) used the vision sensor for robot end-effector position sensing and applied the KKF to fuse the position measurements and inertial sensor data.

3.2.2 Sensor Fusion with Multi-rate Sensors-based Kalman Filter

We consider the problem of fusing the low-sampling-rate positions sensed by the Kinect sensor and the high-sampling-rate accelerations measured by the mobile inertial sensors with a multi-rate Kalman filter.

We have given a brief description on the Kalman filter in section 2.3.3. To continuously estimate the motion state of the hand in 3D space with a KF, we need a process model and a measurement model. For the moving target tracking problem, the state refers to the physical state, which can be described by dynamic variables, such as position, velocity and acceleration of a moving object. The noise in the measurements means that there is a certain degree of uncertainty in them. The dynamical system evolves as a function of time, and there is also noise in the dynamics of the system, that is, the process noise.

Due to the different sampling rates and different noise characteristics of these sensors, we need a multi-rate fusion technique. The idea behind the multi-rate fusion is to build up a size-varying output vector, containing only those measurements that have been sampled at each time instant (Gemeiner et al., 2007). The Kalman filter is such a technique that fuses sensor data from multiple sources and prior information about the behaviour of the system in order to make a most-likely estimate of the system state (Luinge, 2002). Thus, the multi-rate Kalman filter is well suited for fusing the Kinect sensor and inertial sensors.

As an example, we analyse the hand motion along $x-$axis. To continuously estimate the

motion state of the hand, we apply the following discrete-time state space model:

$$x_k = Ax_{k-1} + w_{k-1}. \tag{3.1}$$

In our method, the state vector is composed of position, velocity and acceleration. $x_k = \begin{bmatrix} p_k & \dot{p}_k & \ddot{p}_k \end{bmatrix}^T$, that is $x_k = \begin{bmatrix} p_k & v_k & a_k \end{bmatrix}^T$.
The Kinect measures the hand position. The acceleration can be estimated through inertial sensor fusion. The accelerometer measures the total external force. The rotation is measured by the gyroscope. The accelerations sensed by the inertial sensors and expressed in the Kinect frame are the second derivatives of the positions in the high-sampling-rate space.

We define the following state transition matrix:

$$A = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ & 1 & \Delta t \\ & & 1 \end{bmatrix}, \tag{3.2}$$

where $\Delta t = \frac{1}{90}$s.

We apply the continuous Wiener process acceleration model (Bar-Shalom et al., 2004). The acceleration is perturbed with a white noise process with power spectral density q, which is determined based on the experiment training data. The process noise covariance matrix is

$$Q = \begin{bmatrix} \frac{1}{20}\Delta t^5 & \frac{1}{8}\Delta t^4 & \frac{1}{6}\Delta t^3 \\ \frac{1}{8}\Delta t^4 & \frac{1}{6}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{6}\Delta t^3 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} q. \tag{3.3}$$

The Kinect sensor and inertial sensors are sampled at a rate of 30Hz and 90Hz, respectively. For the multi-rate Kalman filter, $\Delta t_m = \frac{1}{90}$s. In order to compare the sensor fusion system with a position-only system (Kinect), we design a single rate Kalman filter, which only uses the Kinect position measurements for updating the system state. $\Delta t_s = \frac{1}{30}$s. Due to the availability of the different measurements, the measurement model should be changed accordingly.

$$p_k = H_1 x_k + v_k^{'} \tag{3.4}$$

$$a_k = H_2 x_k + v_k^{''}, \tag{3.5}$$

where $v_k^{'}$ and $v_k^{''}$ are the position observation noise and the acceleration measurement noise respectively, as defined in equation (2.2).

When the position observation is available,

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{3.6}$$

During the time period when only the acceleration measurement is available,

$$H_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

Besides the mean estimation, we can also get the uncertainty of the mean estimation through the posteriori estimate error covariance matrix P_k, as defined in equation (2.7).

3.3 System Overview

Equipment

The equipment consists of the Microsoft Kinect and a SHAKE SK7. The Sensing Hardware Accessory for Kinaesthetic Expression (SHAKE) SK7[1] sensor pack (Williamson et al., 2007) contains triple axis accelerometer, triple axis Angular Rate Sensor (Gyroscope) and triple axis magnetometer. Besides the inertial sensors, the Microsoft Kinect is used for skeleton tracking, which gives the skeleton joints positions. In our work, the Kinect sensor is applied for sensing the hand positions.

The frame rate of the Kinect sensor is 30Hz, whereas the inertial sensors are sampled at a rate of 90Hz. The OpenNI drivers and the motion tracking middleware NITE are used (OpenNI, 2014).

3.3.1 Sensor Noise Characteristics

Kinect Sensor Uncertainty

Figure 3.1 illustrates that the uncertainty of the position measurements sensed by the Kinect. The right panel shows the noisy hand position measurements of uncertainty and the right panel shows the histogram of the position uncertainty and its Gaussian fit. The uncertainty is measured to be (SD) $\sigma_p = 8$mm.

[1] http://code.google.com/p/shake-drivers/

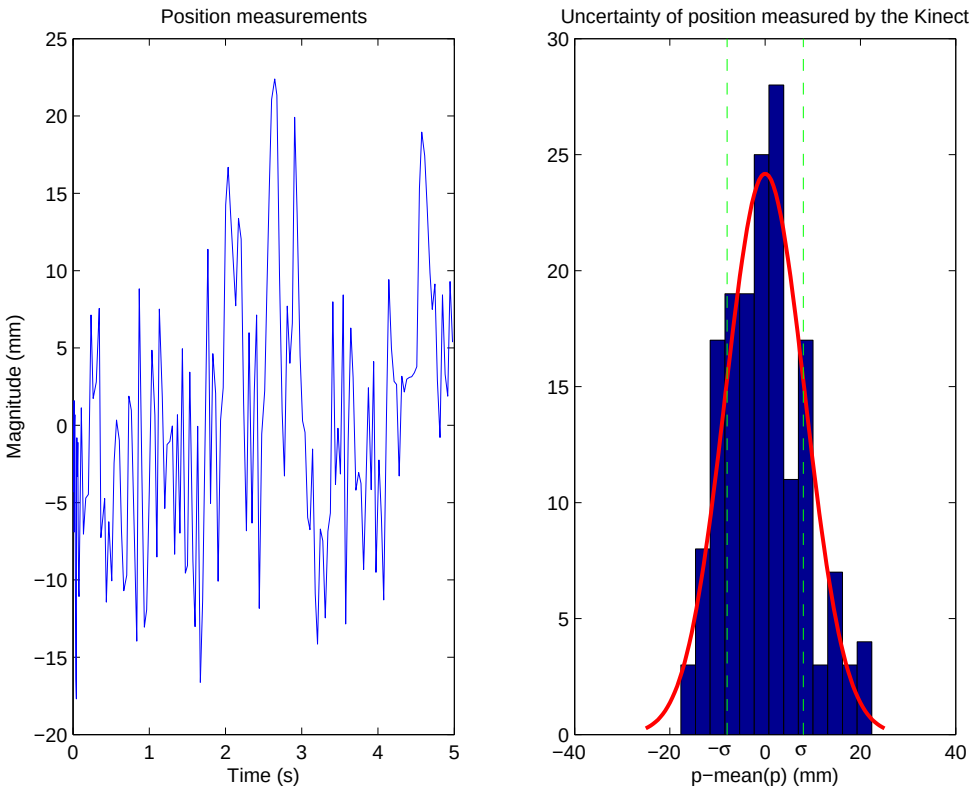


Figure 3.1: The uncertainty of the position measurements sensed by the Kinect. Left panel: position measurement of uncertainty. Right panel: The histogram of position uncertainty and its Gaussian fit ($\sigma_p = 8$mm).

Inertial Sensor Uncertainty

Figure 3.2 illustrates the uncertainty of the acceleration measured by mobile inertial sensors. The left panel shows the noisy acceleration measurement of uncertainty and the right panel shows the histogram of this uncertainty and its Gaussian fit. The uncertainty of acceleration measurement is (SD) $\sigma_a = 100\text{mm/s}^2$ in the Kinect coordinate system.

3.3.2 The Coordinate Systems

When fusing the Kinect sensor and SK7 inertial sensors, several coordinate systems need to be introduced:

- Earth's North-East-Down (NED) frame (e): This is SK7's reference frame. Our algorithm uses data from accelerometer, gyroscope and magnetometer to calculate SK7's attitude (Pitch and Roll angles) and magnetic azimuth (compass heading) with respect to Earth's North-East-Down frame of reference.

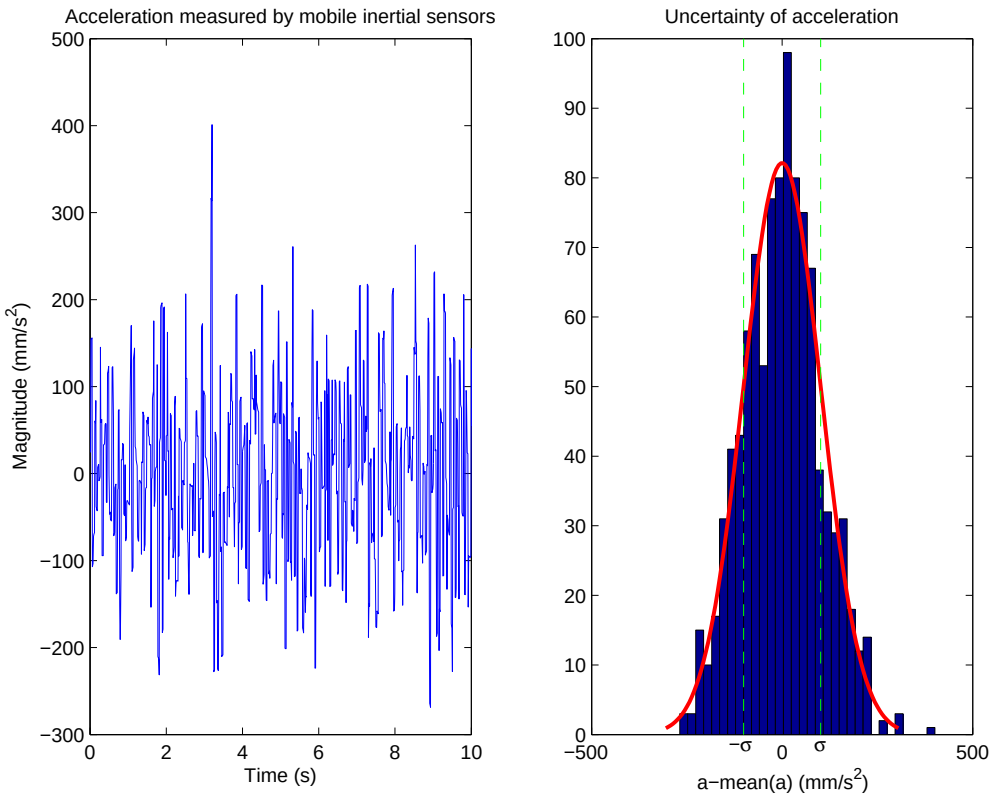


Figure 3.2: The uncertainty of the acceleration measured by mobile inertial sensors (SK7). Left panel: acceleration measurement of uncertainty. Right panel: The histogram of acceleration uncertainty and its Gaussian fit ($\sigma_a = 100\text{mm/s}^2$).

- Kinect frame (k): The skeleton joint positions, i.e. the 3D coordinates of the joint, are expressed in this coordinate system.
- SK7 body frame (b): This is the coordinate system of the SK7. Accelerometer data, gyroscope data and magnetometer data are all expressed in the SK7 body frame.

The Relationships between Different Coordinate Systems

In order to convert coordinates from one coordinate system to another, we need to find the relationships between different coordinate systems (Titterton et al., 2004). We propose a coordinate system transformation method for converting the linear acceleration from the body frame to the Kinect frame.

Coordinate Conversion from SK7 Body Frame to Earth's NED Coordinate System

The inertial sensor data is filtered with a Kalman filter and then fused with a complementary filter. With the Euler angles, we compute the rotation matrix for converting the coordinate

from SK7 body frame to Earth's NED coordinate system.

The accelerometer measures the linear acceleration plus the acceleration due to gravity. The linear acceleration expressed in Earth's NED frame could be computed in the following way:

$$\mathrm{linearAcc}_e = R_b^e \times a_b - g_e. \tag{3.8}$$

Gravity g_e is a constant vector in earth frame. When expressed in the body frame, the altered gravity depends on the orientation of SK7.

$$\mathrm{linearAcc}_e = R_b^e \times (a_b - g_b). \tag{3.9}$$

In our method, we infer the effect of gravity on accelerometer data by estimating the gravity components in the SK7 body frame with equation (3.18). Gravity expressed in the SK7 body frame is denoted as g_b in equation (3.9). We use the rotation matrix R_b^e to convert the linear acceleration $a_b - g_b$ from SK7 body frame to Earth's NED frame. The rotation matrix R_b^e is computed with equation (3.10), where ϕ, θ and ψ represent the Roll, Pitch and Heading, respectively,

$$R_b^e = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}. \tag{3.10}$$

Coordinate Conversion from Earth's NED Coordinate System to the Kinect Coordinate System

Due to the fact that Earth's NED reference frame does not coincide with the Kinect coordinate system, they are separated by a constant rotation. In order to convert the 3D coordinates from Earth's NED reference frame to the Kinect coordinate system, we need to determine this rotation matrix. The calibration process is as follows:

We measure the rotation angles between Earth's NED frame and the Kinect coordinate system. Here the Earth's NED frame is the reference frame. Firstly, rotate through angle $\hat{\psi}$ about reference $z-$axis. Secondly, rotate through angle $\hat{\theta}$ about the current $y-$axis. Finally, rotate through angle $\hat{\phi}$ about the current $x-$axis. We get

$$R_e^k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\phi} & \sin\hat{\phi} \\ 0 & -\sin\hat{\phi} & \cos\hat{\phi} \end{bmatrix} \begin{bmatrix} \cos\hat{\theta} & 0 & -\sin\hat{\theta} \\ 0 & 1 & 0 \\ \sin\hat{\theta} & 0 & \cos\hat{\theta} \end{bmatrix} \begin{bmatrix} \cos\hat{\psi} & \sin\hat{\psi} & 0 \\ -\sin\hat{\psi} & \cos\hat{\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.11}$$

With the rotation matrix R_e^k in equation (3.11), we can calculate the linear acceleration ex-

pressed in the Kinect coordinate system:

$$\text{linearAcc}_k = R_e^k \times \text{linearAcc}_e. \tag{3.12}$$

The accelerometer output data is scaled to 1000 units per g ($g = 9.81\text{m/s}^2$). The rotation matrix R_e^k will not change unless the Kinect is moved and the rotation angles are changed. Thus, we only need to do this calibration once.

3.3.3 The Multi-rate Sensors-based Fusion System

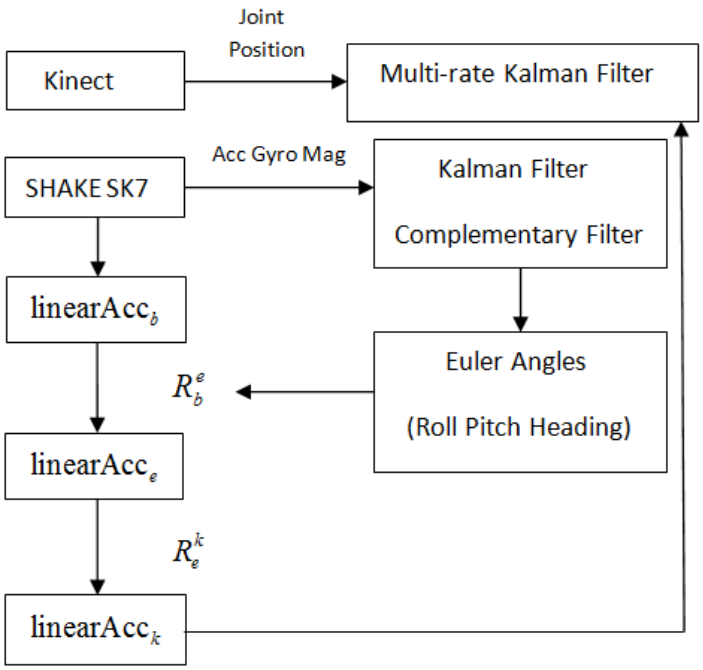


Figure 3.3: The diagram illustrates the process of fusing the Kinect sensor and the inertial sensors with the multi-rate sensors-based Kalman filter. It also shows that how we estimate the acceleration through inertial sensor fusion.

The system uses the Microsoft Kinect and the SHAKE SK7. In the multi-rate Kalman filter-based sensor fusion system, the skeleton tracking positions sensed by the Kinect are sent to the system. Meanwhile, the inertial sensor data is sent to the system via Bluetooth. The inertial sensor data is filtered with a Kalman filter and then fused in a complementary filter. We compute the rotation matrix based on the Euler angles (Roll, Pitch and Heading). With this rotation matrix, we convert the linear acceleration from SK7 body frame to Earth's North-East-Down (NED) coordinate system. There is a constant rotation matrix for converting the coordinates from Earth's NED coordinate system to the Kinect coordinate system. With the proposed orientation calculation and coordinate transformation methods, we get the hand acceleration estimated with inertial sensors and expressed in the Kinect frame.

Following this, we fuse the Kinect position and the estimated acceleration in a multi-rate Kalman filter. Due to the different sampling rates of these sensors, we adjust the measurement models based on the availability of the observations. This is the multi-rate Kalman filter. The flowchart of the sensor fusion approach is shown in Figure 3.3.

3.4 Inertial Sensor Fusion

Since we cannot determine a rotation matrix for converting the movement acceleration from the body frame to the Kinect coordinate system directly, we need to use the proposed coordinate system transformation method described in section 3.3.2. In order to compute the rotation matrix R_b^e in equation (3.10), we need to compute the Euler angles, including Roll, Pitch and Heading. In this section, firstly, we present the inertial sensor fusion approach. Following this, we conduct an experiment and test the proposed inertial sensor fusion method for estimating the hand acceleration in the Kinect coordinate system.

3.4.1 Orientation Estimation

In this part, we discuss how to fuse the inertial sensor data with a complementary filter. The estimated Euler angles are used to calculate a rotation matrix for distinguishing the effect of gravity from the accelerometer data and estimate the linear acceleration in the body frame. How to convert the linear acceleration, i.e. the movement acceleration, from the body frame to the Kinect frame has been discussed in section 3.3.2.

Orientation Estimation with Accelerometer

Now we introduce how to calculate Roll and Pitch with a 3-axis accelerometer. For Roll ϕ calculation:

$$\tan\phi = \frac{a_y}{a_z}. \tag{3.13}$$

A slightly modified version (equation (3.14)) can be found in (Pedley, 2013).

$$\tan\phi = \frac{a_y}{sign(a_z)\sqrt{\mu \cdot {a_x}^2 + {a_z}^2}}, \mu = 0.01, \tag{3.14}$$

where $sign(a_z)$ has a value $+1$ if a_z is non-negative and -1 if a_z is negative. For Pitch θ calculation:

$$\tan\theta = -\frac{a_x}{\sqrt{{a_y}^2 + {a_z}^2}}, \tag{3.15}$$

where a_x, a_y and a_z represent the filtered accelerometer data along $x-$axis, $y-$axis and $z-$axis respectively. When there is no visible and vigorous movement and the accelerometer mainly measures the acceleration due to gravity, this method works well. When there is visible and vigorous movement, we need additional sensors including gyroscope and magnetometer to get a complete orientation estimation with respect to the Earth's NED coordinate system.

Orientation Estimation with Complementary Filter

In our work, we use a complementary filter to calculate the Euler angles including Roll, Pitch and Heading. The angle complementary filter is

$$\varphi = \alpha_1 \times (\varphi + \omega \times dt) + \alpha_2 \times (\varphi_{acc}), \tag{3.16}$$

where φ denote the Euler angles, $\alpha_1 = 0.98$ and $\alpha_2 = 0.02$.

With a $3-$axis accelerometer and a $3-$axis magnetometer, we can calculate the Roll, Pitch and Heading (Caruso, 1997; Ozyagcilar, 2012). The $3-$axis gyroscope can be used to update the angles. This is the angle complementary filter. We apply a Kalman filter for filtering the inertial sensor data first and then fuse the filtered inertial sensor data in a complementary filter to calculate the device's attitude (Roll and Pitch) and magnetic azimuth (compass heading) with respect to Earth's North-East-Down frame of reference. The heading ψ is calculated using the horizontal components of the transformed magnetometer vector according to equation (3.17).

$$\tan\psi = \frac{m_z \sin\phi - m_y \cos\phi}{m_x \cos\theta + m_y \sin\theta \sin\phi + m_z \sin\theta \cos\phi}, \tag{3.17}$$

where m_x, m_y, m_z represent the $3-$axis magnetometer data, ϕ denotes Roll and θ denotes Pitch.

In this way, we get the Euler angles, which are in the order XYZ. They describe the orientation of the device relative to the earth frame of reference in terms of a rotation first by Roll about its X axis followed by a rotation of Pitch about its Y axis, and finally a rotation by Heading about its Z axis.

Estimation of Gravity and Linear Acceleration

Since we have the orientation information, we can estimate the effect of gravity in SK7 body frame. Gravity is a constant vector in earth frame. We denote it as $g_e = \begin{bmatrix} 0 & 0 & 1000 \end{bmatrix}^T$.

When expressed in the SK7 body frame,

$$g_b = R_e^b \cdot g_e = \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} \times 1000, \tag{3.18}$$

where θ denotes Pitch and ϕ denotes Roll. R_e^b is the rotation matrix for converting the coordinates from the earth frame to the body frame. We can compute R_e^b with equation (3.19).

$$R_e^b = (R_b^e)^T, \tag{3.19}$$

where R_b^e is the rotation matrix defined in equation (3.10).

In this way, we find the gravity components expressed in the body frame. The linear acceleration is $a_b - g_b$, which is expressed in the body frame. The following work is to convert this linear acceleration from the body frame to the Kinect frame. This has been introduced in section 3.3.2.

3.4.2 Experiment: Comparison of Acceleration Estimated with Kinect Sensor and Inertial Sensors

Experiment Set-up

To test the feasibility of estimating the hand acceleration with the proposed method, we conducted an experiment. The user stood in front of the Kinect with the mobile device (SK7) held in the hand. At the beginning, the user put the hand at the side of the hip. Then the user raised the hand and walked a few steps towards the Kinect. Finally, the user stopped and lowered the hand. The whole process took 30s.

For this experiment, the calibration parameters for calculating the constant rotation matrix R_e^k are $\hat{\psi} = \frac{-30}{180} \times \pi, \hat{\theta} = 0, \hat{\phi} = -\frac{\pi}{2}$.

Experimental Results

Position Measurements The hand motion data includes the position measurements sensed by the Kinect, which is shown in Figure 3.4, and the inertial sensor data. In Figure 3.4, we show that the whole process can be divided into 6 time periods, each of which corresponds with a movement. We can see clearly that these periods include the 1st period (the hand was motionless), the 2nd period (raise the hand), the 3rd period (the hand was motionless), the 4th period (walk), the 5th period (the hand was motionless) and the 6th period (lower the hand).

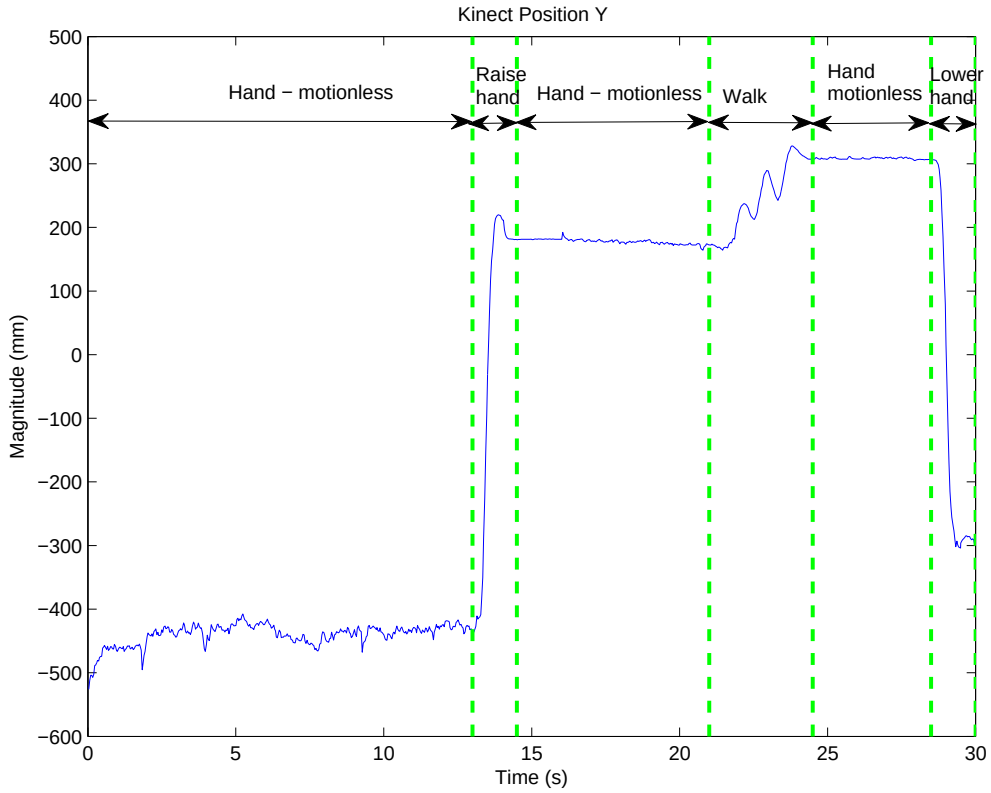


Figure 3.4: The hand position along the y-axis (the vertical axis) measured by the Kinect sensor. The changing of the hand position measured by the Kinect during the 6 time periods is illustrated clearly.

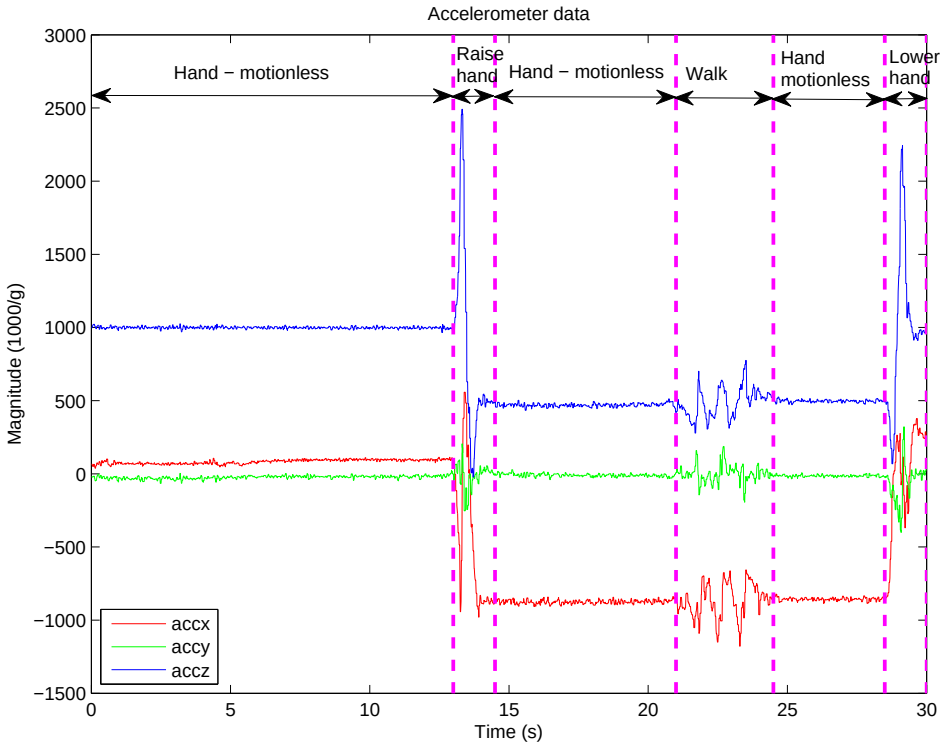


Figure 3.5: The accelerometer data shows the changes of the total acceleration (the linear acceleration plus the acceleration due to gravity) during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

Inertial Sensor Data In addition to the Kinect position measurements, we also have the motion data sensed by the mobile inertial sensors.

Figure 3.5 illustrates the changes of accelerometer data (the total acceleration equals the linear acceleration plus the acceleration due to gravity) during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

The gyroscope data and the magnetometer data are shown in Figure 3.6 and Figure 3.7, respectively.

Euler Angles We calculate the orientation through inertial sensor fusion. The Euler angles, including Roll, Pitch and Heading, are shown in Figure 3.8.

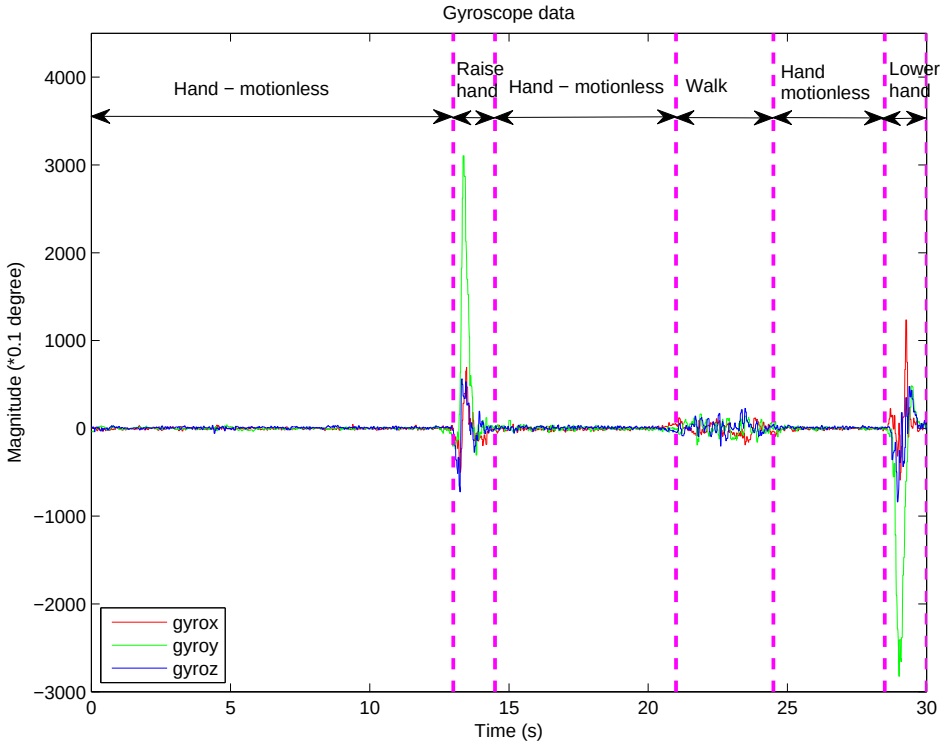


Figure 3.6: The gyroscope data during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

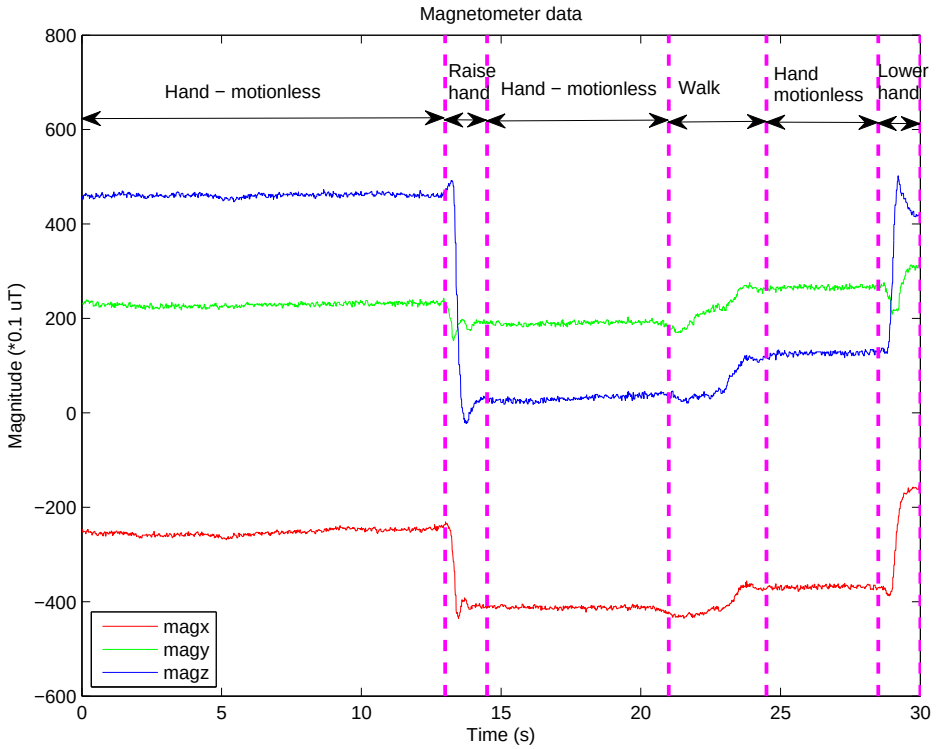


Figure 3.7: The magnetometer data during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

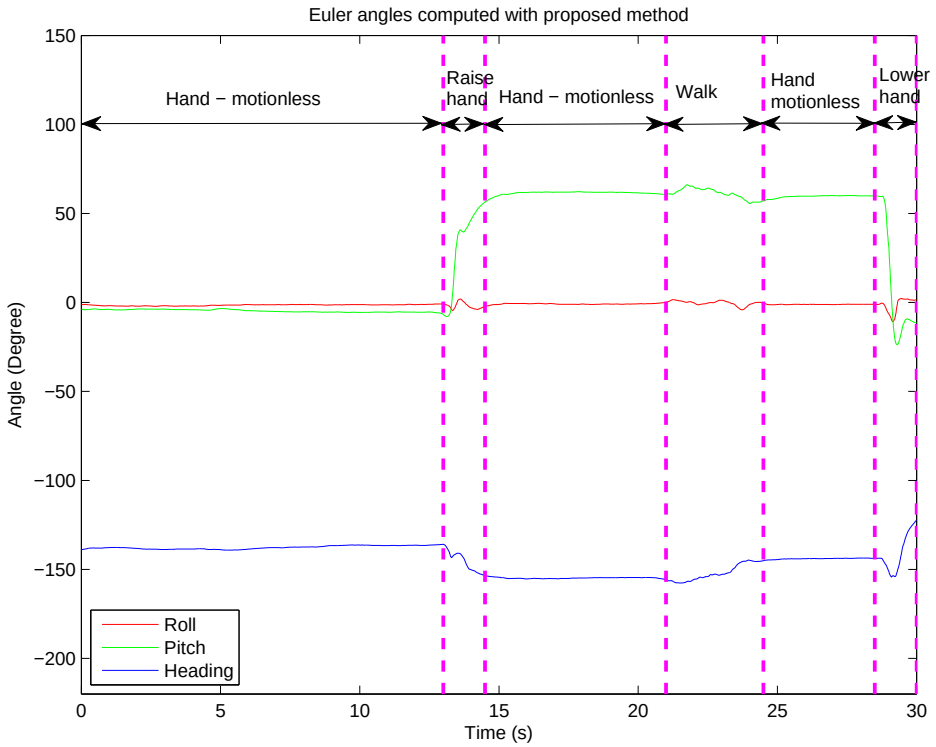


Figure 3.8: The Euler angles (Roll, Pitch and Heading) calculated with the proposed inertial sensor fusion method. Plots show the Euler angles during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

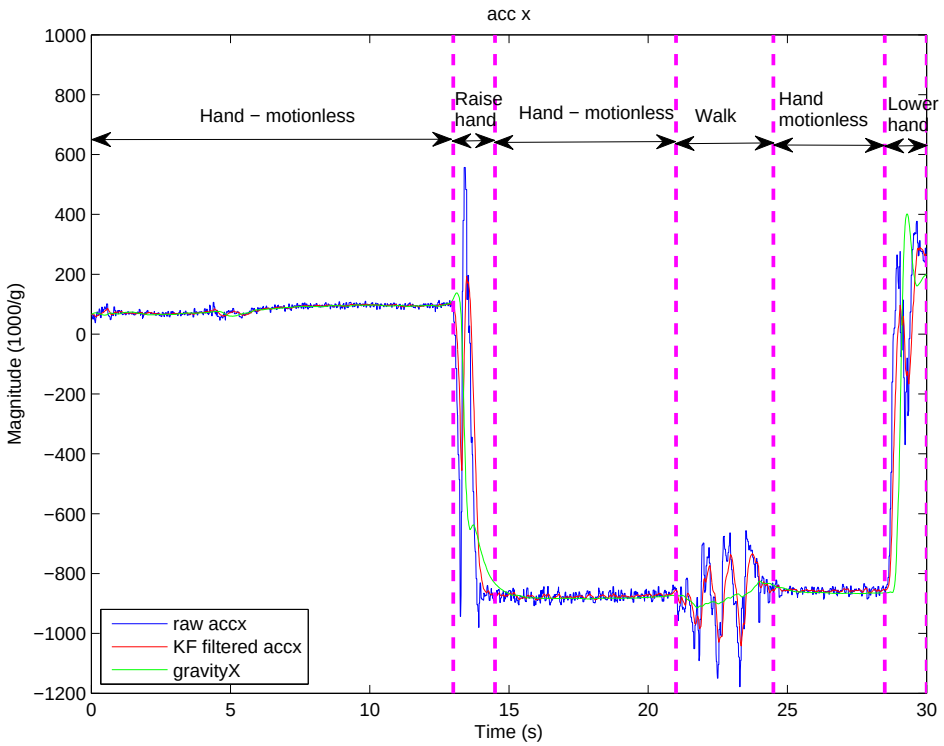


Figure 3.9: Acceleration along $x-$axis in the body frame: The raw acceleration along $x-$axis, the filtered results and the estimated effect of gravity along $x-$axis.

Estimation of Gravity and Linear Acceleration in SK7 Body Frame The Euler angles can be used to calculate the rotation matrix R_b^e in equation (3.10) and R_e^b in equation (3.19).

With the proposed method discussed in section 3.4.1, we infer and subtract the effect of gravity from the accelerometer data, and estimate the linear acceleration in the body frame. The accelerometer data, the filtered accelerometer data and the estimated gravity in the body frame are shown in Figure 3.9 (along $x-$axis), Figure 3.10 (along $y-$axis) and Figure 3.11 (along $z-$axis), respectively.

By subtracting the effect of gravity from the accelerometer data, we get the linear acceleration in the body frame. The result is shown in Figure 3.12.

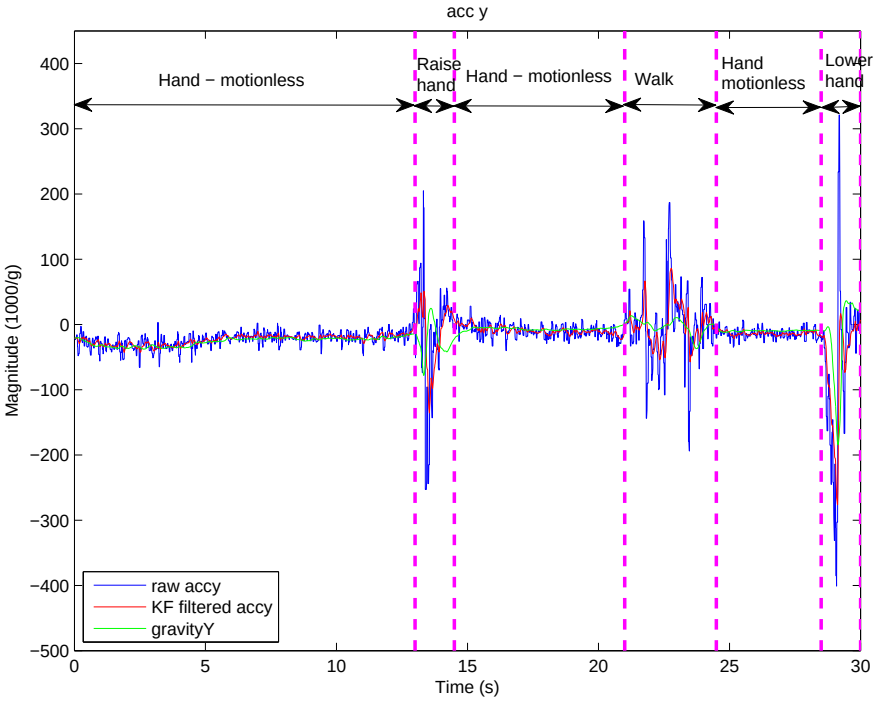


Figure 3.10: Acceleration along $y-$axis in the body frame: The raw acceleration along $y-$axis, the filtered results and the estimated effect of gravity along $y-$axis.

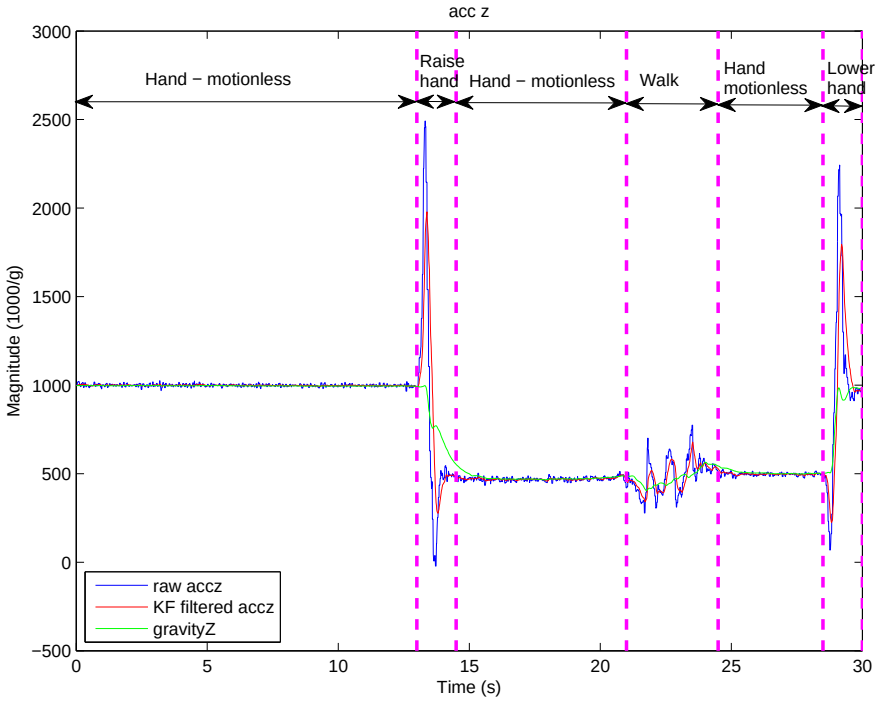


Figure 3.11: Acceleration along $z-$axis in the body frame: The raw acceleration along $z-$axis, the filtered results and the estimated effect of gravity along $z-$axis.

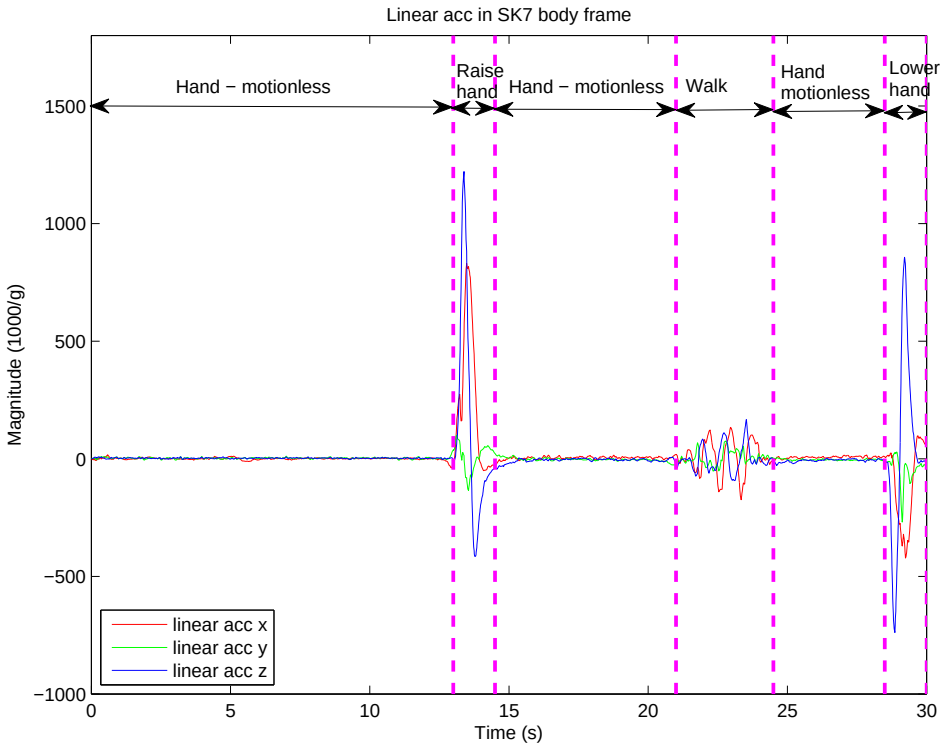


Figure 3.12: The estimated linear acceleration in the body frame with the effects of gravity inferred and subtracted from the accelerometer data. Plots show the changes of the linear acceleration during the different time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

Comparison of the Accelerations The hand acceleration in the body frame needs to be converted into the acceleration in the Kinect frame for comparison with the hand acceleration of the Kinect positions. With the method proposed in section 3.3.2, we convert the linear acceleration (Figure 3.12) from the body frame to the Kinect coordinate system.

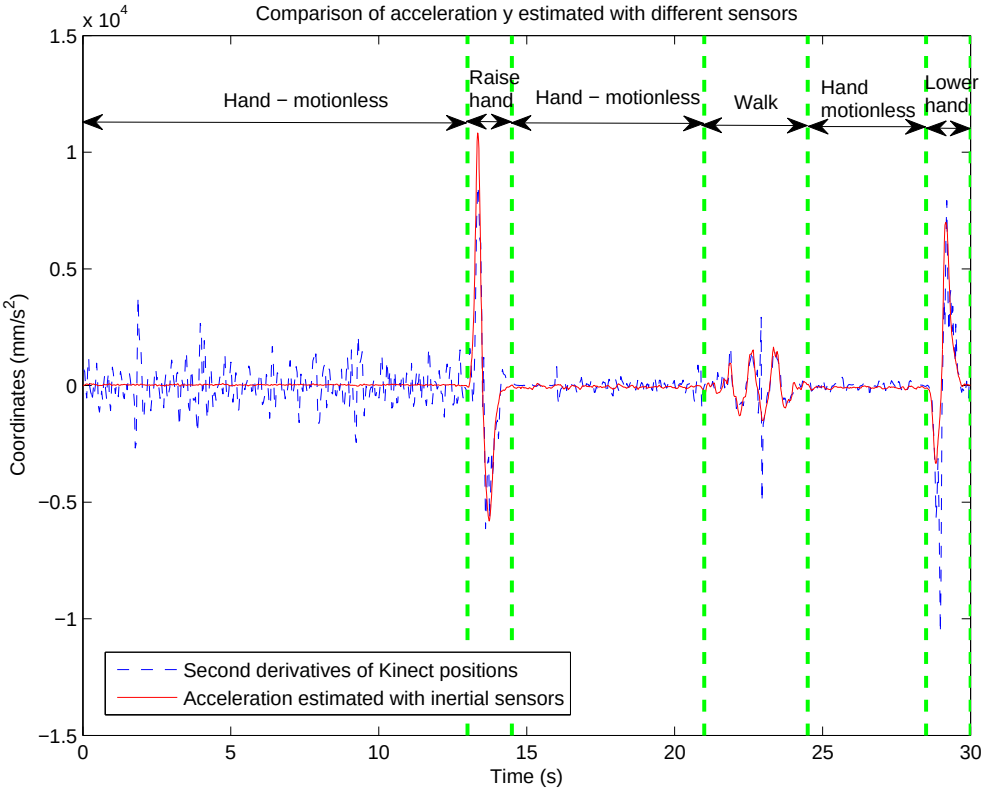


Figure 3.13: Comparison of the hand acceleration (along $y-$axis) estimated with different sensors: the second derivatives of Kinect positions (the dashed line) and the acceleration estimated with inertial sensors (the solid line). The changes of the hand acceleration estimated with the inertial sensors during the 6 time periods are illustrated. The 6 time periods: (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand.

Figure 3.13 illustrates the acceleration comparison results. The converted acceleration (the solid line) and the second derivatives (the dashed line) of the Kinect positions are shown. The changes of the hand acceleration estimated with the inertial sensors during the 6 time periods are illustrated. The 6 time periods include (1) hand–motionless (2) raise the hand (3) hand–motionless (4) walk (5) hand–motionless (6) lower the hand. By comparing this acceleration with the second derivatives of the Kinect positions, we can see that the hand acceleration estimated with inertial sensors and those estimated with the Kinect sensor are comparable and match well.

We can estimate the displacement (position), denoted as d, by double integrating the accel-

eration a measured by the mobile inertial sensors.

$$d = \int_0^t \int_0^t a \, \mathrm{d}t\mathrm{d}t. \tag{3.20}$$

However, we found the drift problem with this displacement (position) estimation. Figure 3.14 illustrates the drift for position estimation by double integrating the acceleration measured by mobile inertial sensors.

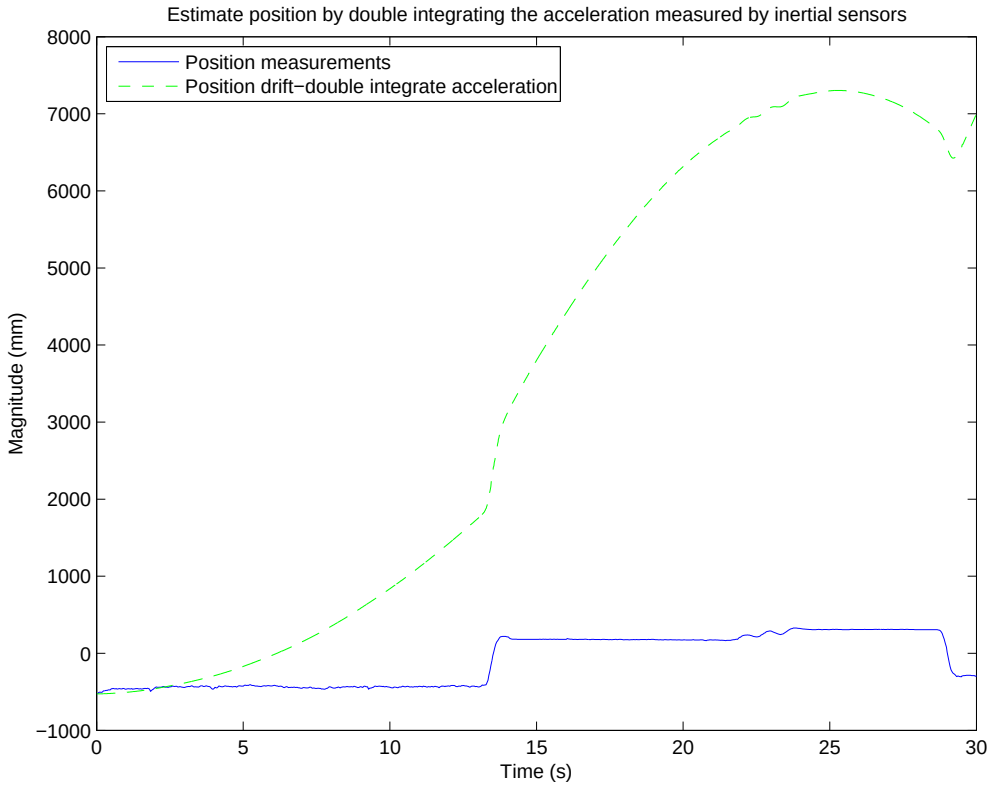


Figure 3.14: By double integrating the hand acceleration (along $y-$axis) estimated with inertial sensors, we found drift problem with position estimation (the dashed line) in comparison with the position measurements (the solid line).

Conclusion

In this experiment, we tested the feasibility of estimating the hand acceleration with inertial sensors in the Kinect coordinate system. The hand movements are the typical human movements, e.g. the hand is motionless, raise or lower the hand, the hand movement during walking, in people's everyday lives. Experimental results show that the proposed method can be successfully applied for estimating the hand acceleration in the Kinect coordinate system. The accelerations estimated with the Kinect sensor and those estimated with inertial sensors are comparable and match well. This proves that our method is effective for hand acceleration estimation with inertial sensors in the Kinect coordinate system.

3.5 Experiment: Fusing Kinect Sensor and Inertial Sensors with Multi-rate Sensors-based Kalman Filter

We conducted this experiment to test the sensor fusion with the multi-rate Kalman filter, specifically, the multi-rate sensors-based Kalman filter.

3.5.1 Experimental Set-up

The Kinect was put in a fixed place in the room. We pre-defined two spatial location points 1 and 2, which were located on the $x-$axis of the Kinect frame. A user stood in front of the Kinect with the mobile device held in the right hand. When the Kinect started tracking, the user put his hand at location 1 and kept the hand as motionless as possible for a while, then moved the hand to location 2 and stayed motionless for a while. The whole process took 5s. Figure 3.15 illustrates the diagram of this hand movement experiment.

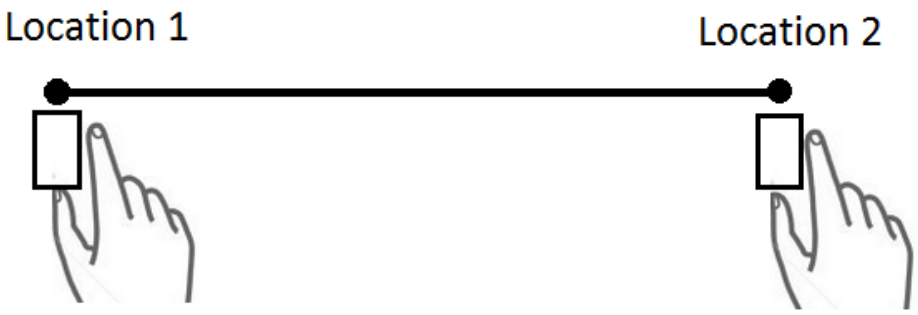


Figure 3.15: The diagram of hand movement experiment for the multi-rate Kalman filter sensor fusion. It illustrates that the user moves the hand from location 1 to location 2. The mobile device equipped with inertial sensors is held in the hand during the whole process.

3.5.2 Experiment Design

For comparison, we designed a single rate Kalman filter, which only used the Kinect position measurements. The Kinect position uncertainty is estimated to be $\sigma_p = 8$mm. For the single rate Kalman filter and the multi-rate Kalman filter, the position measurement noise variance $R = 8^2$.

For estimating the process noise covariance matrix, we applied the continuous Wiener process acceleration model. The power spectral density of the white noise process is estimated

to be $q = 2000$ based on the experiment training data. For the single rate Kalman filter, the process covariance matrix Q_s was computed according to equation (3.3) by setting $\Delta t' = \frac{1}{30}$s. For the multi-rate Kalman filter ($\Delta t = \frac{1}{90}$s), we denoted the process noise covariance matrix as Q_m. When the target (hand) moved faster and the motion changed rapidly, the process noise was larger (Welch & Bishop, 1995). We set a threshold for the hand velocity, i.e. $v_{threshold} = 150$mm/s. When the velocity of the hand was larger than the threshold, we changed the process noise covariance matrix accordingly. During the periods when the hand was motionless, the measurement noise was smaller in comparison with that during the moving period. For the motionless period, the measured noise variance for the accelerations is $R_{a1} = 10^2$ and $R_{a2} = 100^2$ for the moving period.

3.5.3 Position Estimation

We fused the Kinect low-sampling-rate position and the high-sampling-rate acceleration in the multi-rate sensors-based Kalman filter. For comparison, we designed a single rate Kalman filter and also got the state estimation results. The single rate KF estimated the state with only position measurements sensed by the Kinect. The position comparison results are shown in Figure 3.16. The Kinect position measurements, the filtered positions from the single rate KF and the sensor fusion result with the multi-rate KF are all illustrated. For better visualisation, we show the magnified plots of Figure 3.16 in Figure 3.17 and Figure 3.18. We can see that there is an overshoot problem and the lag with position estimation with the single rate KF. The sensor fusion with the multi-rate KF helps increase the accuracy of mean position estimation and reduce the uncertainty of this estimation, and also helps reduce lag in response and the overshooting. The average uncertainty of position estimated with the single rate Kalman filter was (SD) 3.18mm. For the multi-rate KF, it was 2.08mm. The standard deviation was reduced by 34.6%.

Therefore, the multi-rate Kalman filter is superior to the single rate Kalman filter in position estimation. Fusing the Kinect sensor and inertial sensors with the multi-rate Kalman filter helps increase the stability of the position and reduce lag in response, giving a smoother and more responsive experience when the user interacts with a location-aware sensing application in the room.

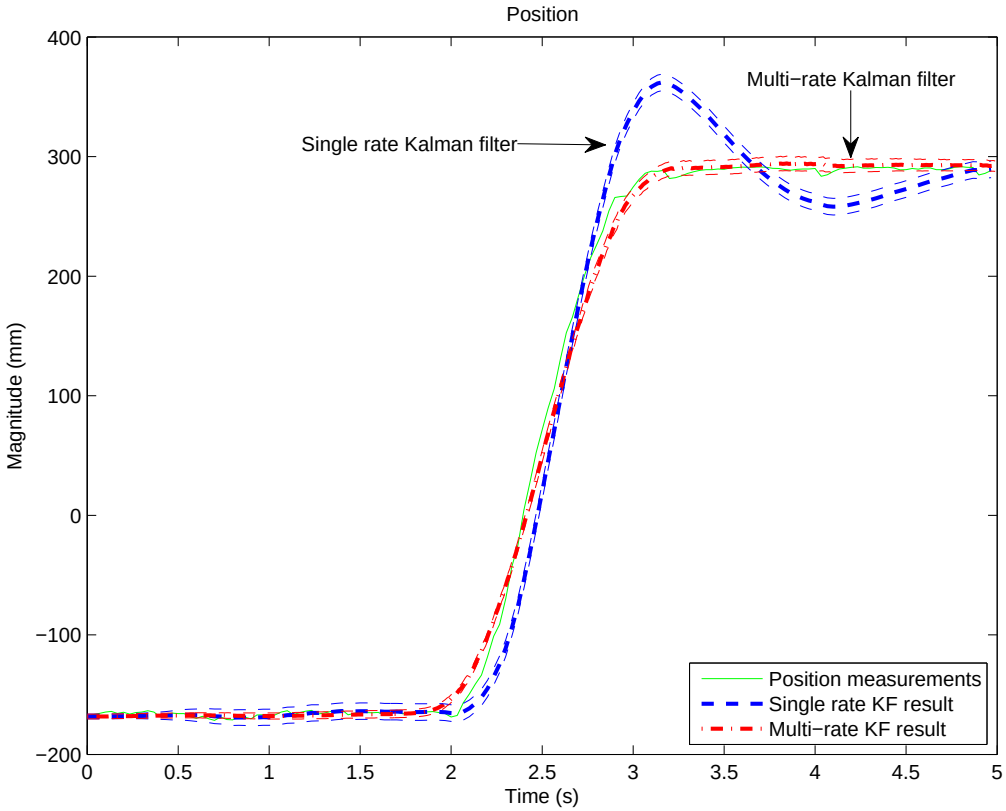


Figure 3.16: Position estimation results with the multi-rate Kalman filter and the single rate Kalman filter are shown and illustrated. Plots show the mean estimation and mean $\pm 2\sigma$ (We get the uncertainty of the mean estimation through equation (2.7). For the single rate KF, σ will be constant after the convergence. For the multi-rate KF, σ changes dynamically.). The position measurements are also shown. It can be seen that there is an overshoot problem with the single rate KF and the sensor fusion with the multi-rate KF helps stabilise the hand positions and reduce the uncertainty of position estimation.

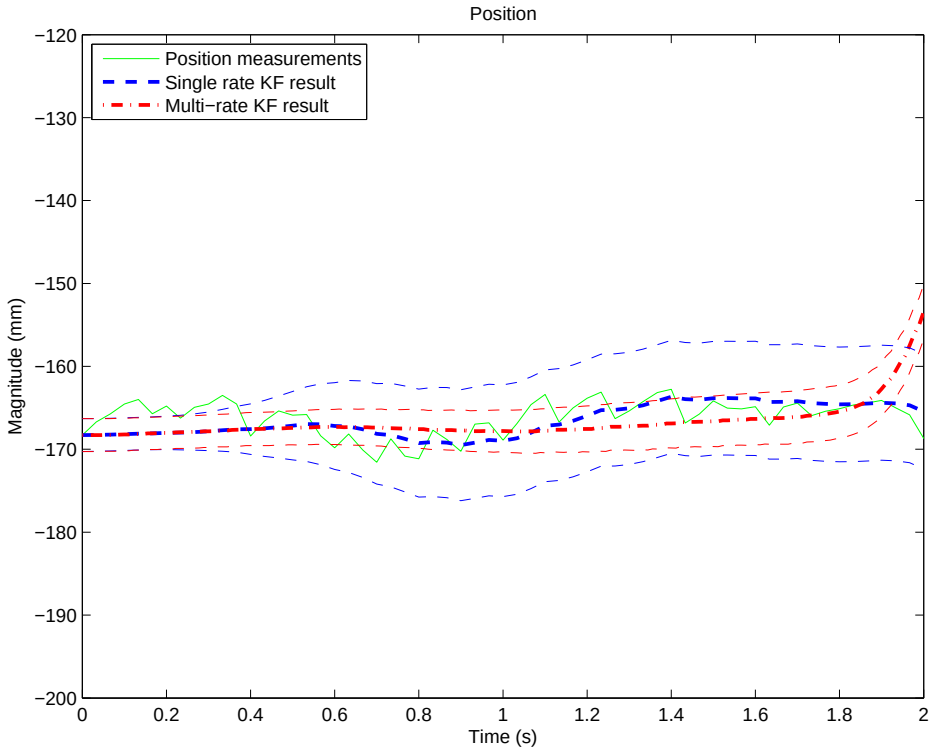


Figure 3.17: The magnified plot of position comparison in Figure 3.16. Plots show the position estimation results $(0 - 2\text{s})$. We can see that the uncertainty of position estimation with the single rate KF is larger in comparison with that with the multi-rate KF.

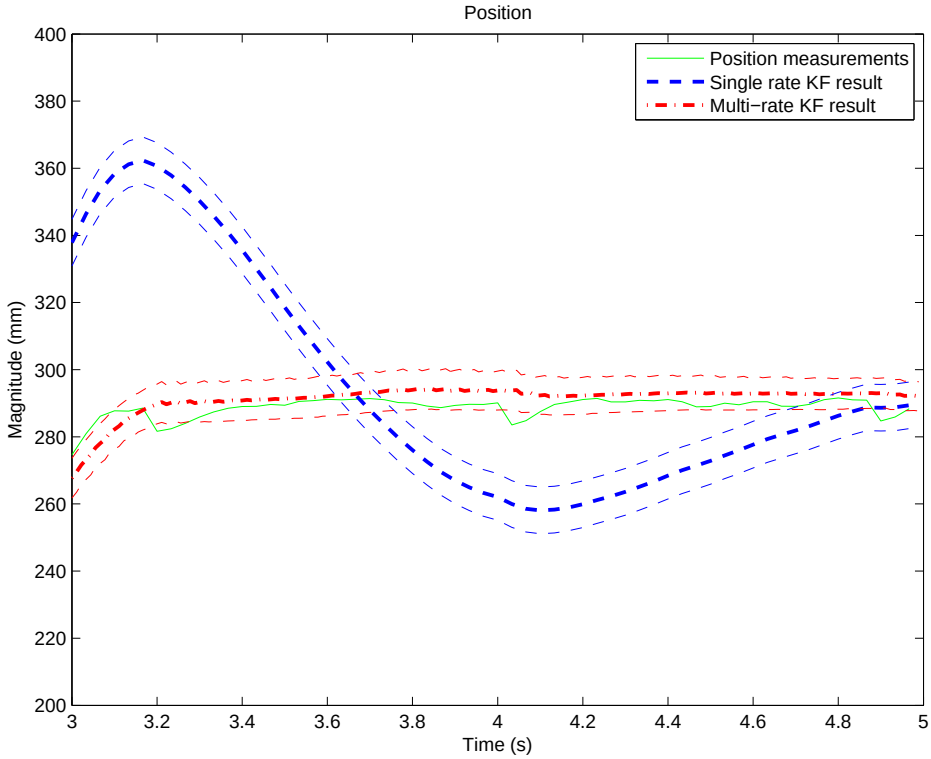


Figure 3.18: The magnified plot of position comparison in Figure 3.16. Plots show the position estimation results $(3 - 5\text{s})$.

3.5.4 Velocity Estimation

In both Kalman filters, the state vector is composed of position, velocity and acceleration. In addition to the position estimation, we also investigated the performance of the proposed sensor fusion method for estimating the velocity.

In Figure 3.19, we show the first derivatives of the Kinect positions, the velocity estimated with a single rate Kalman filter and the velocity estimated with the designed multi-rate Kalman filter. With only inertial sensors, we cannot get a reliable estimation of velocity as drift happens. With only a position sensing device, e.g. the Kinect, the velocity estimation results are not robust. The first derivatives of the Kinect positions and the velocity estimated with a single rate Kalman filter are illustrated clearly in Figure 3.19. We can see that the sensor fusion with the multi-rate Kalman filter helps estimate the mean velocity more accurately and the uncertainty is also reduced. The average uncertainty of velocity estimated with the single rate Kalman filter was (SD) 12.26mm/s. For the multi-rate Kalman filter, it was 4.52mm/s. The standard deviation was reduced by 63.1%.

We conclude that the sensor fusion with the proposed multi-rate Kalman filter helps improve the accuracy of the velocity estimation. This is a very useful by-product of position estimation. It is beneficial for interaction applications that require robust velocity sensing when only position sensing and inertial sensors are available.

3.5.5 Acceleration Estimation

In addition to the position and velocity estimation, we also get the acceleration estimation with both Kalman filters. In Figure 3.20, the second derivatives of the Kinect positions, the accelerations sensed by the inertial sensors, the acceleration estimated with a single rate Kalman filter and the acceleration estimated with the proposed multi-rate Kalman filter are all shown and illustrated clearly. We can see that the sensor fusion with the proposed method improves the accuracy of the mean acceleration estimation and reduce the uncertainty of this estimation. The average uncertainty of acceleration estimated with the single rate Kalman filter was (SD) 33.60mm/s^2. For the multi-rate Kalman filter, it was 29.25mm/s^2. The standard deviation was reduced by 13.0%.

Thus, the sensor fusion helps improve the acceleration estimation by fusing the low-sampling-rate position measurements sensed by the Kinect and the high-sampling-rate accelerations measured by the inertial sensors embedded in the mobile device.

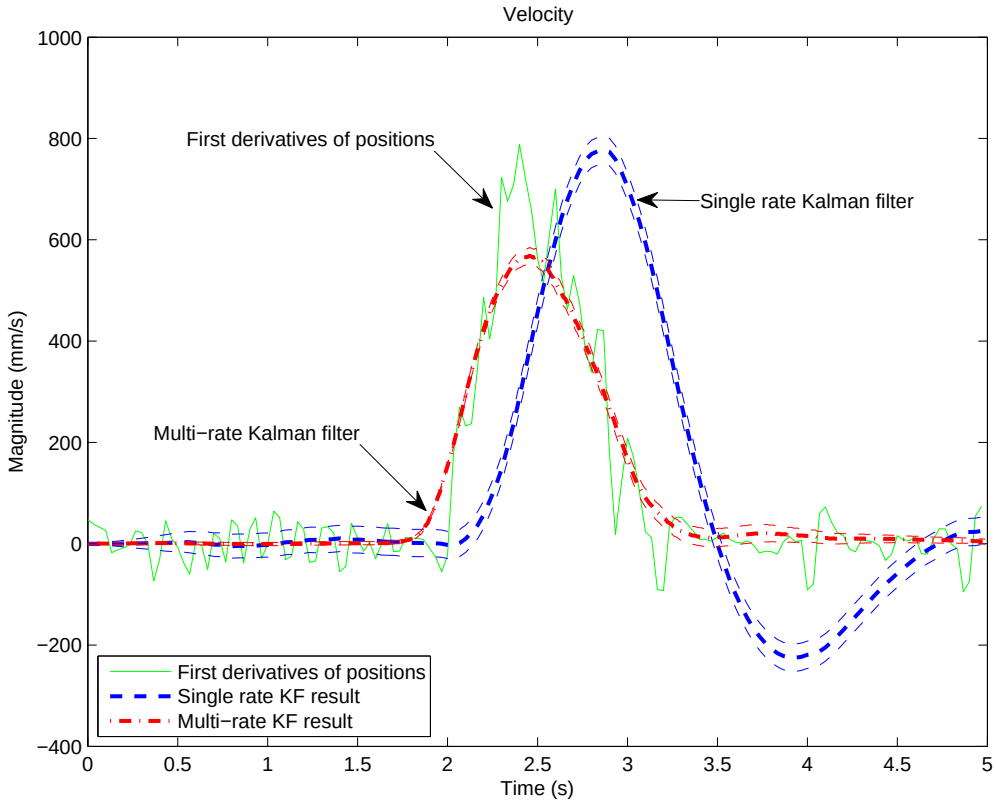


Figure 3.19: Velocity estimation with the multi-rate Kalman filter and the single rate Kalman filter. Plots show the mean estimation and mean $\pm 2\sigma$. The first derivatives of Kinect positions are also shown. It can be seen clearly that the sensor fusion with the multi-rate KF helps improve the accuracy of velocity estimation. The mean velocity is more accurate and the uncertainty is reduced.

3.5.6 Conclusion

In this section, an experiment was conducted to test the multi-rate sensors-based Kalman filter method. We fused the positions sensed by the Kinect and the accelerations measured by the inertial sensors with the proposed method. Experimental results show that the sensor fusion with the proposed method helps improve the accuracy of the system state estimation, including the position, the velocity and the acceleration. Firstly, we investigated the position estimation. We compared the performances of the proposed multi-rate sensors-based Kalman filter and the single rate Kalman filter. Experimental results show that there is lag and also an overshoot problem with the single rate Kalman filter. The proposed method helps increase the stability of the hand position, reduce the uncertainty of position estimation, and reduce overshooting and lag in response. Secondly, we studied the velocity estimation and compared the results of the two filters. The proposed method is superior to the single rate Kalman filter in improving the accuracy of velocity estimation. Finally, we show the acceleration estimation results. The proposed method is also superior to the single rate Kalman filter in improving the accuracy of acceleration estimation.

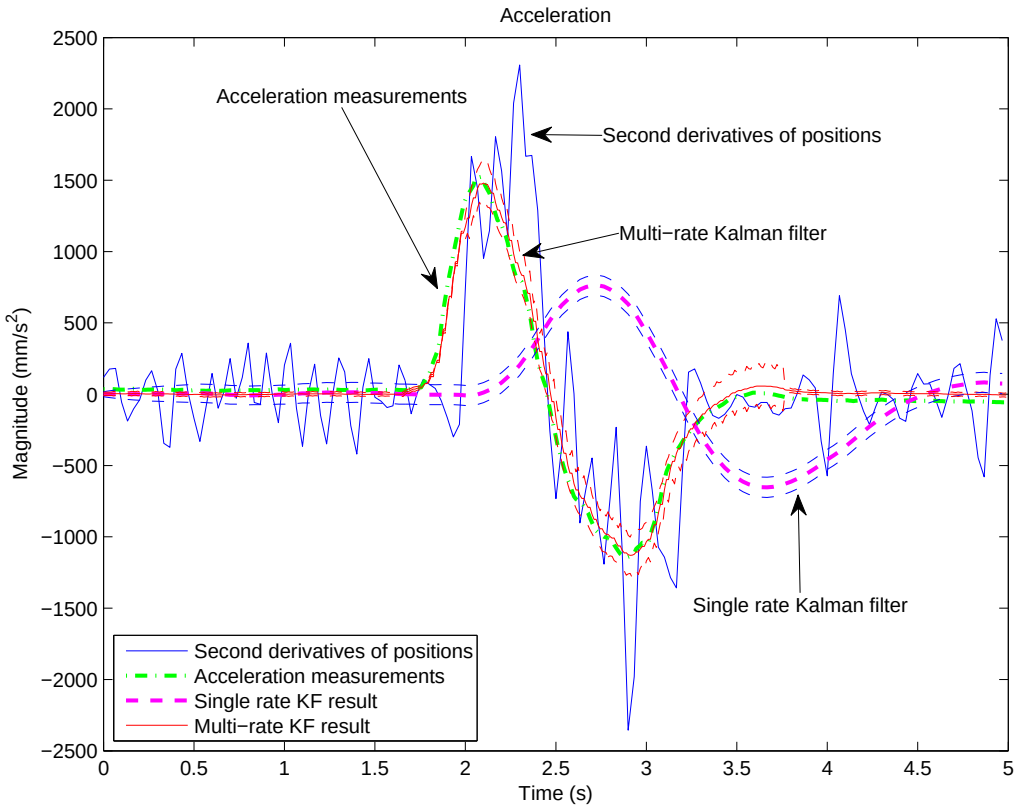


Figure 3.20: Acceleration estimation with the multi-rate Kalman filter and the single rate Kalman filter. Plots show the mean estimation and mean $\pm 2\sigma$. The second derivatives of the Kinect position measurements and the accelerations sensed by the inertial sensors are also shown. It can be seen clearly that the sensor fusion with the multi-rate KF helps improve the accuracy of acceleration estimation. The mean acceleration is more accurate and the uncertainty is also reduced.

3.6 Conclusions

In this chapter, we have presented a method of fusing the Kinect sensor and inertial sensors equipped in a mobile device for improving the system state estimation, i.e. the position, the velocity and the acceleration of the skeleton joints. As an example, we analysed the hand motion. The approach we adopt is to fuse the low-sampling-rate, noisy positions sensed by the Kinect and the high-sampling-rate accelerations measured by the inertial sensors with a multi-rate sensors-based Kalman filter. Firstly, we fused the inertial sensor data for estimating the skeleton joint acceleration in the Kinect coordinate system. We proposed a coordinate system transformation method for converting the accelerations from the body frame to the Kinect frame. Secondly, we fused the Kinect sensor and the inertial sensors with the proposed method.

Two experiments were conducted to test the proposed acceleration estimation method and the sensor fusion method respectively.

In the first experiment, we tested the feasibility of estimating the hand acceleration with the mobile inertial sensors in the Kinect coordinate system. Experimental results show that the proposed method can be successfully applied for estimating the hand acceleration during the typical everyday hand movements. The accelerations estimated with the Kinect sensor and those measured by the inertial sensors are comparable and match well.

In the second experiment, we fused the positions sensed by the Kinect and the accelerations measured by the inertial sensors with the proposed multi-rate sensors-based Kalman filter method. Experimental results show that the sensor fusion with the proposed method can help improve the accuracy of the system state estimation, including the position, the velocity and the acceleration. As an example, we used the Kinect for skeleton tracking and analysed the hand motion. The sensor fusion helps the user better interact in the information space embedded in a physical environment by stabilising the hand position. This is of great benefit for the interaction systems that aim to fuse the external position sensing and the built-in inertial sensors in a mobile device for indoor augmented reality (AR) and other location-aware sensing applications.

Chapter 4

The Sensor Fusion System

This chapter presents a novel Gaussian Process (GP) prior model-based sensor fusion system composed of a Kinect sensor and mobile inertial sensors, where the model provides a principled mechanism for incorporating the low-sampling-rate position measurements and the high-sampling-rate derivatives in multi-rate sensor fusion which takes account of the uncertainty of each sensor type and the latency of the Kinect system. In Chapter 3, we explored the complementary properties of these sensors and designed a multi-rate sensors-based Kalman filter for fusing the synchronized sensor data. In this chapter, we propose that (1) the GP prior model can be applied for calculating the joint log-likelihood of position and acceleration. The user matching and identification application will be presented in Chapter 5; (2) the GP prior model-based sensor fusion approach can be used to deal with position uncertainty and lag problem in the conventional Kinect system. A user study in a spatially aware display application will be presented in Chapter 6.

We give an introduction of the sensor fusion system first. We present that the Kinect can be augmented with mobile devices, including SK7 and a mobile phone. Following this, we give a detailed description on the novel and improved GP prior model and apply it to fuse the Kinect sensor and mobile inertial sensors. We also present an alternative view of the sensor fusion and discuss the relationship between the GP method and the multi-rate Kalman filter. Finally, we conduct an experiment to test the performance of the proposed sensor fusion system. The experimental results show that fusing the Kinect sensor and the built-in inertial sensors in a mobile device with the proposed GP prior model-based sensor fusion approach helps improve the accuracy of position estimation and reduce the lag.

4.1 Introduction

The Microsoft Kinect sensor can be augmented with the built-in inertial sensors in a mobile device. We introduce the complementary properties of the Kinect sensor and mobile inertial sensors in section 1.2.2. In this chapter, we explore the use of a mobile device to augment a Kinect system. We propose a novel and improved Gaussian Process prior model for fusing the low-sampling-rate Kinect position measurements and the higher frequency accelerations sensed by mobile inertial sensors. The inertial sensor fusion gives us the skeleton joint acceleration, thus the mobile device can be used as an acceleration sensor. In the meantime, the Kinect is a skeleton joint position sensor. The fusion of the Kinect sensor and mobile inertial sensors using appropriate probabilistic approaches can improve the position tracking, thus significantly improve the usability of a human-computer interaction system. For human motion tracking with the Kinect, the noisy position measurement is a common problem (Casiez et al., 2012). Advanced sensor fusion techniques could improve the usability by providing more accurate position data, but external states cannot be known with absolute accuracy and uncertainty always persists (Williamson, 2006). Besides sensor sources, hand tremor and human motor variability will also affect the sensor measurements and induce uncertainty (Strachan & Murray-Smith, 2009).

To address this problem, we need to apply filtering or sensor fusion techniques. However, filtering will introduce lags, which reduce the system responsiveness (Casiez et al., 2012). Besides, to minimize both jitter and lag with a filter in the Kinect system is challenging. However, with additional, complementary sensors, e.g. the inertial sensors, we can improve the position estimation, reducing the jitter and the lag of the system. The ubiquitous inertial sensors can be used as aiding sensors for position tracking.

In order to fuse the Kinect sensor and mobile inertial sensors for state estimation, we need dynamical system modelling techniques. Bayesian filtering is a general framework for recursively estimating the state of a dynamic system (Ko & Fox, 2009). The basic idea of Bayesian filtering is that we estimate the state of the system with probabilistic models including the state transition model and the observation model. For instance, the Kalman filter and its variants (EKF and UKF) have been widely used for filtering and sensor fusion (Welch & Bishop, 1995, 1997).

One of the drawbacks of applying Gaussian Processes for dynamical system modelling is that it is computationally expensive. The major computation in a GP is the inversion of the covariance matrix. However, our model is an autoregressive model so the covariance matrix is a fixed matrix for a constant sampling rate (90Hz), making it computationally efficient, as the inversion only needs to be done once.

In this chapter, we propose a novel and improved GP prior model-based sensor fusion ap-

proach to dealing with the position uncertainty and lags problem in an interaction system composed of external position sensing and inertial sensors. We propose a variation of a Gaussian Process prior model (Rasmussen & Williams, 2005) that incorporates the low-sampling-rate measurements and the high-sampling-rate derivatives in multi-rate sensor fusion. It takes the different sampling rates and the different noise characteristics of the Kinect sensor and the inertial sensors into account. The relationship between the novel GP prior model and the Kalman filter is described. Based on the GP model, the system can infer the position (and its uncertainty) more accurately and with less delay than other filters.

4.1.1 Hand Motion Tracking with Kinect Sensor and Inertial Sensors

The hand is one of the skeleton joints sensed by the Kinect sensor. As an example, we analyse the hand motion tracking when the mobile device is held in the hand in this Kinect-augmented system.

The hand can be used as an input device and provides natural human-computer interaction (Erol et al., 2007). Hand motion tracking has many applications, such as virtual reality or augmented reality with a handheld display. Glove-based devices have been used for measuring the location and shape of a user's hand (Sato et al., 2001). However, this may inhibit the user's movement. The Kinect benefits HCI by providing a simple way for hand tracking. However, position uncertainty is a common problem. Besides, due to occlusion or very fast movement, the skeleton tracking may fail. These are the challenges for HCI applications that use the Kinect for hand tracking. Besides, the hand pose estimation is not currently available in the standard Windows SDK API and the OpenNI driver.

Nowadays, mobile phones equipped with inertial sensors are becoming ubiquitous. When people hold the phones in their hands, the phone orientation indicates the hand pose. Meanwhile, the Kinect sensor is low-cost, portable and unobtrusive in a room. By combining a Kinect sensor and the built-in inertial sensors in a mobile device, we can build a more robust system, which can track not only the hand position, but also the hand pose. Compared with "data gloves", this system does not limit a user's movement. Thus, it provides a more natural way of human-computer interaction.

In the circumstances mentioned above, the Kinect and the phone are used separately. The former provides the hand position while the latter indicates the hand pose and fine movement. Although the overall system has more functions than each single system, these sensors are not used in a complementary way. We discussed the complementary properties of these sensors in section 1.2.2. In this chapter, we will describe the approach to this sensor fusion problem.

We will discuss the transformations of Gaussian Process priors in section 4.3.2. With the noisy, low-sampling-rate Kinect position measurements, we can estimate the derivatives including the velocity and the acceleration. In this chapter, we will present the sensor fusion approach using the transformed GP priors.

4.1.2 Challenges

The accuracy of position tracking is critical in the location-aware systems (Khoury & Kamat, 2009). In robotics, a primary challenge is to deal with uncertainty, which arises for many reasons, including the limitations of the model, the limited perceptual capabilities of the sensors and the noisy measurements, and the approximate nature of the algorithm. Probabilistic approaches, among which Kalman filter is a popular method are described in (Thrun, 2002).

The skeleton tracking with a Kinect has two major problems, the position uncertainty and the latency, as introduced in section 1.2.1. The Kinect position measurement is subject to significant uncertainty (Casiez et al., 2012). Besides, there is latency in the Kinect system (Azimi, 2012). The filtering technique can be applied for noise reduction. However, this will induce the lag. With only a Kinect sensor, to minimize both jitter and lag is challenging as we need to find a trade-off between increasing the position accuracy and reducing the lag for the interactive system. To reduce both the position uncertainty and the latency, we need additional sensors.

To address these challenges, we thought about fusing the Kinect sensor and mobile inertial sensors by exploring their complementary properties with sensor fusion method. Sensor fusion combines different sensor sources and provides significant advantages over a single sensor source (Hall & Llinas, 1997). The Kinect sensor measures the skeleton joint position while the inertial sensors in a handheld device can be used for estimating the acceleration. Thus, these sensors have complementary properties. We propose a novel Gaussian Process prior model-based sensor fusion approach.

4.1.3 Applications

To investigate the performance of the GP prior model-based sensor fusion system, we apply this model for fusing the low-sampling-rate position and the high-sampling-rate acceleration, and use the sensor fusion system for two applications, which will be presented in Chapter 5 and Chapter 6, respectively.

One benefit of the GPs sensor fusion is that GPs can provide the joint log-likelihood given different combinations of position signals and acceleration signals. This can be used to associate a particular user with a mobile device. The user matching application will be

described in Chapter 5. The fusion of the Kinect and the mobile devices equipped with inertial sensors with the proposed GP prior model is beneficial for interaction systems that require user matching and identification in context-aware applications.

The second application is based on the GPs position prediction. Based on the GP prior model, the system can infer the position and higher derivatives (and its uncertainty) more accurately and with less delay than other filters. The performance of the system was tested in an experiment. A spatially aware display application was developed for a user study. This will be presented in Chapter 6. The proposed GP prior model-based sensor fusion approach helps increase the stability of the position and reduce the lag of the system. It is of great benefit for combining external position sensing device and inertial sensors embedded in a mobile device for augmented reality (AR) and other location-aware sensing applications.

4.2 System Overview

The equipment includes the Microsoft Kinect, a SHAKE SK7[1] (Williamson et al., 2007) and Nokia N9. The different coordinate systems involved in this sensor fusion have been introduced in section 3.3.2.

The sensor fusion system architecture is shown in Figure 4.1. It illustrates that the Kinect is applied for skeleton joint position tracking. In the meantime, the mobile inertial sensors measure the joint acceleration. The joint position is fused with the acceleration with the proposed GP prior model. The benefits of fusing the Kinect sensor and mobile inertial sensors with the proposed GP prior model have been introduced in section 1.2.2. The GP sensor fusion results include (1) the log-likelihood given the position and the acceleration, and (2) the GP predictive position. Thus, the sensor fusion system can be used for two applications, i.e. (1) user matching and identification in Chapter 5; (2) a spatially aware display application in Chapter 6.

4.2.1 Augmenting the Kinect System with SK7

The Kinect system can be augmented with SHAKE SK7, allowing us to calculate the orientation of the device with respect to the Kinect. In the Kinect-augmented system, the human motion is observed by multiple sensors. The Kinect senses the skeleton joint positions and sends them to the PC. Meanwhile, the SK7 communicates with the PC via Bluetooth. By matching the motion sensed by the Kinect with the motion sensed by the mobile device, we can find the matching relationship between the multiple skeletons and the mobile devices in a room. This is the first application, i.e. user matching.

[1] http://code.google.com/p/shake-drivers/

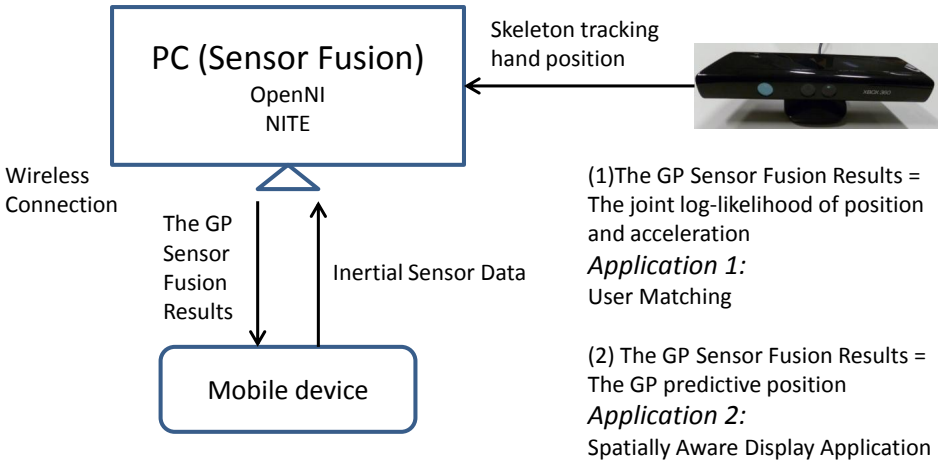


Figure 4.1: System architecture. A wireless connection is set to connect the PC with the mobile device. The OpenNI and NITE middleware are used. The Kinect senses the hand position and sends it to the PC. The inertial sensor data sensed by the mobile device is also sent to the PC. Our novel GP sensor fusion model is applied for fusing the position and the acceleration. Depending on the GP sensor fusion results, the system can be used for two applications. (1) The GP log-likelihood of position and acceleration can be used for user matching and identification. (2) The GP helps improve the system state estimation in a spatially aware display application. This facilitates the user's targeting by stabilising the hand position and reducing the lag of the system.

Applying the novel GP model for sensor fusion consists of two steps. Firstly, fuse the inertial sensor data for estimating the hand acceleration. This has been discussed in section 3.4. Secondly, calculate the joint log-likelihood of the Kinect position and the higher frequency acceleration measured by SK7 with the novel and improved GP prior model. A detailed description will be presented in Chapter 5.

The user matching and identification can be achieved implicitly by augmenting the Kinect system with SK7. The implicit interaction system does not require a user's explicit intention, but can take a user's movement as input and then output useful information. Exploring the use of personal devices for user matching and identification in a Kinect-augmented room is feasible in people's everyday lives. It is beneficial for any interaction systems that aim to identify individual users through matching the motion sensed by the positioning device and the motion sensed by mobile inertial sensors. This enables the system to provide a personalized service to individual users in a proxemic interaction system.

4.2.2 Augmenting the Kinect System with a Mobile Phone

We can augment the Kinect with a mobile phone equipped with inertial sensors. The phone screen can be used as a handheld display for providing visual feedback information that we

get from the Kinect-augmented system.

For mobile augmented reality (AR) and other location-aware sensing applications, e.g. a spatially aware display, the user needs to use explicit hand motion-based interaction to access the digital information embedded in the room. The screen display and feedback information can facilitate the users' interaction when they browse or explore the virtual information, e.g. a digital bookshelf library application, embedded in a physical environment. This motivates us to develop a phone application with screen display.

The Kinect system can be augmented with a mobile device (Nokia N9). The Nokia N9 is a phone with 3.9 inches display (480 pixels $\times$ 854 pixels or 48 mm $\times$ 86 mm). It is equipped with $3-$axis accelerometer, which can be used to estimate the hand acceleration. The fusion of the Kinect and the phone accelerometer is a Kinect-augmented system, where the position tracking is improved as the high-sampling-rate acceleration can augment the low-sampling-rate position measurements. This sensor fusion system helps increase the stability of the position and reduce the lag of the system, thus enables the user to better control the mobile device for enhanced interaction, giving a better performance and experience. This allows us to build a spatially aware display application that benefits from the GP sensor fusion. The GP predictive positions enable the user to better control the device in order to browse the information embedded in a physical environment. An example application is shown in Figure 6.2.

We designed a 2D (the vertical XY plane) version of a spatially aware display application in Chapter 6. We aim to test whether the GP prior model-based sensor fusion system improves user performance by designing a trajectory-based target acquisition task in this application. The acceleration sensed by the phone is sent to the PC via a Wireless Local Area Network (WLAN), which is used for User Datagram Protocol (UDP) connection. The GP sensor fusion results are used to locate the phone and update the screen display.

Accurate positioning and feedback are critical to this application. The use of filtering technique to deal with Kinect position jitter will introduce lags. Besides, the overshoot problems occur when we apply a position-only Kalman filter for filtering the fast jitter on the hand movement data. The novel GP prior model-based sensor fusion helps improve the accuracy of position estimation, and reduce the lag. This enables the user to select the targets more accurately and quickly, improving the user's performance and bringing a better user experience.

4.3 Gaussian Process Prior Model For Fusing Kinect Sensor and Inertial Sensors

4.3.1 Problem Statement for Dynamical System Modelling

We consider the situation when the user holds a mobile device in the hand and tries to explore the digital information stored in the Kinect space in the room. The low-sampling-rate Kinect position measurements and the high-sampling-rate acceleration measured by mobile inertial sensors are available. The problem is that the Kinect position measurements are noisy and delayed. The GP model needs to take into account the different sampling rates and different noise characteristics of these sensors, and the delay.

The human and the environment can be thought of as a combined dynamical system, in which the human motion is observable with multiple sensors. The skeleton data sensed by the Kinect and the hand motion data sensed by mobile inertial sensors are shared via Wireless LAN.

This is a closed-loop system with two subsystems, as illustrated in Figure 4.2. The human is subsystem 1 while the computing device system, including the mobile phone, the multiple sensors and the PC used for sensor fusion, can be treated as subsystem 2. The human motion in subsystem 1 is the input to subsystem 2. The state (the joint position) estimation from sensor fusion is sent to the mobile phone for updating the screen display, which provides visual feedback to the human. The visual feedback, which is based on the improved position estimation, enables the user to better perform a continuous control interaction. It facilitates the action of the user, i.e. the targeting. Thus, our GP prior model-based sensor fusion system enables the user to better control the system.

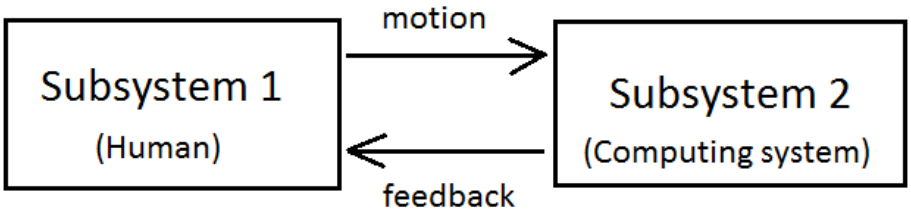


Figure 4.2: Illustration of a closed-loop system with two subsystems including (1) subsystem 1: the human (2) subsystem 2: the computing system consists of the mobile phone, the multiple sensors and the PC.

In subsystem 2, the phone can be seen as a moving target when the hand is moving. The user controls the moving of the phone. We can treat the phone as a flying machine, the input of which is the force of the hand. The operator of this machine is the user. The motion of the phone is observed by multiple sensors. The trajectory is sensed by the Kinect

sensor. Meanwhile, the orientation and the acceleration of the phone are observed by the built-in inertial sensors. The phone position and acceleration measurements are available. The system is a second-order system since the dynamics depend on the second derivatives of position.

The subsystem 2 is observable as we can determine the state of the system through the position observations and the acceleration measurements. We can use a sensor fusion technique to fuse the noisy, low-sampling-rate position and the higher frequency acceleration in order to determine the more accurate position of the target (phone).

Here we approximate and model the whole system as a second-order dynamical system. This subsystem 2 is a time-delay system as the position is sensed by the Kinect, which has latency. The acceleration is sensed by the inertial sensors at a much higher sampling rate. We treat the acceleration as a non-delayed measurement. Our goal is to model this dynamical system with the GP prior method. The phone (hand) trajectory is defined by the movement of the user's muscles which drive nonlinear trajectories of the rigid body. The system we are modelling is a nonlinear dynamical system $g(x)$ with known inputs x and observed outputs y. At each time instant i, we get a measurement y_i, which is a function of the latent state x_i.

$$y_i = g(x_i) + \varepsilon_i, \tag{4.1}$$

where ε_i denotes Gaussian system noise.

In order to estimate the system state by fusing all the available observations including the positions and the accelerations, we need to illustrate the data availability in the sensor fusion system.

Data Availability in the Sensor Fusion System

Now we illustrate the data availability with Figure 4.3. In order to illustrate the availability of sensor measurements at different time instants, we need to take account of the time delay (0.1s) of the Kinect system (Azimi, 2012; Livingston et al., 2012). When we receive the acceleration measurements sensed by the inertial sensors, we also get the delayed position measurements from the Kinect. In order to fuse these sensor data, we need to build a model that takes account of the time delay and the different noise characteristics, and incorporates the different sensor measurements. Now we illustrate the availability of sensor data at different time instants and how we make use of the delayed position measurements.

In Figure 4.3, we show the timing information and the delayed observations at $t = \frac{15}{90}$s. The first row represents the timing information and the second row represents the acceleration measurements from the inertial sensors. In the third row, considering the effect of latency, the corresponding Kinect outputs are denoted as $p_i^d, i = 1, 2, 3, 4$. Due to the 0.1s latency,

these outputs are the delayed outputs and cannot be treated as the real-time position outputs. In the fourth row, it is shown that the actual available observations at $t = \frac{13}{90}$s include 13 acceleration measurements and 2 position measurements, which are the noisy version of the system state (position) at $t = \frac{1}{90}$s and $t = \frac{4}{90}$s, respectively. We denote them as p_3 and p_4. The corresponding Kinect outputs become p_3^d and p_4^d, which are acquired at $t = \frac{10}{90}$s and $t = \frac{13}{90}$s, respectively.

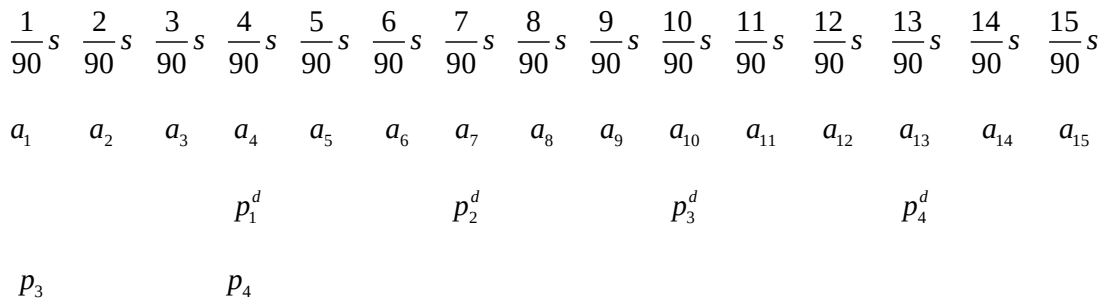


Figure 4.3: Illustration of data availability: (1) The first row represents the time instants (90Hz) (2) The second row represents the non-delayed acceleration measurements. (3) The third row represents the Kinect position measurements. They are the delayed noisy version of the system state (position). (4) Due to the 0.1s latency, we assume that at $t = \frac{13}{90}$s, the available position measurements include p_1^d, p_2^d, p_3^d and p_4^d. p_3^d and p_4^d represent the delayed noisy version of the system state (position) at $t = \frac{1}{90}$s and $t = \frac{4}{90}$s. Considering the latency, the corresponding Kinect position outputs are denoted as p_3^d and p_4^d.

Autoregressive GP Model

Gaussian Process regression is a linear smoother (Hastie & Tibshirani, 1990; Rasmussen & Williams, 2005). Our proposed model is an autoregressive model, which acts like a moving "window". The AutoRegressive Gaussian Process (ARGP) was applied for time series modelling in (Girard et al., 2003a; Girard, 2004; Turner, 2012). In an ARGP of order L, the past L values $\mathrm{Y}_{(L)}$ are taken as the GP input while the output is

$$y_t = f\left(\mathrm{Y}_{(L)}\right) + \varepsilon_t, \tag{4.2}$$

where the GP function $f \sim GP\left(0, k\right)$ and the white noise $\varepsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$.

Here we put the ARGP model in a sensor fusion framework. The sensor observations are the inputs of the ARGP model. The problem is that the sensor observations include the delayed low-sampling-rate positions sensed by the Kinect and the high-sampling-rate accelerations measured by the inertial sensors. We want to build a GP prior model that incorporates these observations and takes the different characteristics of these sensors into account.

We define the state of interest x_t as

$$x_t = f\left(p_{(L)}, a_{(l_a)}\right), \tag{4.3}$$

where x_t represents the GP predictive positions. The last L position measurements sensed by the Kinect are denoted as $p_{(L)}$, whereas $a_{(l_a)}$ are the last l_a acceleration measurements sensed by the inertial sensors, and $l_a = 3L + N_0 - 2$. The past L Kinect positions are the low-sampling-rate measurements in the assumed high-sampling-rate position space. The acceleration measured by the inertial sensors corresponds with the assumed high-sampling-rate position space. The data availability has been illustrated with Figure 4.3 in section 4.3.1.

Considering the different sampling rates of these sensors, we have more acceleration measurements than position measurements. N_0 is a number that represents the latency between the Kinect position measurements and the acceleration measurements.

$$N_0 = \frac{dT}{\Delta t} = dT \cdot f_0, \tag{4.4}$$

where dT denotes the time delay, i.e. 0.1s (Azimi, 2012). f_0 denotes the sampling rate of the inertial sensors, i.e. 90Hz. Thus, $N_0 = 9$.

As defined in equation (4.3), every time the "window" takes the most recent L position measurements and the most recent l_a acceleration measurements. In this way, we propose an overall GP prior model for calculating the joint distribution of position and acceleration. During the time period when the position measurements are unavailable, i.e. the most recent 0.1s latency, the GPs make position prediction based on the most recent L position measurements and the most recent l_a acceleration measurements.

Now we have the state equation of the dynamical system, as defined in equation (4.3). Following this, we propose a novel Gaussian Process prior model for the dynamical system modelling. In this way, we can make predictions based on the previous and current acceleration measurements, and the delayed position measurements.

In our work, the human motion is relatively continuous and smooth in the trajectory-based target acquisition task. Here the covariance function chosen is a general smoother, the parameters of which are tuned to typical human motion. The parameters for the model are learnt from the training data using the maximum likelihood method. As GP regression is a linear smoother (Hastie & Tibshirani, 1990), the prediction is a linear combination of the training targets. The GP prediction was discussed in section 2.4.1. We denote the noisy measurements as Y. The predictive mean of a new input vector x^* is denoted as $\mu(x^*)$. In order to illustrate this, we put equation (2.15) here.

$$\mu(x^*) = C(x^*, X)\left[C(X, X) + \sigma_n^2 I\right]^{-1} Y$$

If we denote $C_Y = C(x^*, X)[C(X, X) + \sigma_n^2 I]^{-1}$, we get $\mu(x^*) = C_Y Y$. We can see that the predictive mean of this new input vector x^* is a weighted average of the noisy measurements Y. Therefore, the GP regression is a linear smoother.

For sensor fusion with the GP prior model, the targets include the L positions and the l_a accelerations. If we can place an appropriate prior on the combination function space, we can make position predictions based on the non-delayed accelerations during the 0.1s. In order to find the joint distribution of the low-sampling-rate position, denoted as P_{low}, and the high-sampling-rate acceleration, denoted as Acc_{high}, we apply the GP prior method and calculate an overall covariance matrix C_{all}, so

$$\begin{bmatrix} P_{low} \\ Acc_{high} \end{bmatrix} \sim \mathcal{N}(0, C_{all}). \tag{4.5}$$

So the following work is to apply GPs in a sensor fusion manner and find this joint distribution of the low-sampling-rate position and the high-sampling-rate acceleration with the GP prior method. Firstly, we discuss the GP prior prediction. Following this, we present the transformed GP priors and propose the novel and improved GP prior model for multi-rate sensor fusion, and give a detailed description on how to apply this model for fusing the Kinect sensor and inertial sensors.

4.3.2 Transformations of GP Priors and Multi-rate Sensor Fusion

Transformations of Gaussian Process Priors

Gaussian Process priors are increasingly used as flexible nonparametric models in a range of application areas (Rasmussen & Williams, 2005), including sensor fusion (Murray-Smith & Pearlmutter, 2005) and human motion analysis (Wang et al., 2008).

Given a set of N training data-points $\{x_i, y_i, i = 1, \cdots, N\}$, where $X = [x_1, \cdots, x_N]^T$ is a D-dimensional vector of inputs ($D = 1$ in our work), and $Y = [y_1, \cdots, y_N]^T$ is a vector of output data. When observations include measurements from different sensors, we can use Gaussian Process priors (Murray-Smith & Pearlmutter, 2005), which proved useful for sensor fusion.

$$Y \sim \mathcal{N}(0, \Sigma), \tag{4.6}$$

where Σ is the $N \times N$ covariance matrix, the elements of which are functions of inputs X.

With the covariance function defined in equation (2.14) and a set of training data, we can maximize the log-likelihood of the parameters-the so called hyperparameters, with the help of an iterative optimisation method.

Instead of observing Y directly, we assume that the observation m is a transformation of the latent variables y. In the continuous case,

$$output = \int_{\Omega} system \times input \, \mathrm{d}\Omega \tag{4.7}$$

$$m(t) = \int K(t,x)y(x) \, \mathrm{d}x, \tag{4.8}$$

which in discrete form is

$$m_k = \sum_{i=1}^{N} K_{ki} Y_i. \tag{4.9}$$

This is beneficial for considering the relationship between the discrete sensor measurements and the latent state of the system. Assume we observe the outputs $M = KY$ with known K. The vector M is drawn from an n-dimensional normal distribution:

$$M \sim \mathcal{N}\left(0, K\Sigma K^T + \Sigma_M\right), \tag{4.10}$$

where Σ_M is the diagonal matrix of observation variances. In this work, the application of transformations of Gaussian Process priors mainly consists of two aspects, inferring derivatives and uncertainties of derivatives of noisy position measurements, and performing sensor fusion of multiple observations in the form of multiple levels of derivatives of a measurand.

Estimating Derivatives of Noisy Measurements

The derivative of a Gaussian Process remains a Gaussian Process (Larson & Shubert, 1979; Solak et al., 2003). By selecting different kinds of transformation matrices, we can get an estimation of derivatives of the noisy measurements. As an example application of Gaussian Process priors, we can estimate the velocity and the acceleration based on the noisy position measurements according to the method introduced in (Murray-Smith & Pearlmutter, 2005).

For instance, we have noisy, low-sampling-rate position measurements Y_{low}, and the corresponding timing inputs X_1. By setting $K_1 = I_n$ in equation (4.11), we have observations $M_1 = Y_{low}$. In order to estimate the acceleration M_2 at the inputs X_2, we just need to set $K_2 = K_a$, as defined in equation (4.13). If we want to estimate the velocity, we just need to set $K_3 = K_v$, as defined in equation (4.12). According to equations (4.14) and (4.15), we can calculate the conditional mean and variance of the acceleration.

$$M_i = K_i Y \tag{4.11}$$

$$K_v = \frac{1}{\Delta t}\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \tag{4.12}$$

$$K_a = \frac{1}{\Delta t^2}\begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \end{bmatrix}, \tag{4.13}$$

where K_v and K_a are the classic first and second difference (derivative) operator, respectively.

$$\mu_{2|1} = K_2\Sigma_{12}K_1^T\left(K_1\Sigma K_1^T\right)^{-1}M_1 \tag{4.14}$$

$$\Sigma_{2|1} = \Sigma_2 - K_2\Sigma_{12}K_1^T\left(K_1\Sigma K_1^T\right)^{-1}K_1\Sigma_{21}K_2^T, \tag{4.15}$$

where Σ_{12} represents the covariance matrix between the training inputs and the test inputs while Σ_{21} represents the covariance matrix between the test inputs and the training inputs, whereas Σ denotes the covariance matrix between the training inputs and themselves. By setting $K_2 = K_v$, $K_1 = I_n$ and $M_1 = Y_{low}$, we get the mean prediction, i.e. the velocity estimation, according to equation (4.14). In the same way, we can get the acceleration estimation when we set $K_2 = K_a$. The corresponding predictive variance can be computed according to equation (4.15).

Multi-rate Sensor Fusion

The Gaussian Process prior framework can incorporate measurements and measurements of derivative information, and allows GPs to perform sensor fusion through an overall transformation matrix, which focuses on performing sensor fusion of multiple observations in the form of multiple levels of derivatives of a measurand. The sensors may include a position sensor, or the derivative sensor, e.g. a velocity sensor or an acceleration sensor. In realistic applications, the position sensor and the acceleration sensor may have different sampling rates and different noise characteristics. In this chapter, we further develop the work on GP priors in Murray-Smith & Pearlmutter (2005) by proposing a novel and improved GP prior model, which takes account of the different sampling rates and different noise characteristics

of the sensors, and the Kinect latency in our problem. We define Y_{high},

$$Y_{high} \sim \mathcal{N}(0, \Sigma), \tag{4.16}$$

where Σ is the $N \times N$ covariance matrix, the elements of which are functions of inputs X.

we denote the Kinect measurements as Y_{low}, which are the low-sampling-rate observations in the high-sampling-rate position space. $Y_{low} = [y_1, \cdots, y_n]^T$ is denoted as M_p, and the high-sampling-rate acceleration measurements $M_a = [a_1, \cdots, a_N]^T$.

Following this, we assume the observations $M = KY_{high}$, where Y_{high} represents the assumed high-sampling-rate position measurements, and K is known. For the Kinect, the low-sampling-rate position measurements $M_p = K_p Y_{high}$, where K_p is defined in equation (4.17). For the mobile device, K_a is defined in equation (4.13), and Δt is the time step, which indicates the time difference between two consecutive inputs. Here, $\Delta t = \frac{1}{90}s$. The acceleration measurements $M_a = K_a Y_{high}$.

The connection between the low-sampling-rate positions and the high-sampling-rate accelerations can be expressed in an overall K_{all} matrix, which is defined in equation (4.18). The two components K_p and K_a are defined in equation (4.17) and equation (4.13), respectively. By constructing an overall K_{all} matrix, we can build a Gaussian Process prior model, which incorporates the low-sampling-rate position measurements and the high-sampling-rate acceleration measurements.

$$K_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\ & & & & \vdots & & & & & \ddots \end{bmatrix} \tag{4.17}$$

Now we can construct an overall transformation matrix,

$$K_{all} = \begin{bmatrix} K_p \\ K_a \end{bmatrix}. \tag{4.18}$$

According to equation (4.5), we need to find the joint distribution of low-sampling-rate position and the high-sampling-rate acceleration. The GP training target M_{all} includes the position and the acceleration.

$$M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} = [y_{n-L+1}, \cdots, y_n, a_{l-l_a+1}, \cdots, a_l]^T, \tag{4.19}$$

where the most recent position y_n and the most recent acceleration a_l are acquired at the same time instant.

Meanwhile, the correlation between M_{all} and K_{all} is expressed as

$$M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} = \begin{bmatrix} K_p \\ K_a \end{bmatrix} Y_{high} = K_{all} Y_{high}. \tag{4.20}$$

With the transformed GP prior method, we find this joint distribution expressed through an overall transformed GP prior

$$M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} \sim \mathcal{N}\left(0, K_{all}\Sigma K_{all}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right). \tag{4.21}$$

The Gaussian Process Prediction According to equations (4.22)) and (4.23), we can calculate the conditional mean and variance of the predictive position P_{fusion} with GP sensor fusion method.

$$\mu_{2|1} = I_{l_a}\Sigma_{12}K_{all}^T\left(K_{all}\Sigma K_{all}^T\right)^{-1}M_{all}, \tag{4.22}$$

$$\Sigma_{2|1} = \Sigma_2 - I_{l_a}\Sigma_{12}K_{all}^T\left(K_{all}\Sigma K_{all}^T\right)^{-1}K_{all}\Sigma_{21}I_{l_a}^T, \tag{4.23}$$

$$P_{fusion} = I_{l_a}\Sigma_{12}{K_{all}}^T\left(K_{all}\Sigma {K_{all}}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right)^{-1}M_{all}. \tag{4.24}$$

Σ_{12} represents the covariance matrix between the training inputs and the test inputs, whereas Σ denotes the covariance matrix between the training inputs and themselves. The Σ_p and Σ_a represent the diagonal matrices of position and acceleration observation variances respectively. Σ is a $l_a \times l_a$ matrix. K_{all} is a $(L + l_a) \times l_a$ matrix as K_p is a $L \times l_a$ matrix in the form of equation (4.17) and K_a is a $l_a \times l_a$ matrix in the form of equation (4.13).

The GP prior model-based sensor fusion can help increase the stability of the position, i.e. improve the accuracy of mean position estimation and reduce the position uncertainty, and reduce the lag of the system. We tested the performance of the GP prior model-based sensor fusion approach in section 4.5.

The Gaussian Process Log-Likelihood In this work, the Gaussian Process likelihood represents the probability density function (pdf) evaluated at the GP target vector. The target vector M_{all} is defined in equation (4.19). The joint likelihood of position and acceleration represents the joint probabilistic density of the combination of position and acceleration under the GP prior model. The higher the likelihood is, the more likely the target vector is. One part of the target vector M_{all} is a particular series of position measurements from a particular user's skeleton sensed by the Kinect, the other part is the acceleration measurements, which are sensed by one of the mobile devices in the room. When we match this position part with multiple acceleration parts, we get multiple combinations for a particular position signal, i.e.

a particular user's skeleton. By comparing the log-likelihoods of matching a particular user's skeleton with mobile devices, we can determine the best acceleration match for this time series of position measurements. The highest joint log-likelihood indicates the best match. In this way, we achieve user matching for this user. By doing this for each user, we get all user matching results. This is the way to match the motion sensed by the Kinect with the motion sensed by the mobile inertial sensors in this thesis.

Another way to associate the acceleration measured by the mobile inertial sensors with the position sensed by the Kinect sensor is to compare the acceleration distance. This is the conventional user matching method. In Chapter 5, we compare our method with the straightforward acceleration distance comparison method and prove that our method is more robust and more applicable.

As discussed in section 2.4.1, the GP log marginal likelihood given training data Y is defined in equation (2.13). When the training data consist of the Kinect low-sampling-rate position measurements and the high-sampling-rate acceleration measured by the mobile inertial sensors, the training data become M_{all}, a combination of position and acceleration. We need to find the joint distribution of the positions and the accelerations. We use the transformations of GP priors to construct an overall GP prior, which can be put on the combination function space. We use the joint GP prior method to compute the joint log-likelihood of position and acceleration. A detailed description on how to fuse Kinect sensor and mobile inertial sensors for calculating this joint log-likelihood will be presented in section 5.3.2.

How does the Sensor Fusion Model Work? From equations (4.17), (4.13) and (4.18), we know how to construct the overall K_{all} matrix. How does it relate to the measurements including the positions and the accelerations here?

In Figure 4.3, by the time we have two observations p_3^d and p_4^d, we will also have 13 accelerations. Here K_p is a 2×13 matrix and K_a is a 13×13 matrix. Thus,

$$K_{all,L=2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta t^2} & \frac{-2}{\Delta t^2} & \frac{1}{\Delta t^2} & & & & & & & & & & \\ & \ddots & & & & & & & & & & & \\ & & & & & & \ddots & & & & & & \\ & & & & & & & & & & & \ddots & \\ & & & & & & & & & & \frac{1}{\Delta t^2} & \frac{-2}{\Delta t^2} & \frac{1}{\Delta t^2} \end{bmatrix}. \tag{4.25}$$

From equation (4.25), we can see that there are $3L + 7$ accelerations (13 here) when there

are L position observations (2 here). There are 9 more accelerations ($a_5, \cdots, a_{13}$ here).

$$K_{all,L=2} Y_{t=\frac{13}{90}s} = M_{all,L=2}, \tag{4.26}$$

where $Y_{t=\frac{13}{90}s}$ is the assumed high-sampling-rate position observations and

$$M_{all,L=2} = \begin{bmatrix} p_3^d & p_4^d & a_1 & a_2 & \cdots & a_{13} \end{bmatrix}^T,$$

are the targets including the low-sampling-rate positions and the higher frequency accelerations. If we set the test inputs to $t = \frac{5}{90}s, \cdots, \frac{13}{90}s$, the prediction is based on $a_5, \cdots, a_{13}$ during the test inputs period and we get 9 predictive positions with GPs according to (4.24).

When we receive the acceleration measurement a_{10} at $t = \frac{10}{90}$s and the delayed position p_3^d, we assume that this delayed position is acquired at $t = \frac{1}{90}$s. For $\frac{10}{90}$s, this is the only available position measurement. For $t = \frac{13}{90}$s, we have 13 acceleration measurements and 2 position measurements including p_3^d and p_4^d, which are assumed to be observed at $t = \frac{1}{90}s$ and $t = \frac{4}{90}s$, respectively. For $\frac{16}{90}s$, we will have 16 acceleration measurements and 3 position measurements. For the following time instants, the process goes on like this. In this way, we apply GPs for real-time position prediction.

Figure 4.4 illustrates how the sensor fusion model works. The position Y measurements and the GP sensor fusion results are shown. Since there is time delay (0.1s) in the Kinect system, the position acquired at the current time is the delayed measurement. The real-time predictive positions with GP are shown. 3 predictive positions, which are based on the test inputs and denoted with the asterisks, become available as we receive 3 acceleration during each $\frac{1}{30}$ time interval. Whenever we receive a position update, the uncertainty of the GP prediction is decreased. During the $\frac{1}{30}$s time interval when no position is available, the uncertainty of GP prediction keeps increasing. It can be seen that the real-time GP prediction helps stabilise the hand position as the curve (real-time GP predictive positions) is becoming smoother in comparison with the curve (Kinect position measurements). The GPs also make use of the past and current observations to estimate all the states including the previous and current positions. We can see that the results become smoother and the uncertainty is further reduced.

We have illustrated how the sensor fusion model works through Figure 4.4. The GP prior model-based sensor fusion incorporates the delayed, noisy and low-sampling-rate Kinect position measurements and the higher frequency acceleration measured by the mobile inertial sensors for position prediction, taking account of the different noise characteristics of each sensor type.

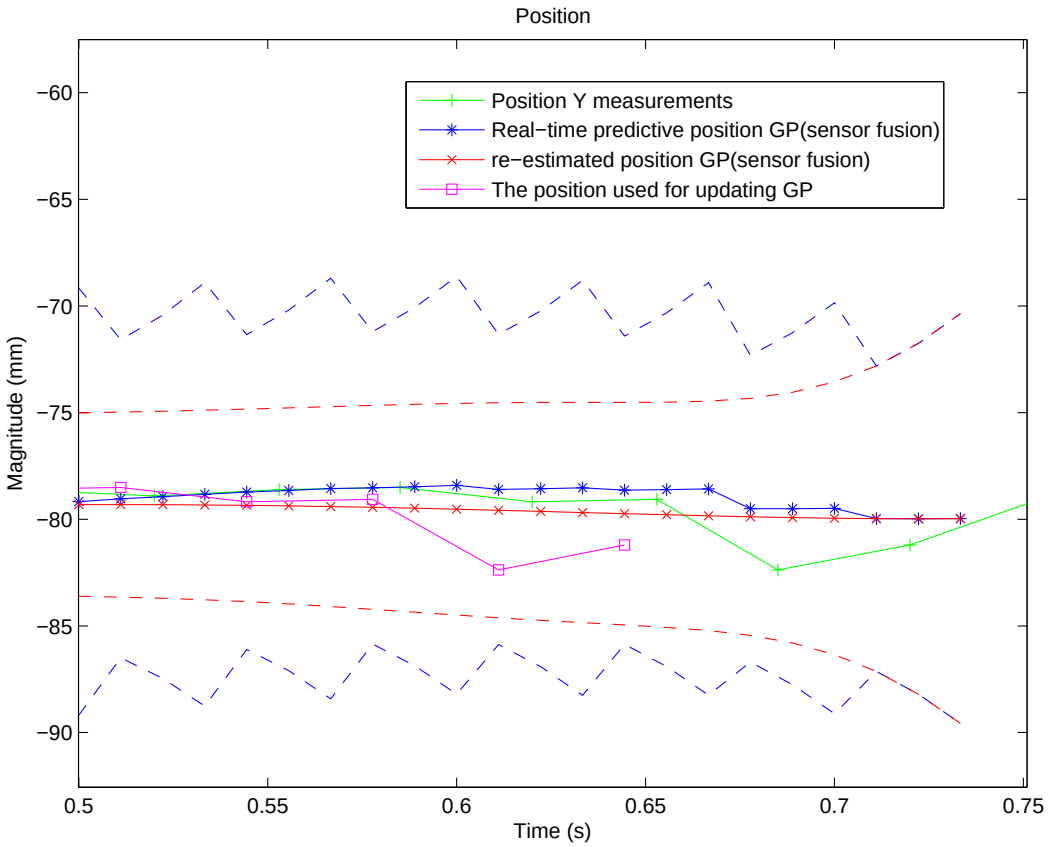


Figure 4.4: Plots show mean and mean$\pm 2\sigma$ as solid line and dashed contours. There are 4 signals: (1) The noisy position measurements (2) The real-time GPs: There are 3 predictive positions, which are based on the test inputs and denoted with the asterisks, each time. (3) The GPs estimate the previous and current states based on all the available observations (4) The position measurements used for updating GPs. Whenever we receive a position measurement, the uncertainty of GP prediction is decreased. During the $\frac{1}{30}$s time interval when no position is available, the uncertainty of GP prediction keeps increasing. It can be seen that the real-time GP prediction helps stabilise the hand position.

4.4 Alternative View of the Sensor Fusion – Multi-rate Kalman Filter

Since the Kalman filter has been widely used for sensor fusion, we put our problem in a Kalman filter framework in order to better illustrate this concept. We consider the autoregressive model in the state space here. This is a time-delayed system as there is 0.1s latency in the Kinect system. Also, this is a multisensor data fusion problem as the Kinect sensor and the inertial sensors have different sampling rates and different noise characteristics. Recent work on using a Kalman filter to solve the state estimation problem for multisensor systems with irregular sampling and time-varying delays is introduced in (Peñarrocha et al., 2012).

Let us denote the system state as x_k at time instant k, where the state of the moving target includes the position p_k, the velocity v_k and the acceleration a_k. This is defined in equa-

tion (4.27)

$$x_k = [p_k, v_k, a_k]^T. \tag{4.27}$$

Note that the time step $\Delta t = \frac{1}{90}s$ and p_k is the position in the high-sampling-rate position space. In the target tracking area, the solution for state estimation is to build a process model and a measurement model. By extending the order of the state and building a new state vector, we can employ the following autoregressive (AR) model:

$$x_{k+1} = A_0 x_k + A_1 x_{k-1} + \cdots + A_{m-1} x_{k-(m-1)} + w_k, \tag{4.28}$$

where A_i ($i = 0, 1, \cdots, m-1$) denotes the corresponding state transition matrix for x_{k-i}. $p(w) \sim \mathcal{N}(0, Q)$, where Q represents the process noise covariance matrix.

The state process model equation (4.28) can be written as

$$\begin{bmatrix} x_{k+1} \\ x_k \\ \vdots \\ x_{k-(m-1)+1} \end{bmatrix} = \begin{bmatrix} A_0 & A_1 & \cdots & A_{m-1} \\ & & & 0 \\ I & & & \vdots \\ & & & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-(m-1)} \end{bmatrix} + w_k. \tag{4.29}$$

The equation (4.29) can be rewritten as

$$x_{k+1}{}' = A_k{}' x_k{}' + w_k. \tag{4.30}$$

Considering the availability of observations in Figure 4.3, we also define the available measurements include the most recent L positions and the l_a accelerations. Here we define $x_k{}'$ in equation (4.30) as

$$x_k{}' = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-(m-1)} \end{bmatrix} \tag{4.31}$$

That is,

$$x_k{}' = \begin{bmatrix} p_k & v_k & a_k & p_{k-1} & v_{k-1} & a_{k-1} & \cdots & p_{k-(m-1)} & v_{k-(m-1)} & a_{k-(m-1)} \end{bmatrix}^T, \tag{4.32}$$

where $m = 3L + N_0 - 2$.

We can see that the number of elements in the extended order state vector $x_k{}'$ is $3 \cdot m$, among which there are m positions, m velocities and m accelerations.

In our GP prior model, the available observations include the L low-sampling-rate positions

and the l_a high-sampling-rate accelerations. We set $m = 3L + N_0 - 2$ here. Among these available measurements at time instant k, there are L positions and m accelerations.

$$H_{pl} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

$$H_{pL} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.34}$$

$$H_{ai} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \tag{4.35}$$

where $l = 1, 2, \cdots, L-1$ and $i = 1, 2, \cdots, m$. H_p is a $L \times (3N_0 + 9(L-1) + 3) = L \times 3 \cdot m$ matrix. H_a is a $m \times 3 \cdot m$ matrix.

$$H_p = \begin{bmatrix} & H_{p1} & & & & \\ & & H_{p2} & & & \\ 0_{L\times 3N_0} & & & \ddots & & \\ & & & & H_{p(L-1)} & \\ & & & & & H_{pL} \end{bmatrix} \tag{4.36}$$

$$H_a = \begin{bmatrix} H_{a1} & & & \\ & H_{a2} & & \\ & & \ddots & \\ & & & H_{a(m)} \end{bmatrix} \tag{4.37}$$

$$H_k{}' = \begin{bmatrix} H_p \\ H_a \end{bmatrix} \tag{4.38}$$

The extended measurement model is

$$y_k{}' = H_k{}' x_k{}' + v_k, \tag{4.39}$$

where $y_k{}' = \begin{bmatrix} p_{k-N_0} & p_{k-(N_0+3)} & \cdots & p_{k-(m-1)} & a_k & a_{k-1} & \cdots & a_{k-(m-1)} \end{bmatrix}^T$ represents the available measurements at time instant k. a_k denotes the most recent acceleration measurement while p_{k-N_0} represents the most recent position measurement. $x_k{}'$ is the extended state vector ($3 \cdot m \times 1$) as defined in equation (4.31). $p(v) \sim \mathcal{N}(0, R)$, where R represents the measurement noise covariance matrix. w_k and v_k are independent Gaussian white noise.

The Kalman filter estimates the system state by building a state transition model, i.e. the process model, and a measurement model. A simple case ($m = 1$ in equation (4.28)) is not applicable for the time-delayed system. Thus, we need to use the extended order model in

equation (4.29) ($m > 1$). We define A_i,

$$A_i = \alpha_i \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ & 1 & \Delta t \\ & & 1 \end{bmatrix}^{i+1}, i = 0, 1, \cdots, m-1. \tag{4.40}$$

Now we explain the reason why we set A_i in the form of equation (4.40). If there is no delay in the system, we can rewrite equation (4.28) as

$$x_{k+1} = B_0 x_k + w_k. \tag{4.41}$$

It also means

$$x_k = B_0 x_{k-1} + w_{k-1}. \tag{4.42}$$

Thus, we get

$$x_{k+1} = B_0^2 x_{k-1} + B_0 w_{k-1} + w_k. \tag{4.43}$$

If we define B_0 as

$$B_0 = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ & 1 & \Delta t \\ & & 1 \end{bmatrix}, \tag{4.44}$$

we can find the connection between A_i defined in equation (4.40) and B_0.

$$A_i = \alpha_i B_0^{i+1}, i = 0, 1, \cdots, m. \tag{4.45}$$

α_i is a weight that defines how much influence the associated state, i.e. x_{k-i}, has on the state prediction.

The Kalman filter can be seen as a special case of a GP, as discussed in section 2.3.3. In a Kalman filter, the state transition model, i.e. the process model, is applied for predicting the future state and the measurements are used for updating the system state. In our GP prior model, we employ the covariance function, as defined in equation (2.14). For position prediction, the influences of the available observations depend on the distances, that is, the closer observations have higher influences on the predictive positions than those that are far away. The distance is based on the time instants when the observations are received. The predictive position is a weighted sum of the recent available observations including the L low-sampling-rate positions and the l_a high-sampling-rate accelerations.

The key problem in correlating our proposed novel GP prior model with this Kalman filter solution is how to form ${A_k}'$ in equation (4.30) such that the ${x_k}'$ has the desired covariance function $K_{all}\Sigma K_{all}^T$. The problem of reformulating temporal Gaussian Process regression models as linear-Gaussian state space models has been investigated in (Hartikainen & Sarkka, 2010;

Särkkä et al., 2013). In this work, we focus on the GP prior model for sensor fusion.

4.5 Experiment

We conducted an experiment to test the performance of the proposed GP prior model-based sensor fusion system. In this experiment, we used a Leap Motion Controller to sense the hand position (90Hz). The V2 Tracking Beta SDK provides the hand tracking with high accuracy and near-zero latency (Motion, 2014). This was used as the baseline for evaluating the performance of the GP sensor fusion method. Meanwhile, we collected the hand position data sensed by the Kinect and the hand acceleration measured by the mobile inertial sensors. We compared the sensor fusion approach with the position-only Kalman filter prediction method and the position-only GP, and concluded that the GP prior model-based sensor fusion is superior to the two methods. The proposed approach can be used to improve the accuracy of position estimation and reduce the lag.

4.5.1 Experiment Design

Before starting the experiment, we calibrated the position tracking systems including the Leap Motion Controller and the Kinect sensor. The inertial sensors have also been calibrated. We aligned the Kinect frame and the Leap Motion tracking frame, and analysed the hand movement along the $x-$axis as an example. In this way, the two frames have the same origin along the $x-$axis in the space.

In this experiment, the user's right hand motion was sensed by the Leap Motion Controller, the Kinect and the inertial sensors pack. The user put the hand above the Leap Motion Controller (the height is approximately 20cm), and performed a hand movement with a mobile device (SK7) held in the hand in the Kinect field of view. The distance between the Kinect and the Controller is 1.5m. At the beginning, the user put the hand above the controller, then moved the hand along the $+x-$axis (the distance is approximately 20cm) and then stopped. The process took 2s. The hand position sensed by the Controller is used as the baseline data for comparison. We also recorded the hand position sensed by the Kinect. The hand motion data sensed by mobile inertial sensors were fused to estimate the hand acceleration.

The proposed GP prior model was applied for fusing the low-sampling-rate Kinect position and the high-sampling-rate acceleration measured by mobile inertial sensors.

4.5.2 Experimental Method

In this experiment, we test the GP prior model-based sensor fusion approach. We chose $L = 5$ as this is very computationally efficient and can give a good prediction result. When the number of the available assumed non-delayed position measurements, denoted as n, is smaller than L, we adaptively change the K_p defined in equation (4.17) and K_a defined in equation (4.13) as this equals to the situation that n is a smaller L. When $n \geq L$, a fixed K_{all} matrix is used. The method is presented in the following algorithm.

Algorithm 1 Illustration of how the proposed Autoregressive GP model works

1. Measure the acceleration with mobile inertial sensors and the Kinect position.
2. The number of the assumed non-delayed position (the fourth row in Figure 4.3) is denoted by n. According to n, construct K_p and K_a accordingly (see (4.17) and (4.13)).

   When $n < L$, adjust K_p and K_a accordingly (This equals a smaller L). K_p is a $n \times (3 \cdot n + N_0 - 2)$ matrix and K_a is a $(3 \cdot n + N_0 - 2) \times (3 \cdot n + N_0 - 2)$ matrix, where $(n = 1, \cdots, L - 1)$.

   When $n \geq L$, K_p and K_a are both fixed matrices.
3. Construct the target vector M_{all} according to (4.19)
4. Start the GP fusion. Make prediction according to (4.22) and (4.23).

We compared the proposed approach with the position-only Kalman filter and the position-only GP method. We built a position-only KF, which uses a continuous Wiener process acceleration model as discussed in section 3.2.2. This position-only KF makes 1 step ($\frac{1}{30}$s) prediction first, then the Kinect position measurement is used to update the system state. Based on the updated state, this KF makes 3 steps ahead prediction to deal with the 0.1s delay. We also compared the GP sensor fusion with the position-only ARGP method, which uses the most recent L position measurements for multi-step ahead prediction. As there is a 0.1s delay and the sampling rate of the Kinect is 30Hz, the position-only GP makes 3 steps prediction. The position-only GP and the GP sensor fusion use the same hyperparameters, the maximum likelihood estimate of which can be calculated using the time-stamped human motion training data and the standard optimisation algorithm. The hyperparameters are tuned to typical human motion, which is relatively continuous and smooth.

The uncertainty of the Kinect position and the acceleration measured by the mobile inertial sensors have been discussed in section 3.3.1. The GP hyperparameters are set to $v_0 = 5.66 \times 10^4$, $\omega_1 = 4.19$, $\sigma_y^2 = 64$ and $\sigma_a^2 = 100^2$.

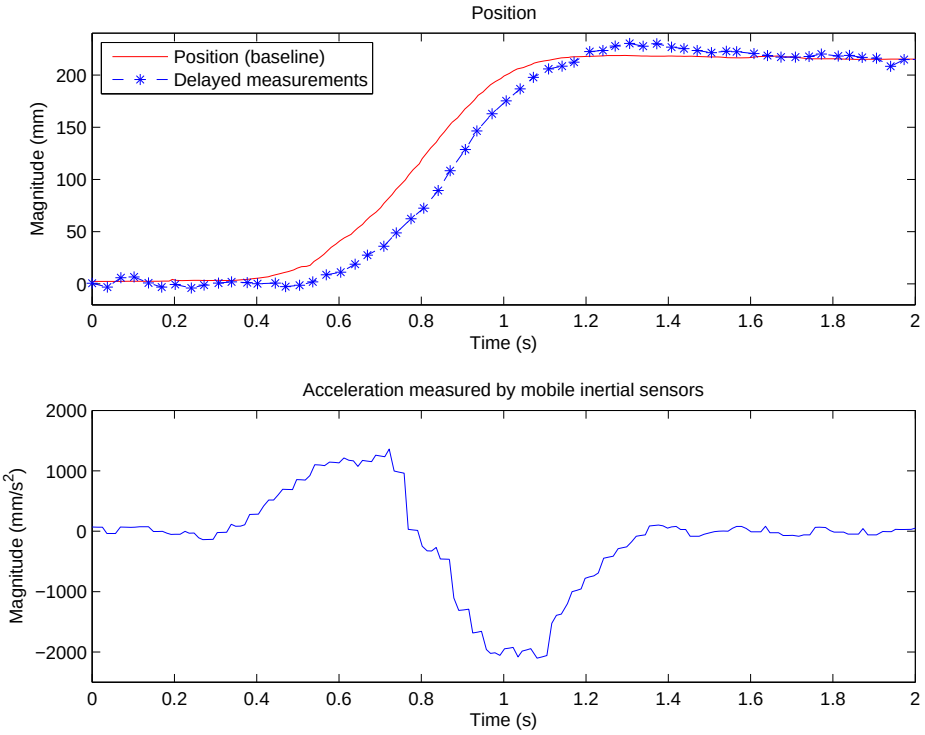


Figure 4.5: Upper panel: The position X measurements and the baseline data. Lower panel: acceleration X estimated with inertial sensors and expressed in Kinect coordinate system.

4.5.3 Experimental Results

Measurements

In the experiment, the hand position measurement was sensed by the Kinect sensor and the acceleration was measured by mobile inertial sensors held in the hand. The hand position sensed by the Leap Motion Controller was used as the baseline. Figure 4.5 illustrates the position X measurements (in the upper panel) and the corresponding acceleration X measurements (in the lower panel). We can see that the Kinect position measurements are noisy and delayed. The GP sensor fusion is to fuse the noisy, delayed low-sampling-rate position observations and the higher frequency acceleration measurements with the proposed GP prior model.

Sensor Fusion and Comparison

In this part, the Kinect position observations and the acceleration measurements are fused with the GP prior model-based sensor fusion approach. In order to illustrate the benefits of the proposed approach, we compare it with the position-only Kalman filter prediction and the position-only GP method.

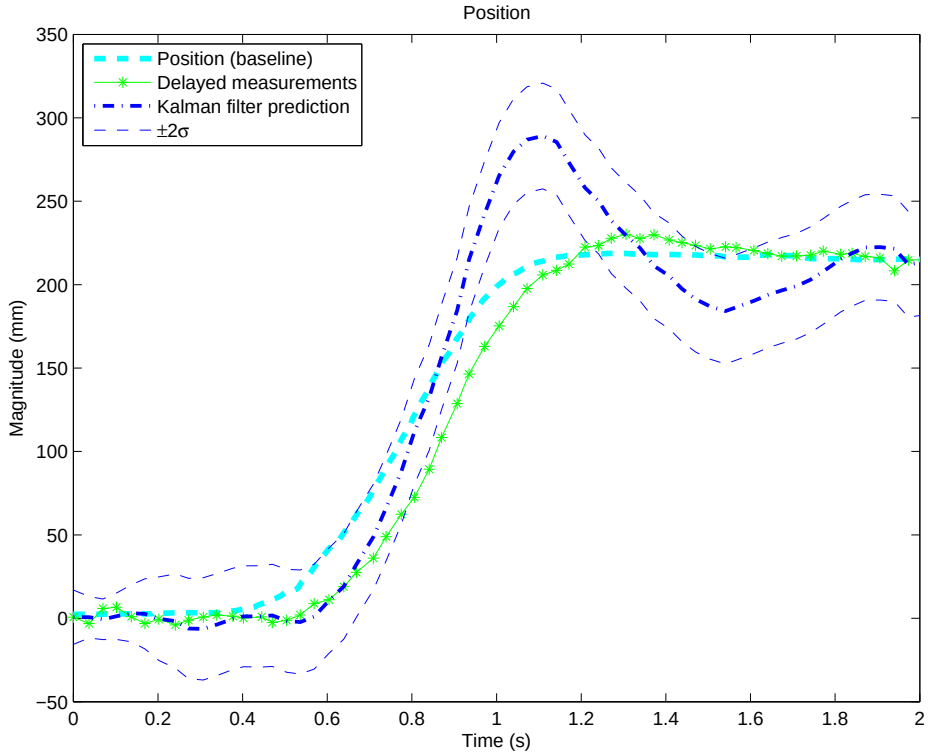


Figure 4.6: The position-only Kalman filter prediction. Plots show the mean $\pm 2\sigma$. The figure shows 3 signals: (1) the baseline data (2) the position measurements (3) the predictive positions with the position-only KF. Plots show the mean $\pm 2\sigma$.

The Position-only Kalman Filter Prediction Figure 4.6 shows 3 signals, including (1) the baseline data, (2) the position measurements and (3) the predictive positions with the position-only KF. We can see that there is an overshoot problem with the position-only KF prediction. We analysed the accuracy of the position predicted with this position-only KF by comparing the prediction results with the baseline data. The results are summarised in Table 4.1.

Comparison with the Position-only GP In addition to the position-only KF, we also compare the GP sensor fusion with the position-only GP. The experimental results are shown in Figure 4.7, which shows 4 signals, including (1) the baseline data, (2) the position measurements, (3) the position-only GP prediction result and (4) the predictive positions with the GP sensor fusion method. We use the method described in the Algorithm 1 in section 4.5.2. We can see that the position prediction with the GP sensor fusion is smoother in comparison with the position-only GP result. Besides, the uncertainty of position prediction with the GP sensor fusion is much smaller than that of the position predicted with the position-only GP method. The comparison results are summarised in Table 4.1. Moreover, the system lag is reduced with the GP sensor fusion approach. This proves that the high-sampling-rate acceleration can compensate for the effect of position uncertainty and lag in the Kinect system.

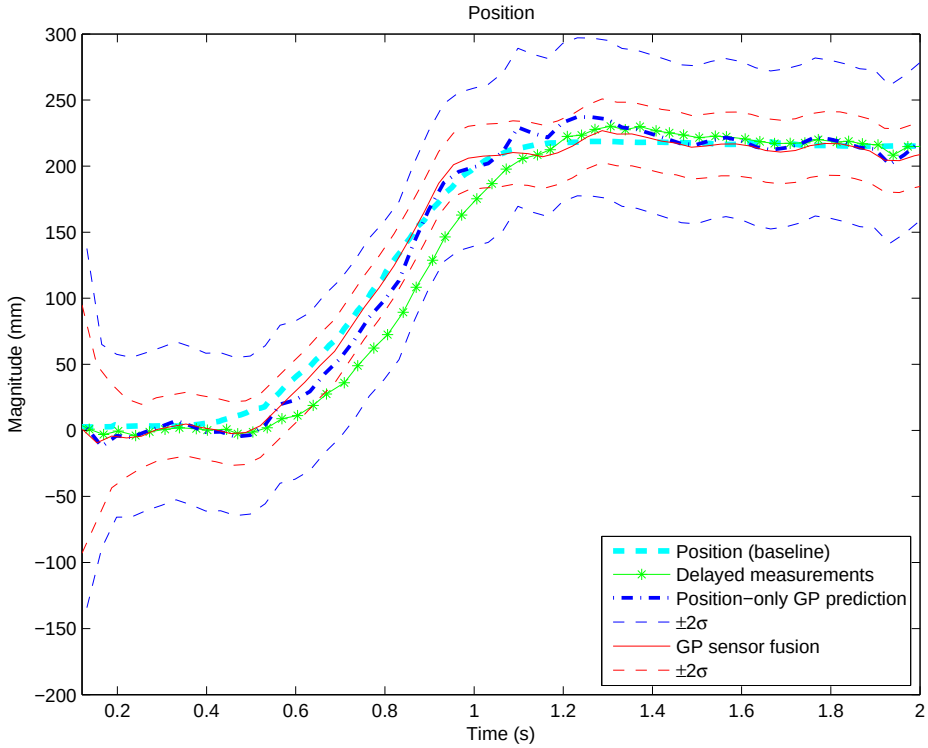


Figure 4.7: Comparison of position-only GP and sensor fusion with GP (L=5). Plots show the mean $\pm 2\sigma$. The figure shows 4 signals: (1) the baseline data (2) the position measurements (3) the position-only GP prediction (4) the prediction with the GP sensor fusion.

Accuracy of Position Estimation In order to analyse the accuracy of the mean position prediction, we calculate the RMSE based on the baseline data. For the KF, the position-only GP and the GP sensor fusion approach, this RMSE is the root of the average of the squares of the difference between the mean predictive positions and the baseline data. We compared the GP sensor fusion approach with the position-only KF prediction and the position-only GP prediction method. The results are summarised in Table 4.1.

In comparison with the baseline position data, the RMSE of the noisy and delayed position measurements sensed by the Kinect is 19.75mm. The measured uncertainty (standard deviation SD) is 8mm. The RMSE of the mean position predicted by the position-only Kalman filter is 29.19mm. The uncertainty (SD) after the KF converges is 15.84mm. The RMSE and uncertainty of the mean position predicted with the GP approaches are illustrated in Table 4.1. We can see that the sensor fusion with GP helps reduce the error of mean position prediction and the uncertainty of the prediction. In comparison with the position-only GP, the RMSE of the mean position prediction is reduced by 35.8% and the uncertainty of the mean position prediction was reduced by 59.7%.

Thus, the proposed approach is superior to the position-only KF and the position-only GP method. The KF is a special case of a GP and the proposed approach can be put in a KF framework and implemented by carefully designing a customised variant of the multi-rate

KF. This has been discussed in section 4.4. We conclude that the proposed GP prior model-based sensor fusion helps improve the accuracy of the position estimation.

Table 4.1: Comparison of accuracy – compare the GP sensor fusion approach with the position-only KF and the position-only GP method.

		Accuracy	
		RMSE(mm) of mean prediction	Uncertainty(SD σ)(mm)
Methods	Position-only KF	29.19	15.84
	Position-only GP	10.76	29.89
	GP sensor fusion	6.91	12.04

Lag Reduction Besides the improved accuracy of the position estimation, the proposed approach can also help reduce the system lag. We can see this in Figure 4.7. Now we use the unbiased estimate of the cross-correlation function to analyse the time delay between the GP predictive position signal and the Kinect position measurement signal. Figure 4.8 shows the cross-correlation sequence in a length 359 vector, where the GP predictive position signal and the Kinect measurement signal are both vectors of length 180 (interpolation 90Hz), respectively. The peak was acquired at 190. Thus, the lag was reduced by 0.11s.

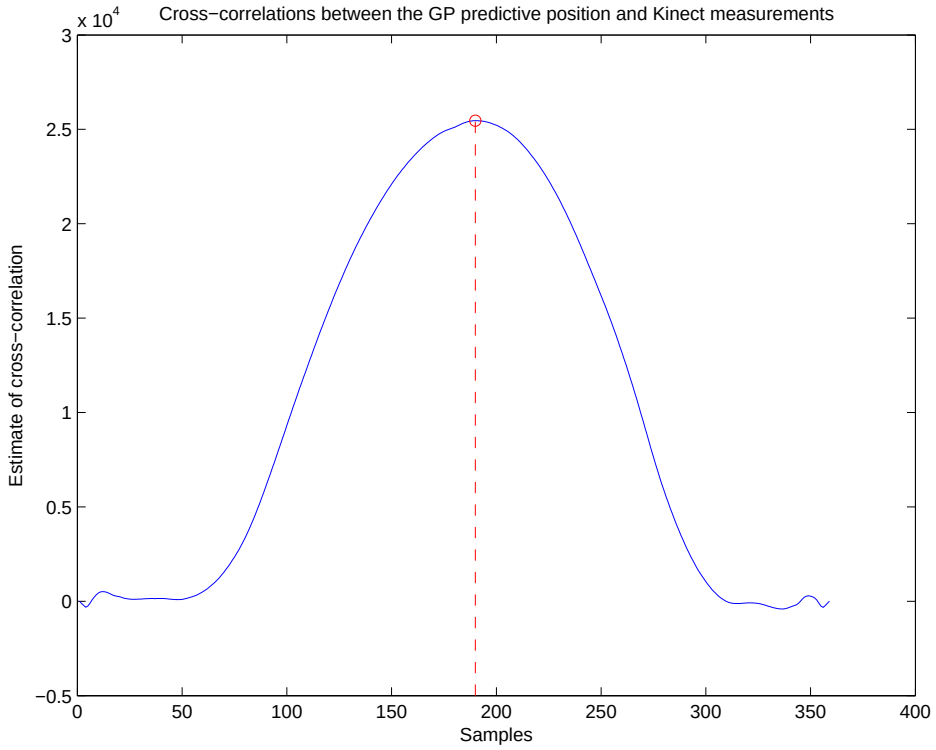


Figure 4.8: The GP sensor fusion helps reduce the lag. Plot show the cross-correlation sequence in a length 359 vector, where the GP predictive position signal and the Kinect measurement signal are both vectors of length 180 (interpolation 90Hz). The peak was acquired at 190. Thus, the lag was reduced by 0.11s.

4.5.4 Conclusion

In this experiment, we tested the proposed GP prior model-based sensor fusion approach. We applied the novel and improved GP prior model to fuse the noisy, low-sampling-rate position measurements sensed by the Kinect and the high-sampling-rate accelerations measured by the mobile inertial sensors. We compared the proposed approach with the position-only KF and the position-only GP method. Experimental results show that the mean position prediction with the proposed GP prior model-based sensor fusion is more accurate and the uncertainty of the position prediction is much smaller. Moreover, the lag of the Kinect system is reduced by 0.11s with the GP sensor fusion approach. Thus, the sensor fusion with the proposed GP prior model helps improve the accuracy of position estimation, and reduce the lag of the conventional Kinect system.

4.6 Conclusions

In this chapter, we present a sensor fusion system composed of a Kinect sensor and a mobile device for improving the accuracy of the skeleton joint position estimation and reducing the lag by fusing the Kinect sensor and the built-in inertial sensors in a mobile device. We explore the complementary properties of the position sensing device (Kinect) and the mobile inertial sensors.

The sensor fusion system is based on a novel and transformed Gaussian Process prior model, which incorporates the low-sampling-rate measurements and the high-sampling-rate derivatives, and takes the different noise characteristics of these sensors into account. In order to better illustrate the sensor fusion approach, we discussed the relationship between our GP prior model and the Kalman filter in detail. It is beneficial for multi-rate sensor fusion in a non-linear dynamical system.

The GP prior model-based sensor fusion approach can be used for user matching and identification. In section 4.3.2, we described the use of the proposed GP prior model to calculate the joint log-likelihood of the Kinect positions and the accelerations measured by the mobile inertial sensors. This allows us to identify individual users, by matching the observed Kinect skeletons with the sensed inertial data from their mobile devices using the GP-based sensor fusion algorithm.

Moreover, the GP prior model-based sensor fusion approach is very beneficial for improving the usability of a location-aware interaction system by increasing the stability of the position and reducing the lag. We have discussed how to apply this model for position prediction in section 4.3.2. Firstly, the sensor fusion can improve the quality of inferred joint positions, as the high-sampling-rate acceleration signal can augment the low-sampling-rate, noisy posi-

tion measurements. It can also help to reduce lags, as the inertial sensing has a lower latency than the position sensed by the Kinect.

We conducted an experiment to test the GP prior model-based sensor fusion system. In this experiment, the user's hand movement was sensed by the Leap Motion Controller, the Kinect sensor and the mobile inertial sensors. We applied the proposed sensor fusion method to fuse the Kinect position and the higher frequency acceleration. Experimental results show that the proposed GP prior model-based sensor fusion approach is superior to the position-only KF prediction and the position-only GP prediction method. The GP sensor fusion helps increase the accuracy of the position estimation, and reduce the lag of the conventional Kinect system.

The GP prior model-based sensor fusion approach is of great benefit for combining mobile inertial sensors and external position sensing device for mobile augmented reality (AR) and other location-aware sensing applications, in which the human movement is typical of everyday movement, relatively continuous and smooth. The increased stability of the position and the faster response of the sensor fusion system give the user a smoother experience and a faster response.

Chapter 5

Transformations of Gaussian Process Priors for User Matching

In this chapter, we apply the Gaussian Process prior model-based sensor fusion approach to user matching. Firstly, we give an introduction to user matching and identification in human-computer interaction (HCI). Secondly, we describe how to calculate a joint log-likelihood of the low-sampling-rate position and the high-sampling-rate acceleration with the proposed GP prior model. Thirdly, we test the use of the transformations of GP priors in a simulation experiment. Fourthly, we conduct three experiments including (1) subtle hand movement (2) mobile device in the user's trouser pocket and (3) walking with the device held in the hand. We investigate the performance of the joint log-likelihood comparison method using the proposed GP prior model and compare the proposed approach with the direct acceleration comparison method in all 3 contexts.

5.1 Introduction

As discussed in the proposed scenario in section 1.1, user matching and identification plays a crucial role in providing user-specific information and services in a proxemic interaction system. In this chapter, we investigate the usability of the proposed sensor fusion system for user matching and identification through people's everyday movements. We apply the proposed Gaussian Process prior model for fusing the low-sampling-rate position measurements sensed by the Kinect and the high-sampling-rate acceleration measured by the mobile inertial sensors. The sensor fusion combines data from multiple sensors (Hall & Llinas, 1997), and can be applied for matching a particular user's skeleton with a mobile device. When several users are in a room and each of them carries a mobile device, we can find the matching relationship between the users and the mobile devices.

The advanced human motion tracking techniques and ubiquitous sensors enable the HCI researchers to exploit the mobile device-based user identification techniques (Guna et al., 2012; Rofouei et al., 2012). In addition to the Kinect, the inertial sensors provide another way of sensing human movement. An increasing number of consumer devices, such as tablets and smartphones, are equipped with advanced inertial sensors, which can be used to sense how the device is moved. If the Kinect can sense multiple people in the room and each has a device in the hand or pocket, which person carries which device?

In order to identify the correlation between the person and the device, we need to associate the motion sensed by the Kinect sensor with the motion sensed by the mobile inertial sensors. The Kinect sensor and inertial sensors have complementary properties, as discussed in section 1.2.2. Solving this efficiently by exploring the complementary properties of these sensors, minimising the extent of the movement needed for matching, would allow a system to rapidly link the Kinect sensed people with their devices equipped with inertial sensors, and customise services appropriately for them.

User matching and identification has been studied in the literature, e.g. proxemic interaction (Greenberg et al., 2011) and human-computer interaction (HCI) (Shahidi et al., 2010). The user matching and identification systems in the literature will be introduced in section 5.2.

The fusion of the Kinect sensor and the mobile inertial sensors enables us to build a context-aware application. For example, by matching the skeletons sensed by the Kinect and the accelerations measured by the mobile devices carried by the family members using the GP prior model-based sensor fusion method, we can build a user matching and identification system. A potential application is to help family members personalize their own multimedia services and TV programs. Besides user matching, another benefit of the model is that we can apply the high-sampling-rate accelerations sensed by the mobile inertial sensors to augment the noisy, low-sampling-rate positions measured by the Kinect. Imagine this is a family environment and each member carries a mobile phone. Everyone can use his/her own phone for an augmented reality (AR) application in the room. The fusion of the hand position and acceleration increases the stability of the position and reduce the lag, improving the user experience. In order to fuse the position from a skeleton (a specific user) and the acceleration from a phone, we need to associate this particular user with one of the mobile phones in the room. After that, we can apply the acceleration sensed by this particular mobile phone to augment the noisy position from this specific user's skeleton tracking. In this chapter, we focus on user matching, i.e. finding the correlation between the multiple skeletons and the mobile devices.

Addressing these problems, we apply the GP prior model proposed in Chapter 4 for fusing the low-sampling-rate position sensed by the Kinect sensor and the high-sampling-rate acceleration measured by the inertial sensors. In section 4.3.2, we briefly introduced the

application of the GP model for user matching and identification. In this chapter, we give a detailed description on how to apply the GP prior model for matching the skeletons sensed by the Kinect with the time series of acceleration signals measured by the mobile devices.

We show how to calculate the log-likelihood of the low-sampling-rate position sensed by the Kinect and the high-sampling-rate acceleration sensed by the mobile inertial sensors using the GP prior model, which allows you to match the most likely combinations of sensors, consistent with the uncertainty in each sensor channel, instead of comparing the accelerations estimated with different sensors. An approach to this is described in (Rofouei et al., 2012), in which the authors proposed the ShakeID method, which is a technique for associating multi-touch interactions with individual users and their mobile devices. However, this method requires the user to perform a vigorous and visible hand movement.

In this work, we investigate the feasibility and usability of the GP prior model-based sensor fusion approach to associating a particular user with a mobile device carried by the user, based on the data from an embedded Kinect, and acceleration information from a mobile device, which can be held in the user's hand or be in the user's trouser pocket. We test the performance of our system in different contexts, including (1) subtle hand movement; (2) with the mobile device in the user's trouser pocket; (3) walking with the device held in the hand. Experimental results show that 6 users can be correctly matched and identified in all the contexts with our approach. We compared the proposed approach with the direct acceleration comparison method in all 3 contexts.

5.2 Background

We consider the problem of fusing the Kinect sensor and the built-in inertial sensors in a mobile device for user matching and identification in a proxemic interaction system using a novel GP prior model. We cover related work on user matching and identification systems.

Although user identification can be dealt with computer vision approaches, e.g. face recognition and gait (Kale et al., 2004; Chellappa et al., 2007), the focus of our work is not on identifying different users, but on the method of user matching, i.e. finding the correlation between multiple skeletons (users) and the mobile devices. Specifically, the user matching is to associate a particular skeleton sensed by the Kinect with one of the mobile devices. The matching results can be easily applied for user identification. Although the Kinect skeleton tracking only gives a user ID for each user, we can find who the user is when we know which mobile device the user is carrying, assuming each user carries his/her own mobile device and we know who the mobile device belongs to.

User Matching and Identification Systems

User matching and identification plays an important role in many fields, such as proxemic interaction (Greenberg et al., 2011), human-computer interaction (HCI) (Shahidi et al., 2010), smart environments (Kadouche et al., 2010), multimedia services and applications (Guna et al., 2012), and the multi-touch interactive displays for personalization and collaboration (Ramakers et al., 2012; Blažica et al., 2013).

User matching and identification in multi-touch interactive displays has attracted much interest (Rofouei et al., 2012; Ackad et al., 2012; Ramakers et al., 2012). Ackad et al. (2012) presented a system that could identify users and keep track of their actions around interactive tabletops. They explored the use of combining personal devices and the Microsoft Kinect sensor to provide a method for continuous user identification on tabletop surfaces. Ramakers et al. (2012) introduced a technique for user identification by observing the dorsal region of their hands with a high-resolution camera mounted above the interactive surface.

There is relatively little research effort on acceleration-based gesture for user identification (Guna et al., 2012). Farella et al. (2006) studied the use of inertial sensors embedded in mobile devices for personal identification. The implicit gesture-based user identification using a 3-axis accelerometer was studied in (Guna et al., 2012). Conti et al. (2011) focused on the movement that a user performs when answering (or placing) a phone call, and assessed the feasibility of using this movement as a biometric authentication measure.

The combination of the Kinect and inertial sensors for user matching and identification was studied in the literature (Rofouei et al., 2012). An accelerometer-equipped mobile phone was combined with a Kinect for associating the users with the devices. A Kalman filter (Welch & Bishop, 1995) was applied for estimating the acceleration of hand position. However, no detailed process noise covariance Q and the measurement noise covariance R were indicated. The acceleration estimated with the phone sensor was compared with that estimated with the Kalman filter over a time window. The ShakeID method requires the user to perform a vigorous and visible hand movement. As mentioned in (Rofouei et al., 2012), an important limitation of this method involves the case where the hand holding the phone is stationary. When the hand (phone) movement is subtle, it will be difficult for the ShakeID method to work because of the low signal strength.

5.3 Fusing Kinect Sensor and Inertial Sensors for User Matching

The proposed GP prior model presented in Chapter 4 is of great importance for multi-rate sensor fusion, e.g. fusing the Kinect sensor and the mobile inertial sensors. Firstly, we can

fuse the position and the acceleration with the GP prior model to improve the accuracy of position estimation and reduce the lag. This has been described in Chapter 4. Secondly, the model can be applied for computing the joint log-likelihood given different combinations of position and acceleration. This can be used for user matching, i.e. associating a particular user with one of the mobile devices. In this chapter, we match the skeletons sensed by the Kinect with the time series of acceleration signals measured by the mobile devices. In this way, we match a skeleton that represents a particular user with a mobile device in the room. The user matching is achieved.

In this section, we present the problem statement for user matching with GP priors first. Following this, we discuss the multi-rate sensor fusion technique for user matching and identification. We illustrate how to use the log-likelihood comparison method for user matching and identification with an example.

5.3.1 Problem Statement for User Matching with GP Priors

In our work, the user motion is typical of everyday movement, i.e. relatively smooth. The parameters specified by the covariance function are tuned to typical human motion. The maximum likelihood estimate of the hyperparameters can be calculated given the covariance function and the time-stamped human motion data using standard optimisation algorithm. We put a prior on the function space of the combination of position and acceleration. For the acceleration function, we apply a transformed GP prior, which is based on a GP prior on the position function space. In this way, we have an overall GP prior on the combination (position and acceleration) function space. Under the GP prior model, we can calculate the log-likelihood given different combinations of position and acceleration. We can determine how likely the different combinations of position and acceleration are under this model by calculating the joint log-likelihood with GPs.

In order to compare the log-likelihood of different combinations, we introduce the GP log marginal likelihood, given training data Y.

$$\log L(\theta) = -\frac{1}{2}Y^T\big(C(\theta) + \sigma^2 I\big)^{-1}Y - \frac{1}{2}\log|C(\theta) + \sigma^2 I| - \frac{N}{2}\log 2\pi, \qquad (5.1)$$

where the observations $Y \sim \mathcal{N}(0, C + \sigma^2 I)$ and C denotes the matrix of covariances among the training inputs. $\theta = \{v_0, \omega_k, \sigma_n^2\}$ are the hyperparameters defined in equation 2.14.

In our work, we study the user matching in these contexts: (1) the subtle hand movement, (2) with the device in the user's trouser pocket and (3) walking with the device held in the hand. The maximum likelihood estimation of the hyperparameters θ gives us a model, based on which we can compute and compare the log-likelihoods given different combinations of position and acceleration observations. The joint likelihood represents the joint probability

density function (pdf) evaluated at the joint vector under the model. The higher the likelihood is, the more likely the combination is, that is, the more likely this position signal is associated with the acceleration signal.

In order to find the joint distribution of the low-sampling-rate position P_{low} and the high-sampling-rate acceleration Acc_{high}, we apply the GP prior method and calculate an overall covariance matrix C_{all}, so

$$\begin{bmatrix} P_{low} \\ Acc_{high} \end{bmatrix} \sim \mathcal{N}(0, C_{all}) . \tag{5.2}$$

In the following section, we will apply GPs in a sensor fusion manner and find this joint distribution of position and acceleration with the GP prior method. Following this, we give a detailed description on how to fuse the Kinect sensor and mobile inertial sensors for calculating the log-likelihood of the low-sampling-rate position and the high-sampling-rate acceleration. Finally, we illustrate how to match a particular position signal with time-series of acceleration measurements through log-likelihood comparison.

5.3.2 Multi-rate Sensor Fusion for User Matching

In this section, we apply the novel and improved GP prior model presented in Chapter 4 for multi-rate sensor fusion. We emphasize the log-likelihood of GP prior model and describe how to calculate this log-likelihood given different combinations of the low-sampling-rate position measurements sensed by the Kinect and the high-sampling-rate accelerations measured by the mobile inertial sensors. We discuss how to determine a user matching result by comparing the log-likelihoods of pairing a specific position signal with all the acceleration signals sensed by multiple mobile devices.

Now we introduce how to fuse multiple observations from different sensors, e.g. the position sensor (Kinect) and the acceleration sensor (the built-in inertial sensors in a mobile device). We describe how to fuse the Kinect position and the acceleration sensed by the mobile inertial sensors with the proposed GP prior model, which incorporates the low-sampling-rate position measurements and the high-sampling-rate acceleration measurements, and takes the uncertainty of different sensors into account.

Multi-rate Sensor Fusion

As discussed in section 4.3.2, we found the joint distribution of the low-sampling-rate position and the high-sampling-rate acceleration, as defined in equation 4.21. Given the observations, i.e. the combination of position and acceleration measurements, we can apply the GP

sensor fusion model for calculating the log marginal likelihood

$$\log L = -\frac{1}{2}\log\left|K_{all}\Sigma {K_{all}}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right| - \frac{1}{2}{M_{all}}^T\left(K_{all}\Sigma {K_{all}}^T + \begin{bmatrix} \Sigma_p & \\ & \Sigma_a \end{bmatrix}\right)^{-1} M_{all} - \frac{1}{2}N\log 2\pi. \quad (5.3)$$

In equation (5.3), the time instants when the inertial sensor samples are acquired are used as the training inputs $(N \times 1)$. This gives the covariance matrix Σ. M_{all} represents the overall targets, which include the low-sampling-rate positions and the high-sampling-rate accelerations. $M_{all} = \begin{bmatrix} M_p \\ M_a \end{bmatrix} = [y_1, \cdots, y_n, a_1, \cdots, a_N]^T$. The Σ_p and Σ_a represent the diagonal matrices of position and acceleration observation variances respectively. The uncertainty of measurements can be found in section 3.3.1. The GP hyperparameters are set to $v_0 = 4.69 \times 10^3$, $\omega_1 = 7.85$, $\sigma_y^2 = 64$ and $\sigma_a^2 = 100^2$.

Under this model, we calculate and compare the joint log-likelihoods given different combinations of positions and accelerations, and match a specific user's skeleton with one of the mobile devices.

Log-Likelihood Comparison for User Matching

Now we illustrate how to use equation (5.3) for associating the position with the acceleration. For example, there are 2 users and each carries a mobile device. There are 2 position sequences denoted as p_1 and p_2, and 2 acceleration sequences denoted as a_1 and a_2. Thus, there are 4 combinations, i.e. $\begin{bmatrix} p_1 & a_1 \end{bmatrix}^T$, $\begin{bmatrix} p_1 & a_2 \end{bmatrix}^T$, $\begin{bmatrix} p_2 & a_1 \end{bmatrix}^T$ and $\begin{bmatrix} p_2 & a_2 \end{bmatrix}^T$. Our goal is to find the acceleration match for p_1 and p_2, respectively.

For user 1, we denote the two combinations as Y_{11} and Y_{12}. That is, $Y_{11} = \begin{bmatrix} p_1 \\ a_1 \end{bmatrix}$ and $Y_{12} = \begin{bmatrix} p_1 \\ a_2 \end{bmatrix}$. By applying equation (5.1), we can evaluate the joint log-likelihood $\log L_{11}$ given the combination Y_{11}, and $\log L_{12}$ given Y_{12}, respectively. If $\log L_{11} > \log L_{12}$, it is more likely that p_1 is associated with a_1. If $\log L_{11} < \log L_{12}$, it is more likely that p_1 is associated with a_2.

$$\log L_{11} = -\frac{1}{2}\begin{bmatrix} p_1 \\ a_1 \end{bmatrix}^T (C_{all})^{-1} \begin{bmatrix} p_1 \\ a_1 \end{bmatrix} - \frac{1}{2}\log|C_{all}| - \frac{N}{2}\log 2\pi \quad (5.4)$$

$$\log L_{12} = -\frac{1}{2}\begin{bmatrix} p_1 \\ a_2 \end{bmatrix}^T (C_{all})^{-1} \begin{bmatrix} p_1 \\ a_2 \end{bmatrix} - \frac{1}{2}\log|C_{all}| - \frac{N}{2}\log 2\pi \quad (5.5)$$

For the situation when we have n users, there will be n combinations for each user. The highest log-likelihood indicates the best match.

5.4 User Matching System Overview

The equipment consists of the Microsoft Kinect and a SHAKE SK7. This has been described in section 3.3. A detailed description of the coordinate systems involved in this sensor fusion has been presented in section 3.3.2.

The joint positions are sensed by the Kinect. Meanwhile, the inertial sensors communicate with the computer via Bluetooth and the inertial sensor data are fused for estimating the hand acceleration. The Kinect positions and the acceleration measurements are fused with the proposed Gaussian Process prior model. The sensor fusion system compares the joint log-likelihoods given a position signal from a particular user and multiple acceleration signals sensed by the mobile devices, and gives the highest joint log-likelihood result, which indicates the matching of this positional signal and the acceleration signal. The system finds the matched acceleration sequence for each skeleton position sequence. In this way, we associate the motion sensed by the Kinect with the motion sensed by the mobile inertial sensors. Thus, the user matching is achieved.

User Matching through the Joint Log-Likelihood Comparison

In order to match a specific user's skeleton with a mobile device, we use the proposed Gaussian Process prior model to calculate the joint log-likelihood of the Kinect position measurements and the accelerations measured by mobile inertial sensors. The joint log-likelihood function defined in equation (5.3) can be used to associate a position signal with an acceleration signal. How likely is a time-series of acceleration signals associated with measurements of position signals? We estimate and compare all the log-likelihoods of pairing one specific position signal with all other acceleration signals, and find the best match. We do this for all the skeleton position signals, respectively, allowing us to infer a user match.

5.5 Simulation Experiment: Estimation of Position, Velocity and Acceleration with GP Priors

In this simulation experiment, we used the transformed GP priors to estimate the position, the velocity and the acceleration. We applied the transformed GP prior method to estimate

the position, the velocity and the acceleration based on the simulation (position) data. The real position signal is

$$y = 10 \times \sin(x), \tag{5.6}$$

where $x = 0 : \frac{\pi}{20} : 2\pi$. We set the position measurements equal to the real signal plus Gaussian white noise (the variance σ^2). Two noise levels include (1) $\sigma^2 = 1$ and (2) $\sigma^2 = 25$.

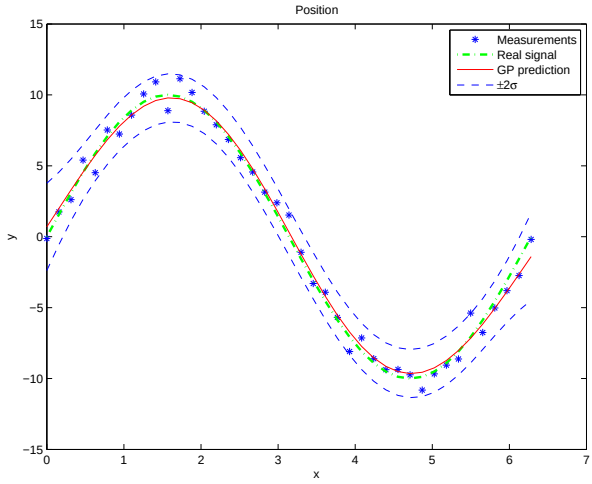
(1) Noise level $\sigma^2 = 1$

Figure 5.1a illustrates the noisy position measurements, the real position signal and the GP predictive positions. We can see that the GP prior method gives good estimation of positions.

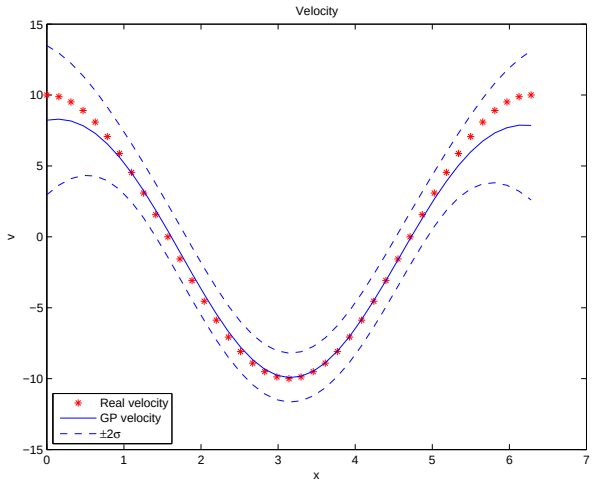
In addition to the position, we also used the transformed GP prior method to predict the velocity and the acceleration. Figure 5.1b illustrates the predictive velocity results with GPs. Figure 5.1c shows the predictive acceleration with GPs.

(2) Noise level $\sigma^2 = 25$

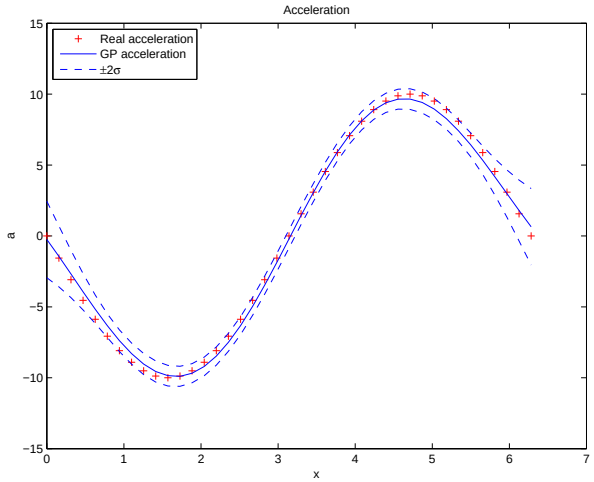
When we increase the noise in the measurement signal, we get the results, including the position (Figure 5.2a), the velocity (Figure 5.2b) and the acceleration (Figure 5.2c).



(a) Plots show the noisy position measurements, the real position signal, and the position prediction with the transformed GP prior method.

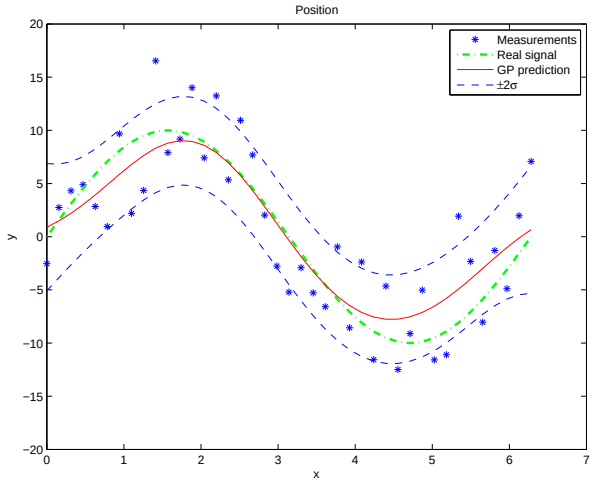


(b) Comparison of velocity: the real velocity and the velocity estimated with the transformed GP prior method.

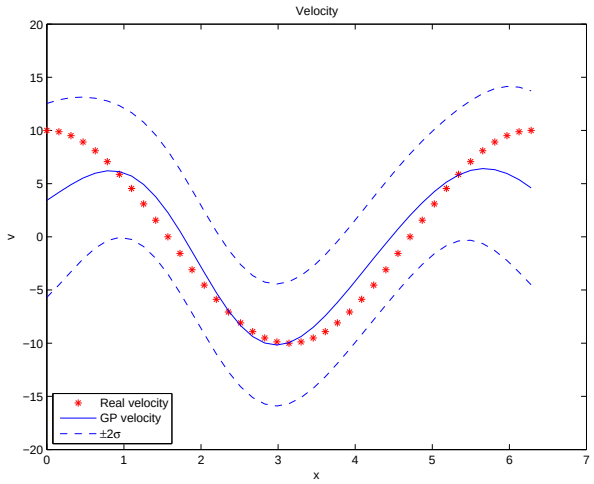


(c) Comparison of acceleration: the real acceleration and the acceleration estimated with the transformed GP prior method.

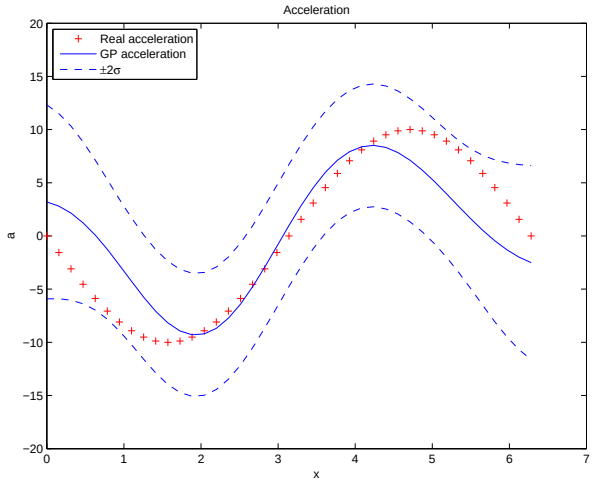
Figure 5.1: (Simulation experiment) Estimation of position, velocity and acceleration with the transformed GP priors. $\sigma^2 = 1$



(a) Plots show the noisy position measurements, the real position signal, and the position prediction with the transformed GP prior method.



(b) Comparison of velocity: the real velocity and the velocity estimated with the transformed GP prior method.



(c) Comparison of acceleration: the real acceleration and the acceleration estimated with the transformed GP prior method.

Figure 5.2: (Simulation experiment) Estimation of position, velocity and acceleration with the transformed GP priors. $\sigma^2 = 25$

Conclusion

In this experiment, we used the transformed GP priors to estimate the position, the velocity and the acceleration of the noisy measurements. We can see that the GP predictive results are comparable with the noise-free simulation data including the position, the velocity and the acceleration.

5.6 The User Matching Experiment I: Subtle Hand Movement

5.6.1 Experiment Design

In this experiment, we have 6 participants (2 female, 4 male, aged between 24 and 30, mean age 26). Each user performed a subtle hand movement in front of the Kinect with a mobile device (SK7 Dimensions 43 mm $\times$ 32 mm $\times$ 18 mm) held in the right hand. The user was instructed to stand there, put the hand at the side of the hip and not to move the feet during the experiment. He/She then swayed the hand naturally and subtly. We analysed the hand movement along the $x-$axis, i.e. the horizontal movement. The range of measured movement magnitude is defined as (0, max(x)-min(x)). In this way, we get the hand movement range for each user, $(0, 14.4)$cm (user 1), $(0, 15.9)$cm (user 2), $(0, 10.1)$cm (user 3), $(0, 24.8)$cm (user 4), $(0, 11.6)$cm (user 5) and $(0, 14.3)$cm (user 6). The maximum magnitude of measured movement acceleration is 2.25×10^3 mm/s^2. The whole process took 10s.

For each participant, we gathered 1 sample. We collected the participant's skeleton tracking positions, i.e. the right hand position measurements, and the hand motion data sensed by mobile inertial sensors. The hand position measurements along the $x-$axis for the 6 participants are shown in the left panel of Figure 5.5. For the 6 participants, the hand acceleration measured by mobile inertial sensors are shown in the right panel of Figure 5.5.

5.6.2 Experimental Results

Kinect Sensor Noise

When the user moves the hand slightly, the Kinect cannot sense the changes of hand positions due to the noise, as the noise occludes the real slightly-changed hand positions. Before we started the subtle hand movement experiment, we asked one of the participants to perform a subtle hand movement.

Position Sensing We recorded the hand positions sensed by the Kinect and the accelerations measured by the inertial sensors. The whole process took 15s, including 3 time periods: (1) Hand–motionless (2) Hand–subtle movement (3) Hand–motionless. The hand position tracking result is shown in Figure 5.3. It can be seen that the Kinect cannot sense the changes of hand position correctly due to the sensor noise and the subtle hand movement.

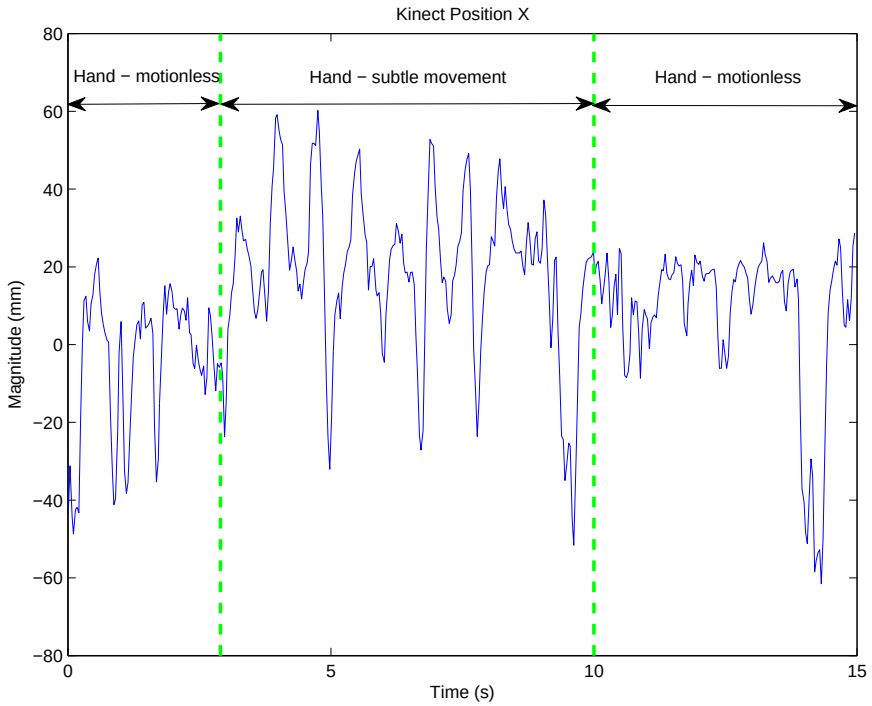


Figure 5.3: (Subtle hand movement) The hand position (along the $x-$axis) measured by the Kinect sensor. The changing of the hand position measurements during the 3 time periods ((1) Hand–motionless (2) Hand–subtle movement (3) Hand–motionless) is illustrated. It can be seen that the hand positions cannot be determined correctly due to the Kinect sensor noise and the subtle hand movement.

Acceleration Sensing Although the hand position cannot be determined in Figure 5.3, we still have the accelerations measured by the mobile inertial sensors. The result is shown in Figure 5.4. It can be seen that the hand motion data sensed by the inertial sensors are less noisy in comparison with the very noisy Kinect position measurements shown in Figure 5.3.

Subtle Hand Movement Experiment

After illustrating the limitation of the Kinect sensor, we conducted the subtle hand movement experiment. The 6 participants were asked to perform subtle hand movements with the SK7

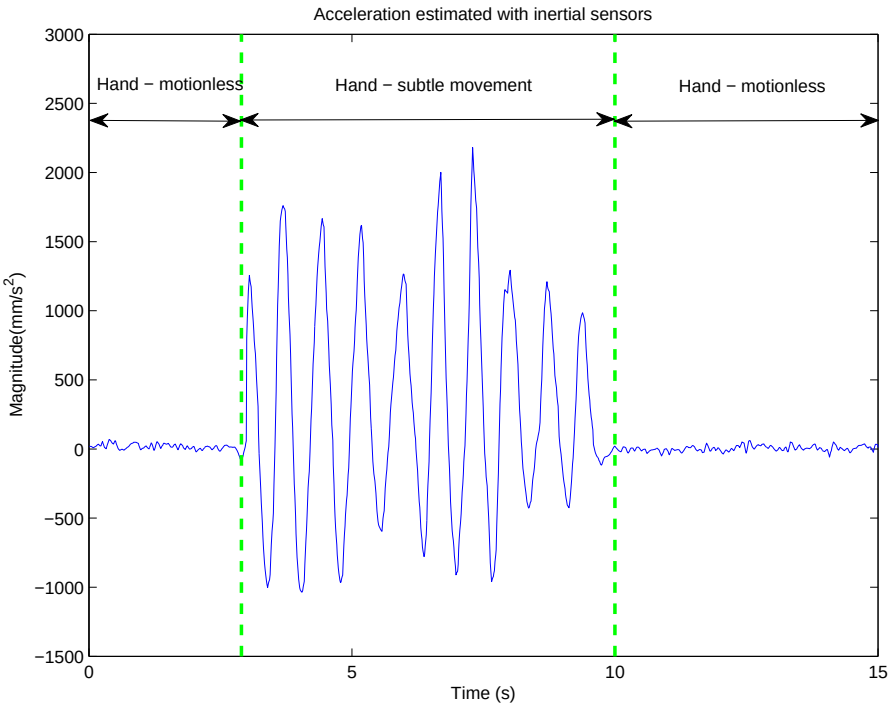


Figure 5.4: (Subtle hand movement) The hand acceleration (along the $x-$axis) sensed by the inertial sensors. Although the hand positions cannot be correctly sensed by the Kinect (Figure 5.3), the hand accelerations can be measured with the inertial sensors. The changing of the accelerations during the different time periods is illustrated. We can see the difference between the accelerations during different time periods: (1) Hand–motionless (2) Hand–subtle movement (3) Hand–motionless.

held in the right hand when they stood in front of the Kinect. Figure 5.5 illustrates the position observations and the acceleration measurements for all 6 participants.

Comparison with the State-of-the-art Work In order to compare our method with the ShakeID method (Rofouei et al., 2012), we associated the motion sensed by the Kinect with the motion sensed by the mobile inertial sensors for user 1 and user 2 with the direct acceleration comparison method.

In order to simulate the acceleration distance comparison method proposed in the literature (Rofouei et al., 2012), we applied a Kalman filter for estimating the hand acceleration based on the Kinect positions. No detailed parameters including the process covariance matrix Q and the measurement covariance matrix R were given in (Rofouei et al., 2012). Here we set the state vector

$$x_k = \begin{bmatrix} p_{xk} & p_{yk} & p_{zk} & v_{xk} & v_{yk} & v_{zk} & a_{xk} & a_{yk} & a_{zk} \end{bmatrix}. \tag{5.7}$$

The measurements include $3-$axis positions. We estimated the $3-$axis accelerations of the

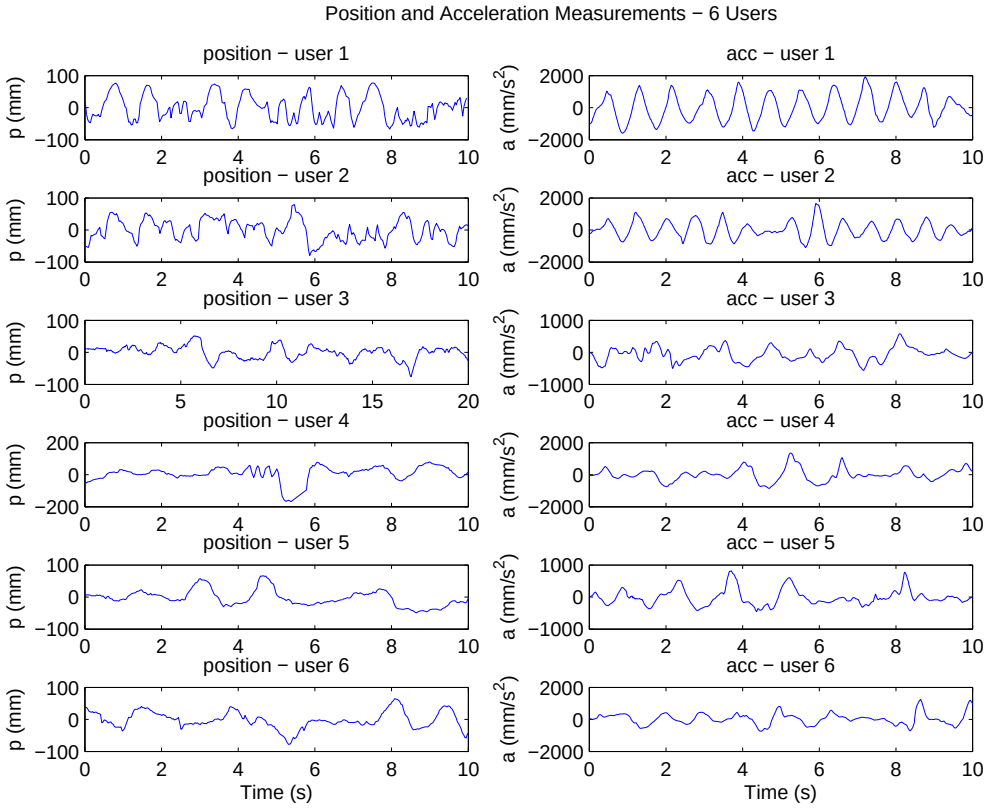


Figure 5.5: (Experiment 1: Subtle hand movement) Left column: the hand position X measurements (all 6 users). Right column: the acceleration X measurements sensed by the mobile device (all 6 users).

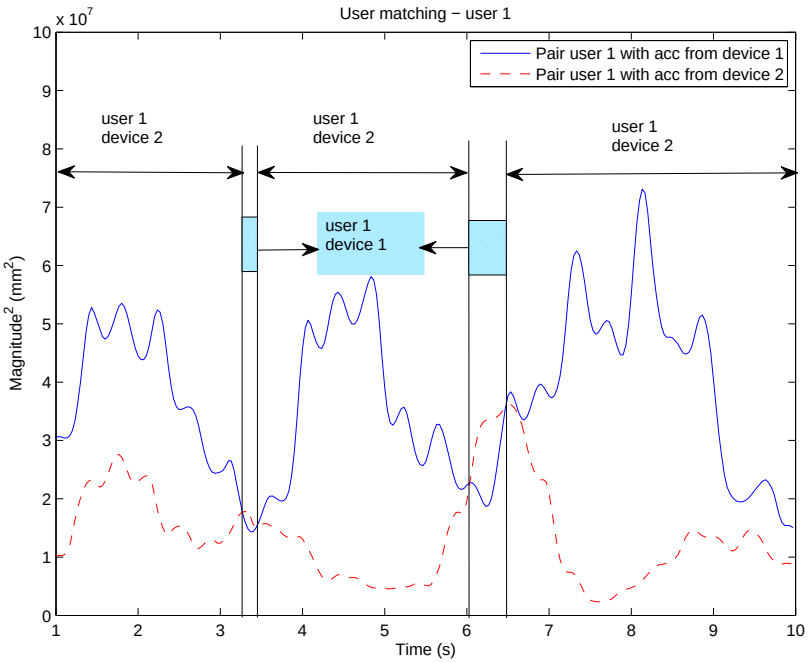


Figure 5.6: (Experiment 1: Subtle hand movement) Simulation of ShakeID method. The user 1 matching results are illustrated. The two shade areas (during $3-4$s and $6-7$s), which represent the time periods of correct match, are clearly illustrated. It can be seen that the direct acceleration comparison method gave the wrong results during the periods in $1-3$s, $4-6$s and $7-10$s respectively.

Kinect positions for user 1 and user 2 respectively. The $3-$axis acceleration estimated with the inertial sensors in the mobile device was compared with the $3-$axis acceleration estimated with a Kalman filter based on Kinect positions over a time window (1s). The squared Euclidean distance D was calculated.

$$D = \arg\min \sum_{t-1}^{t} \left(\|a_x - a_{x,device}\|^2 + \|a_y - a_{y,device}\|^2 + \|a_z - a_{z,device}\|^2 \right), \tag{5.8}$$

where a_x, a_y and a_z represent the $3-$axis acceleration of the Kinect positions in the 1s time window. $a_{x,device}$, $a_{y,device}$ and $a_{z,device}$ denote the $3-$axis ($x-$axis, $y-$axis, $z-$axis) acceleration measured by mobile inertial sensors, respectively. For identifying the user who held the device 1, we compared the acceleration of Kinect position (user 1) with the two acceleration signals measured by the devices 1 and 2. In Figure 5.6, we show the squared distance between the acceleration of Kinect position (user 1) and the acceleration measured by device 1 and device 2, respectively. The ground truth is that user 1 was associated with device 1. It can be seen that the ShakeID method does not give the correct pairing result. For user and device association for user 2, we show the results in Figure 5.7. The ground truth is that user 2 was holding device 2. We can see that the ShakeID method does not work for the subtle hand movement situation.

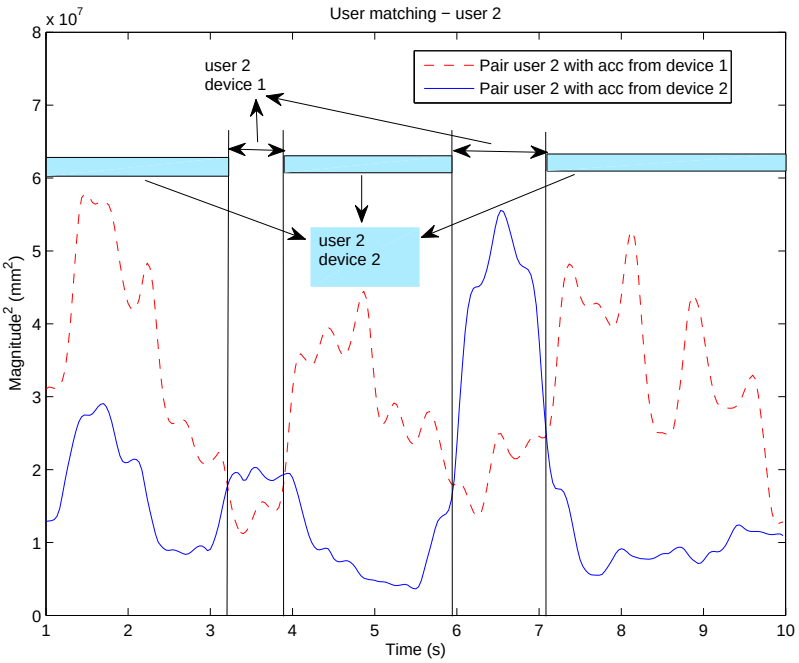


Figure 5.7: (Experiment 1: Subtle hand movement) Simulation of ShakeID method. The user 2 matching results are illustrated. The time periods when user 2 was matched with device 2 are denoted as shade areas. It can be seen that the direct acceleration comparison method gave the wrong results during the periods in $3-4$s and $6-7$s respectively.

User Matching with Sensor Fusion Method In the above steps, we show that the ShakeID method does not work for matching the motion sensed by the Kinect and the motion sensed by inertial sensors for user 1 and user 2, respectively, in the subtle movement situation.

Now we apply the GPs user matching algorithm for associating the skeletons sensed by the Kinect with the accelerations measured by the inertial sensors for 6 users. The position measurements and the accelerations sensed by the inertial sensors are shown in Figure 5.5. The following work is to find the matching relationship between the 6 position signals and the 6 acceleration signals. With the proposed method, we built a GP prior model, under which we want to compare the log-likelihoods given different combinations of the low-sampling-rate positions and high-sampling-rate accelerations.

Given a combination of positions and accelerations, i.e. M_{all}, the log-likelihood is calculated according to equation (5.3). In order to find the best acceleration match for the position from user 1, we have $M_{ij} = \begin{bmatrix} p_i & a_j \end{bmatrix}^T, \forall i = 1, j = 1, \cdots, 6$. By comparing $\log L_{ij}, \forall i = 1, j = 1, \cdots, 6$, we can find the highest joint log-likelihood. The joint log-likelihood comparison results for user 1 are shown in Figure 5.8. It indicates that the log-likelihood of associating the position of user 1 with the accelerations of device 1 is the highest. Thus, we identify that user 1 was carrying device 1. This is the correct matching result. In a similar manner, we find the match between the position and the acceleration for other 5 users respectively. The

joint log-likelihood comparison results are shown in Figure 5.9, Figure 5.10, Figure 5.11, Figure 5.12 and Figure 5.13, respectively. We summarise the matching results in Table 5.1. The row "user i" ($i = 1, \cdots, 6$) represents the matching results for user i. The column j ($j = 1, \cdots, 10$) denotes the time instant. Thus, user 1 was matched with device 4 at time instant 1s. The ground truth data is that user k was associated with the mobile device k ($k = 1, \cdots, 6$). In Table 5.1, we can see that the matching time for user $1, \cdots, 6$ are 4s, 1s, 3s, 4s, 1s, 2s, respectively.

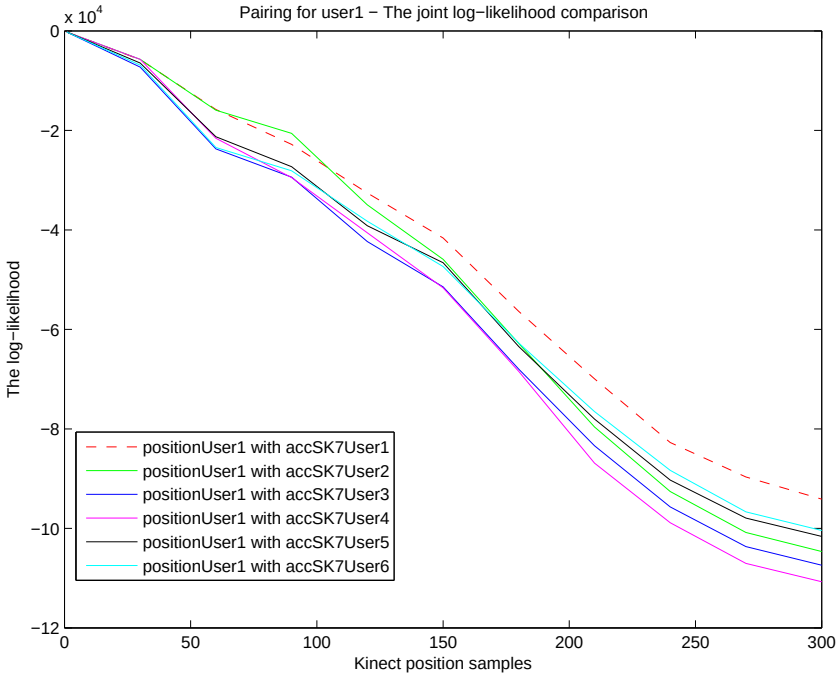


Figure 5.8: (Experiment 1: Subtle hand movement) User 1 was associated with the mobile device 1. The curves represent the joint log-likelihoods given position measurements from user 1 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

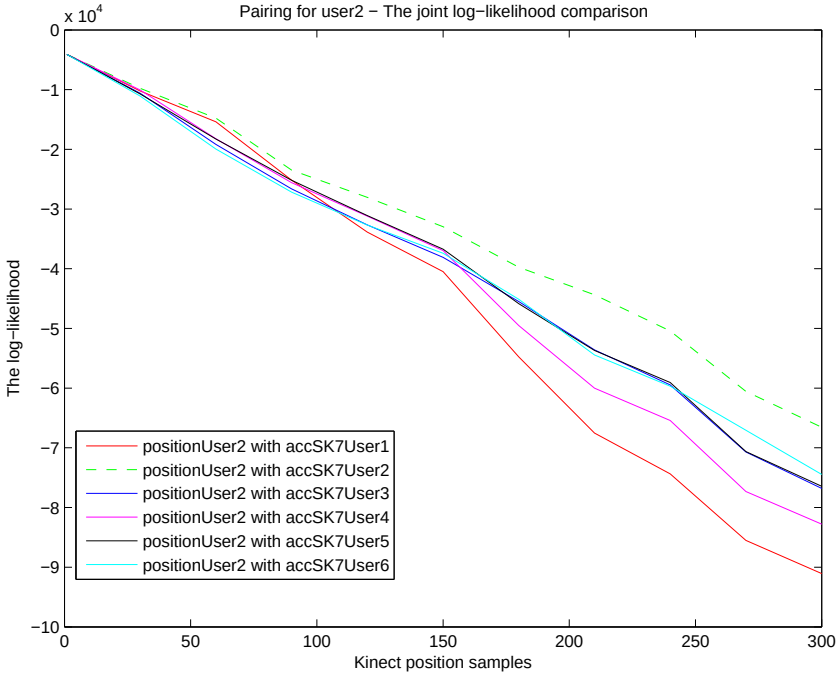


Figure 5.9: (Experiment 1: Subtle hand movement) User 2 was associated with the mobile device 2. The curves represent the joint log-likelihoods given position measurements from user 2 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

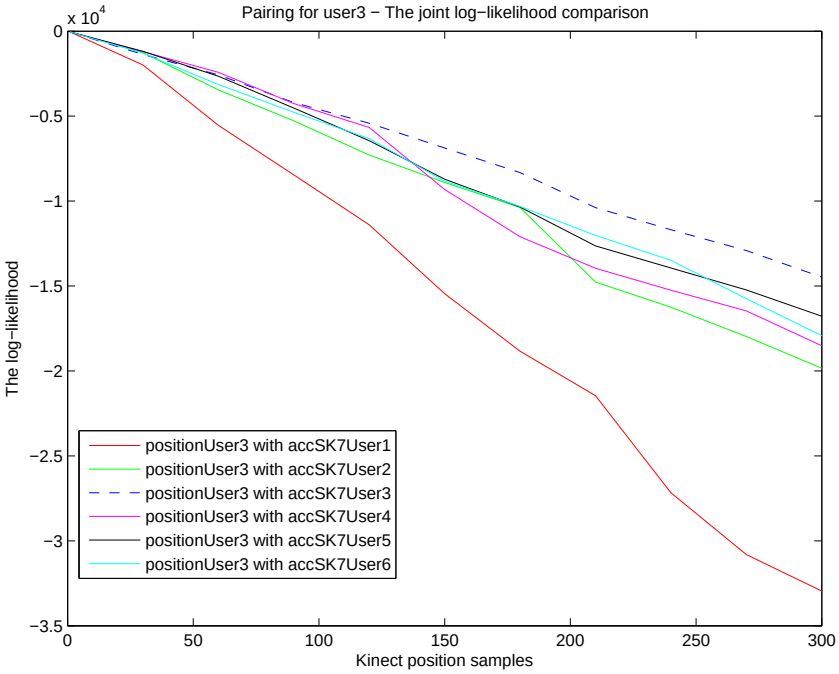


Figure 5.10: (Experiment 1: Subtle hand movement) User 3 was associated with the mobile device 3. The curves represent the joint log-likelihoods given position measurements from user 3 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

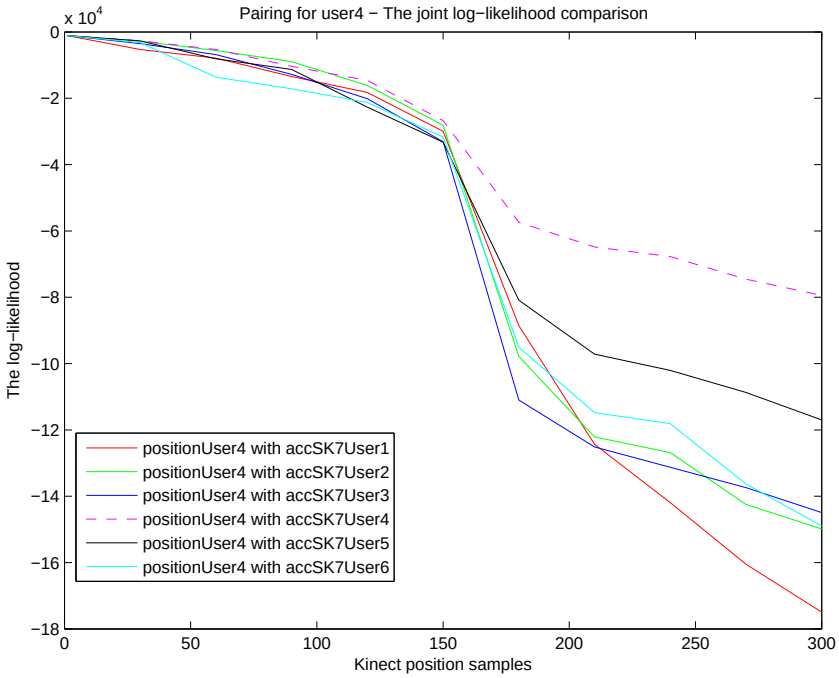


Figure 5.11: (Experiment 1: Subtle hand movement) User 4 was associated with the mobile device 4. The curves represent the joint log-likelihoods given position measurements from user 4 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

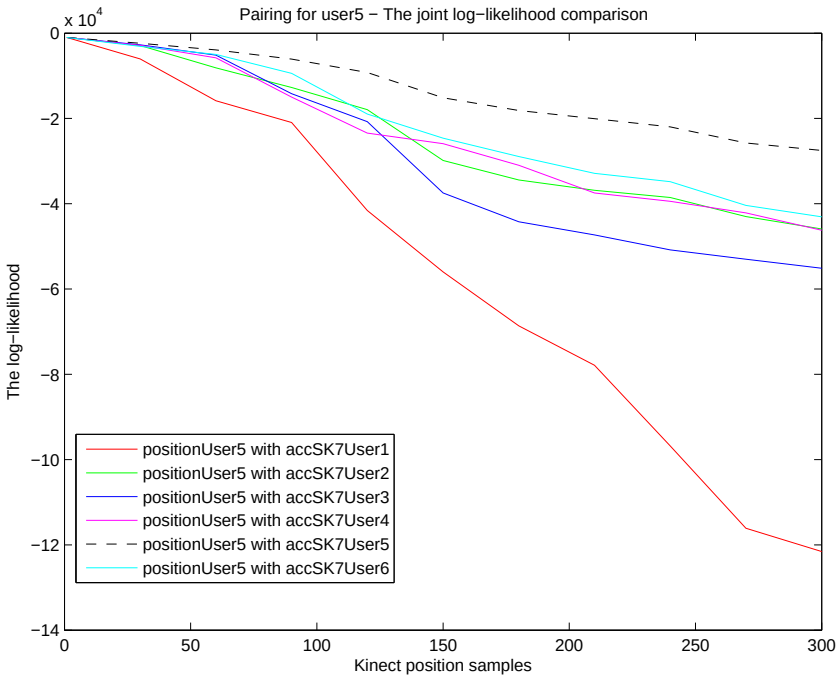


Figure 5.12: (Experiment 1: Subtle hand movement) User 5 was associated with the mobile device 5. The curves represent the joint log-likelihoods given position measurements from user 5 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

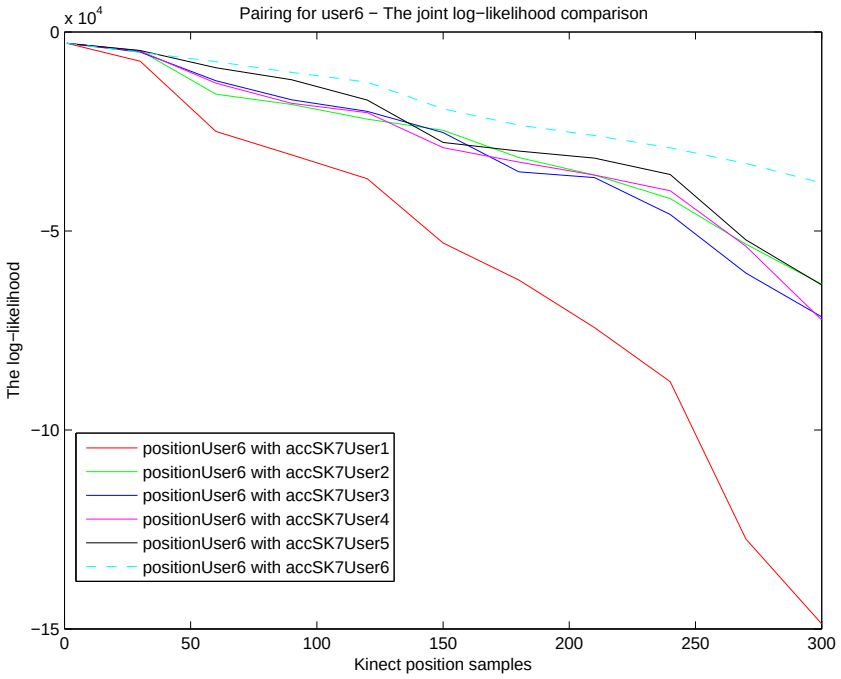


Figure 5.13: (Experiment 1: Subtle hand movement) User 6 was associated with the mobile device 6. The curves represent the joint log-likelihoods given position measurements from user 6 and 6 acceleration signals sensed by the inertial sensors. The estimate of best match is indicated by the highest curve.

Here we also compare our method with the direct acceleration comparison method. We directly compare the acceleration vectors for user matching by calculating the sum of squared Euclidean distance between the acceleration of the Kinect positions and the acceleration measured by mobile inertial sensors to determine the matched device for a particular user.

$$d = \arg\min \sum_{0}^{t} \|a_{knt} - a_{i,device}\|^2, \tag{5.9}$$

where a_{knt} denotes the acceleration of the Kinect positions from a particular user and $a_{i,device}$ denotes the acceleration measured by the device ($i = 1, \cdots, 6$). The matched device gives the minimum sum of squared distance. In this experiment, we used the acceleration along the $x-$axis. We found that this acceleration comparison method does not work either in the subtle hand movement situation. We summarise the user matching results in Table 5.2 and the comparison results in Table 5.3. Thus, the direct acceleration distance comparison method is not applicable for associating the hand accelerations of Kinect positions with the accelerations estimated with the mobile inertial sensors in the subtle hand movement situation.

Table 5.1: (Experiment 1: Subtle hand movement) The user matching results using GP sensor fusion method.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	4	1	2	1	1	1	1	1	1	1
user 2	2	2	2	2	2	2	2	2	2	2
user 3	5	4	3	3	3	3	3	3	3	3
user 4	4	2	2	4	4	4	4	4	4	4
user 5	5	5	5	5	5	5	5	5	5	5
user 6	5	6	6	6	6	6	6	6	6	6

Table 5.2: (Experiment 1: Subtle hand movement) The user matching results using acceleration comparison method. This method does not give the correct results.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	3	3	4	3	4	4	4	3	3	3
user 2	4	4	4	4	4	4	4	4	4	4
user 3	5	5	3	5	5	5	5	5	5	5
user 4	6	6	6	3	3	3	3	3	3	3
user 5	4	5	4	4	4	3	3	3	3	3
user 6	5	5	5	4	3	3	3	3	3	3

Table 5.3: The user matching results for experiment 1 – Comparison of GP sensor fusion method and acceleration comparison method. Y denotes "Yes"(matched). N denotes "No"(matching failure).

		User ID						Total Matched No.
		1	2	3	4	5	6	
Subtle movement	GP sensor fusion	Y	Y	Y	Y	Y	Y	6
	Acceleration comparison	N	N	N	N	N	N	0

5.6.3 Conclusion

In this experiment, we investigated the performance of the proposed GP prior model for user matching in a subtle hand movement experiment. We compared our method with the state-of-the-art work in the literature and conclude that our method can achieve user matching in seconds ($1-4$s in this experiment) while the direct acceleration comparison method fails to find a match in the subtle hand movement experiment.

5.7 The User Matching Experiment II: Mobile Device in User's Trouser Pocket

5.7.1 Experiment Design

In this experiment, we studied the usability of our sensor fusion system when the users put their mobile devices in the trouser pockets. We used the same 6 users, who participated in the experiment 1. We asked each participant to put the mobile device in his/her own trouser pocket. At the beginning, the participant put the mobile device (SK7 Dimensions 43 mm $\times$ 32 mm $\times$ 18 mm) in the right trouser pocket and stood at a distance of approximately 3m from the Kinect. The participant faced the Kinect and started walking toward the Kinect and kept moving in an area of 3m $\times$ 1.5m in the Kinect field of view. The whole process took 10s. We analysed the position and acceleration measurements along the $y-$axis (the vertical axis). We did not define a moving trajectory and did not limit the user's movement as this was more realistic. As long as the two joints, including the right knee and the right hip, are moving, the Kinect can infer the vertical movement of the pocket and the mobile device can sense the acceleration along the vertical axis, the movement along the other axes will not influence the vertical movement. Thus, how the user moves along the $x-$axis and $z-$axis does not influence the user matching as we only analyse the inferred pocket movement along the $y-$axis.

For each participant, we gathered 1 sample. We collected the participant's skeleton tracking positions, i.e. the right hip and the right knee position measurements, and the hand motion data sensed by the inertial sensors embedded in the mobile device.

The Kinect cannot detect the position of the trouser pocket exactly, as the sensor can only detect the skeleton joints positions. One way to infer the pocket position is to calculate the weighted sum of relevant joints positions,

$$\hat{p} = \sum_{i=1}^{n} \omega_i p_i, \tag{5.10}$$

where n denotes the number of relevant skeleton joints. p_i denotes the position of joint i and ω_i is the corresponding weight of joint i. In this work, we assume that the trouser pocket is located at the middle point between the right hip and the right knee, as shown in Figure 5.14. That is, the position Y of the pocket is

$$Y_{pocket} = \frac{1}{2}\left(Y_{hip} + Y_{knee}\right). \tag{5.11}$$

The inferred pocket positions along the $y-$axis (vertical axis) for the 6 participants are shown

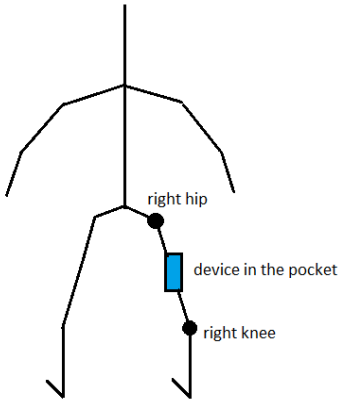


Figure 5.14: Infer the position of the trouser pocket that is located at the middle point between the right hip and the right knee.

in the left panel of Figure 5.16. The inferred pocket acceleration is the acceleration measured by mobile inertial sensors. The acceleration measurements for the 6 participants are shown in the right panel of Figure 5.16.

5.7.2 Experimental Results

For user 1, we show the relationship between the two joints (the right hip and the right knee) and the trouser pocket in Figure 5.15. The position measurements along the y-axis are analysed. The positions of the right hip (dashed line) and the positions of the right knee (dash-dot line) are shown. We inferred the positions of the trouser pocket (solid line) by computing the mean of the positions of the right hip and the positions of the right knee.

For the other 5 users, we inferred the positions of the trouser pocket in the same way. For all 6 users, the inferred positions of the trouser pockets and the acceleration measurements sensed by the mobile inertial sensors are shown in Figure 5.16. Our goal is to associate the 6 inferred position signals with the 6 acceleration signals, that is, match the pocket position with the acceleration sensed by the mobile device. In this way, we identified which user carried which mobile device.

We used our sensor fusion model for calculating the joint log-likelihood of the low-sampling-rate position measurements and the high-sampling-rate acceleration measurements. By pairing a specific position signal, e.g. the positions for user 1, with the 6 acceleration signals, we can find the best match, i.e. the joint log-likelihood of pairing the positions from user 1 with the accelerations from user 1. For all 6 users, the matching results are summarised in Table 5.4. The row "user i" ($i = 1, \cdots, 6$) represents the matching results for user i. The column j ($j = 1, \cdots, 10$) denotes the time instant. Thus, user 1 was matched with device 2 at time instant 1s. The ground truth data is that user k was associated with the mobile device

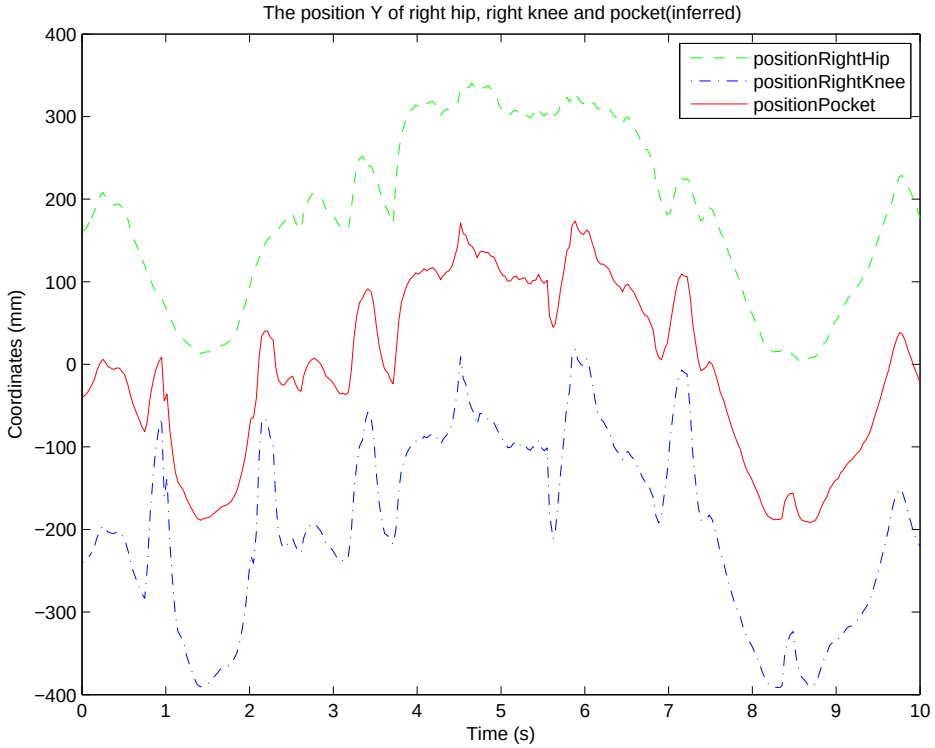


Figure 5.15: (Experiment 2: Mobile device in the trouser pocket) The position measurements Y (the vertical axis) of the right hip (dashed line) and the right knee (dash-dot line) for user 1 are shown. The inferred positions of the trouser pocket are illustrated (solid line).

k ($k = 1, \cdots, 6$). In Table 5.4, we can see that the matching time for user $1, \cdots, 6$ are 3s, 3s, 3s, 3s, 1s, 5s, respectively.

In this experiment, we also compared our approach with the direct acceleration comparison method. The results are summarised in Table 5.5. With the direct acceleration comparison method, we found that only user 4 and user 5 were successfully matched with their devices. This is due to the fact that user 4 and user 5 performed vigorous movements, which can be seen from the inferred pocket positions in Figure 5.16. Thus, the direct acceleration comparison method works only when there are vigorous inferred pocket movements. By contrast, 6 users were correctly matched with their devices with our method. Therefore, the GP fusion method outperforms the direct acceleration method in this experiment.

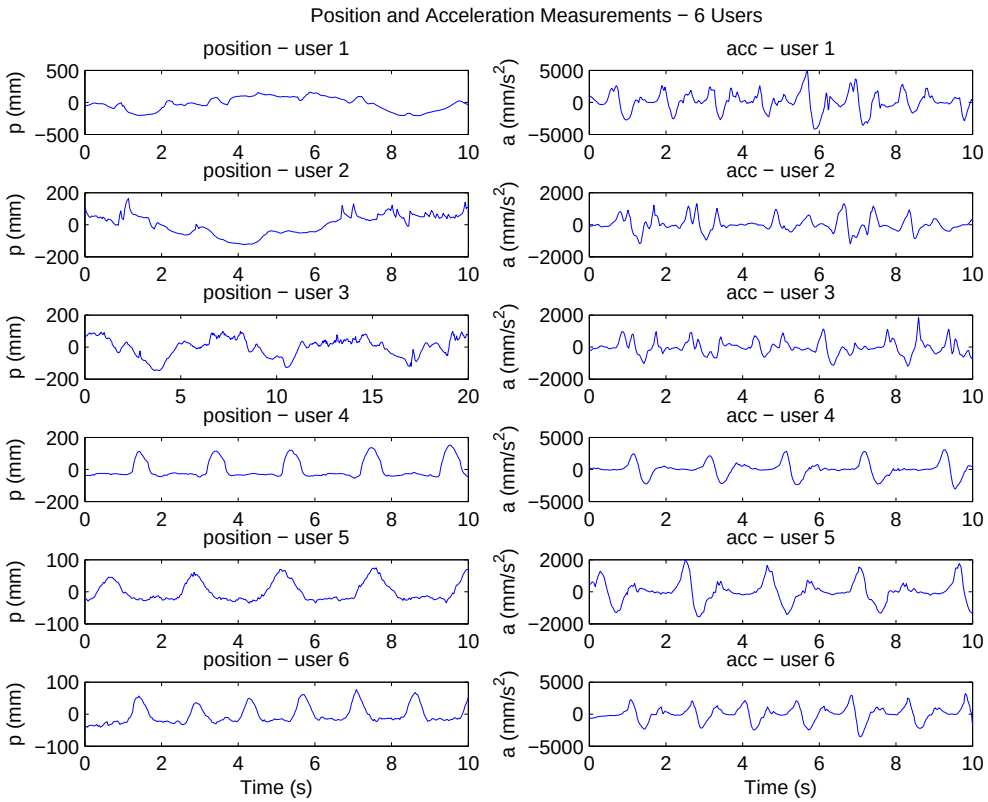


Figure 5.16: (Experiment 2: Mobile device in the trouser pocket) Left column: the position Y measurements of the inferred location of the trouser pocket (all 6 users). Right column: the acceleration Y measurements sensed by the mobile device located in the trouser pocket (all 6 users).

Table 5.4: (Experiment 2: Mobile device in the trouser pocket) The user matching results using GP sensor fusion method.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	2	5	1	1	1	1	1	1	1	1
user 2	2	1	2	2	2	2	2	2	2	2
user 3	6	2	3	3	3	3	3	3	3	3
user 4	4	6	4	4	4	4	4	4	4	4
user 5	5	5	5	5	5	5	5	5	5	5
user 6	3	3	3	3	6	6	6	6	6	6

Table 5.5: (Experiment 2: Mobile device in the trouser pocket) The user matching results using acceleration comparison method.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	3	2	2	2	2	2	2	2	2	2
user 2	6	6	3	3	3	3	2	3	3	3
user 3	4	2	3	3	3	3	2	2	2	2
user 4	4	5	4	4	4	4	4	4	4	4
user 5	5	5	5	5	5	5	5	5	5	5
user 6	4	5	3	3	3	3	3	3	3	3

Table 5.6: The user matching results for experiment 2 – Comparison of GP sensor fusion method and acceleration comparison method. Y denotes "Yes"(matched). N denotes "No"(matching failure).

		User ID						Total Matched No.
		1	2	3	4	5	6	
Device in pocket	GP sensor fusion	Y	Y	Y	Y	Y	Y	6
	Acceleration comparison	N	N	N	Y	Y	N	2

5.7.3 Conclusion

In this experiment, we tested the performance of the proposed GP prior model for user matching when the mobile device was in the user's trouser pocket. The pocket position was inferred from the Kinect skeleton joints positions. We prove that the proposed GP prior model can be used for associating the inferred positions with the accelerations sensed by the mobile inertial sensors. In comparison with the direct acceleration comparison method, which only matched 2 of the 6 users, our method is more robust and 6 users were successfully matched with their devices.

5.8 The User Matching Experiment III: Walking with Mobile Device in the Hand

5.8.1 Experiment Design

To test the user matching algorithm, a third experiment was conducted. There were 6 participants in total (1 female, 5 male). They were aged between 20 and 30 years. We captured walking samples from these experiment participants who volunteered from the academic community. In the experiment, at the beginning, the user stood at a distance of approximately 3m from the Kinect sensor with a mobile device (SK7 Dimensions 43 mm $\times$ 32 mm $\times$ 18 mm) held in the right hand. As the Kinect started tracking, the participant walked toward the Kinect and kept walking in an area of 3m $\times$ 1.5m in front of the Kinect. We measured the user's hand position along the $y-$axis (vertical axis). The whole process took 10s. We were interested in the vertical position measurements, thus we did not limit the user's movement trajectory. As long as the user is walking, there will be periodic motion pattern of the hand along the vertical axis.

For each participant, we gathered 1 walking sample. We collected the participant's skeleton tracking positions, i.e. the right hand position measurements, and the hand motion data sensed by mobile inertial sensors. The hand position measurements along the $y-$axis (vertical axis) during the walking for the 6 participants are shown in the left panel of Figure 5.20. The estimated hand accelerations during the walking for the 6 participants are shown in the right panel of Figure 5.20.

5.8.2 Experimental Results

In the experiment, we analysed the walking samples from 6 users. Firstly, user 1 is analysed as an example. We applied the transformed GP priors for estimating the position, the velocity and the acceleration. Secondly, we show all the position measurements and acceleration measurements for 6 users, and illustrate how the sensor fusion works. Finally, we compute and compare the log-likelihoods of different combinations of position and acceleration for user matching.

Prediction (Position, Velocity and Acceleration) with GPs

In section 4.3.2, we described how to estimate the derivatives, i.e. the velocity and the acceleration, based on the measurements with the transformed GP priors. Now we apply this method for system state estimation in a Kinect system. With the walking sample data from

user 1, we present the applications of the transformed GP priors, i.e. estimating the position, the velocity and the acceleration of hand movement in this experiment.

Position For user 1, the position measurements and the GP prediction results are shown in Figure 5.17. We can see that the GP method gives a good mean position prediction and uncertainty curves.

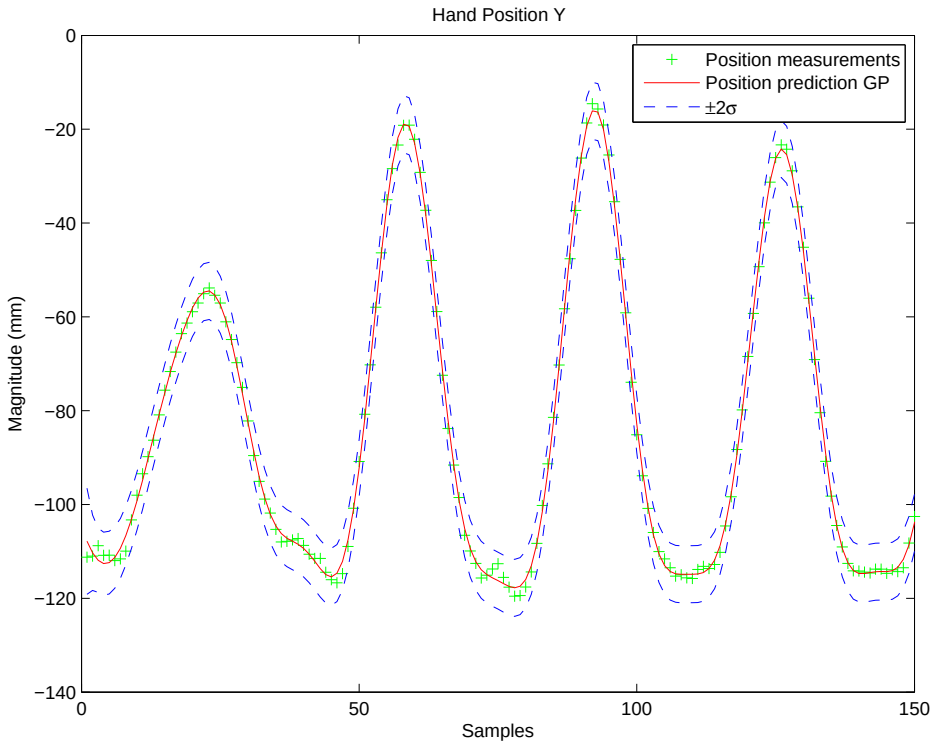


Figure 5.17: (Experiment 3: Walking with the device held in the hand – user 1) Position measurements and position prediction with GPs. Plots show mean and mean $\pm 2\sigma$ as solid line and dashed contours.

Velocity For user 1, the velocity comparison results are shown in Figure 5.18. Plots show the mean velocity prediction with GPs and the uncertainty curves (mean $\pm 2\sigma$) as solid line and dashed contours. The first derivatives of Kinect positions are also computed and shown. We can see that the noisy velocity data-points are within the GPs uncertainty curves. Thus, the transformed GP prior method is effective for estimating the first derivative of noisy position measurements, i.e. the velocity.

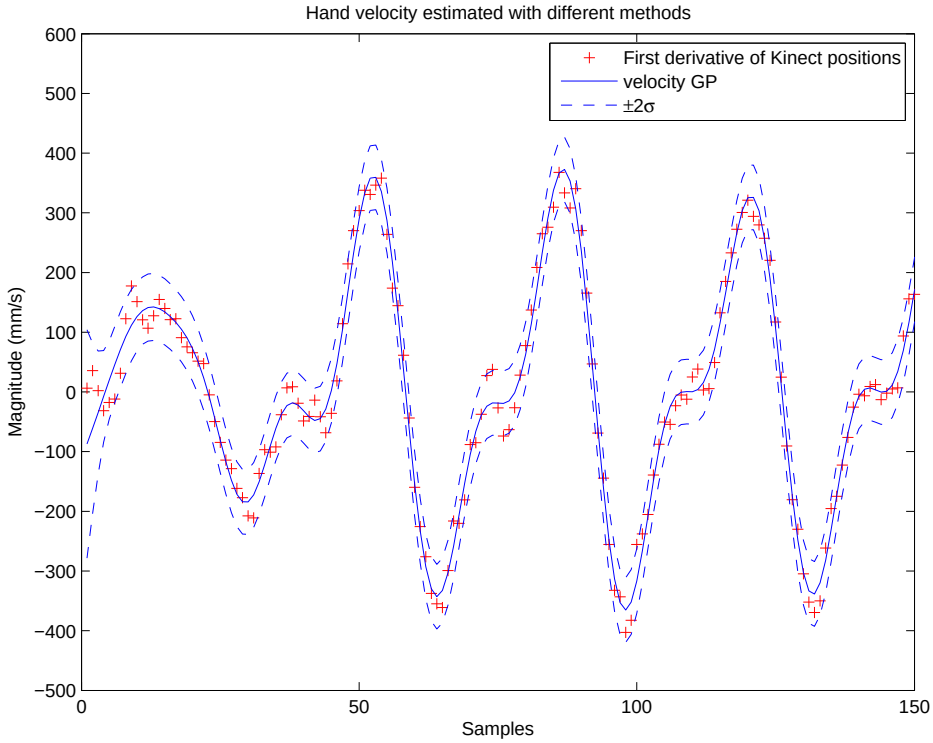


Figure 5.18: (Experiment 3: Walking with the device held in the hand – user 1) The velocity prediction with GPs. Plots show mean and mean $\pm 2\sigma$ as solid line and dashed contours. The first derivatives of Kinect positions are also shown.

Acceleration For user 1, Figure 5.19 shows all the acceleration results including the second derivatives of Kinect positions, the predictive acceleration with GPs and the acceleration sensed by the inertial sensors. It can be seen that the GPs predictive acceleration and the acceleration measured by the inertial sensors are comparable. In addition, the second derivatives of Kinect positions are within the uncertainty curves of GPs prediction. Therefore, the transformed GP prior method is useful for estimating the second derivatives of noisy position measurements, i.e. the acceleration.

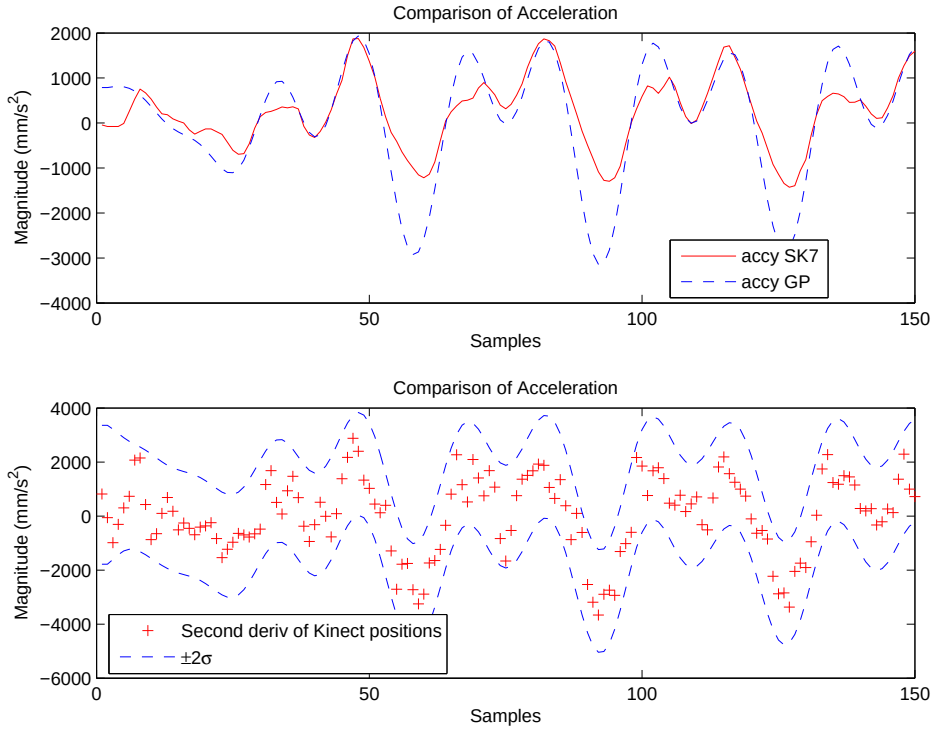


Figure 5.19: (Experiment 3: Walking with the device held in the hand – user 1) Comparison of acceleration estimated with different methods. Upper panel: the predictive acceleration with GPs (dashed line), the acceleration estimated with the inertial sensors (solid line). Lower panel: the second derivatives of Kinect positions and the GP uncertainty contours (mean $\pm 2\sigma$).

Conclusion In this part, we applied the transformed GP prior method for estimating the position, the velocity and the acceleration. Experimental results show good estimation results including the position, the velocity and the acceleration. Thus, the transformed GP priors are very useful for estimating the derivatives of noisy measurements.

Sensor Fusion with Gaussian Process Model

Now we associate the skeleton data sensed by the Kinect with the motion data sensed by the mobile inertial sensors. The hand position and the acceleration measurements for 6 users are shown in Figure 5.20.

In order to illustrate how the user matching algorithm works, we denote the position measurements for the 6 users as $p1$, $p2$, $p3$, $p4$, $p5$ and $p6$, respectively. The accelerations measured with the inertial sensors are denoted as $a1$, $a2$, $a3$, $a4$, $a5$ and $a6$, respectively. In GPs, we construct a target vector by combining the position and the acceleration. For user 1, the targets include $\begin{bmatrix} p_1 \\ a_i \end{bmatrix}, \forall i = 1, \cdots, 6.$

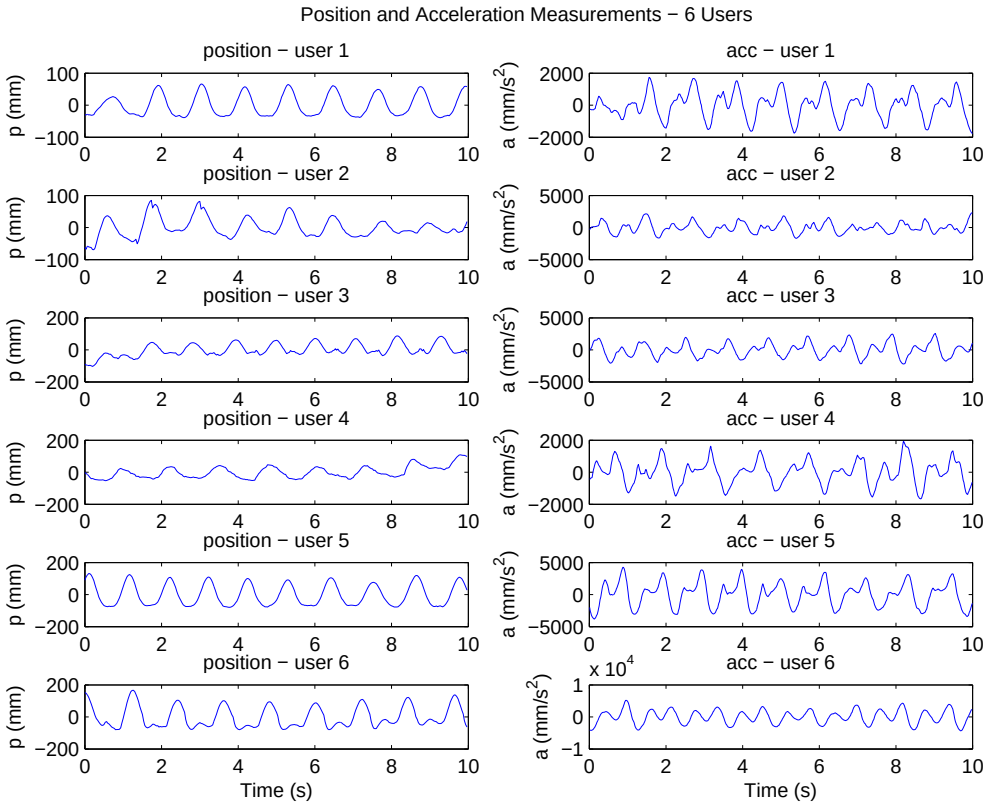


Figure 5.20: (Experiment 3: Walking with the device held in the hand) Left column: the hand position Y measurements (all 6 users). Right column: the acceleration Y measurements sensed by the mobile device (all 6 users).

Since we have all the position and the acceleration observations, we can apply our GP model for calculating the joint log-likelihoods given different combinations of positions and accelerations.

Comparison of Log-Likelihoods for User Matching

The matching results for all 6 users are summarised in Table 5.7. The row "user i" ($i = 1, \cdots, 6$) represents the matching results for user i. The column j ($j = 1, \cdots, 10$) denotes the time instant. Thus, user 1 was matched with device 1 at time instant 1s. The ground truth data is that user k was associated with the mobile device k ($k = 1, \cdots, 6$). In Table 5.7, we can see that the matching time for user $1, \cdots, 6$ are 1s, 3s, 1s, 1s, 1s, 1s, respectively.

Table 5.7: (Experiment 3: Walking with the device held in the hand) The user matching results using GP sensor fusion method.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	1	1	1	1	1	1	1	1	1	1
user 2	3	3	2	2	2	2	2	2	2	2
user 3	3	3	3	3	3	3	3	3	3	3
user 4	4	4	4	4	4	4	4	4	4	4
user 5	5	5	5	5	5	5	5	5	5	5
user 6	6	6	6	6	6	6	6	6	6	6

Table 5.8: (Experiment 3: Walking with the device held in the hand) The user matching results using acceleration comparison method.

Users	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
user 1	1	1	1	1	1	1	1	1	1	1
user 2	2	2	2	2	2	2	2	2	2	2
user 3	2	2	2	2	2	3	3	3	3	3
user 4	4	4	4	4	4	4	4	4	4	4
user 5	4	6	4	4	5	5	5	5	5	5
user 6	6	6	6	6	6	6	6	6	6	6

In this experiment, we also compared our method with the direct acceleration comparison method, which gave the user matching results illustrated in Table 5.8. We found that both methods gave the correct matching results, which are shown in Table 5.9. In the walking experiment, the device was held in the hand. The periodic movement of the hand provides a good estimation of hand acceleration. We concluded that both methods work in the situations when the users perform vigorous and visible movements.

Table 5.9: The user matching results for experiment 3 – Comparison of GP sensor fusion method and acceleration comparison method. Y denotes "Yes"(matched). N denotes "No"(matching failure).

		User ID						Total Matched No.
		1	2	3	4	5	6	
Walking	GP sensor fusion	Y	Y	Y	Y	Y	Y	6
	Acceleration comparison	Y	Y	Y	Y	Y	Y	6

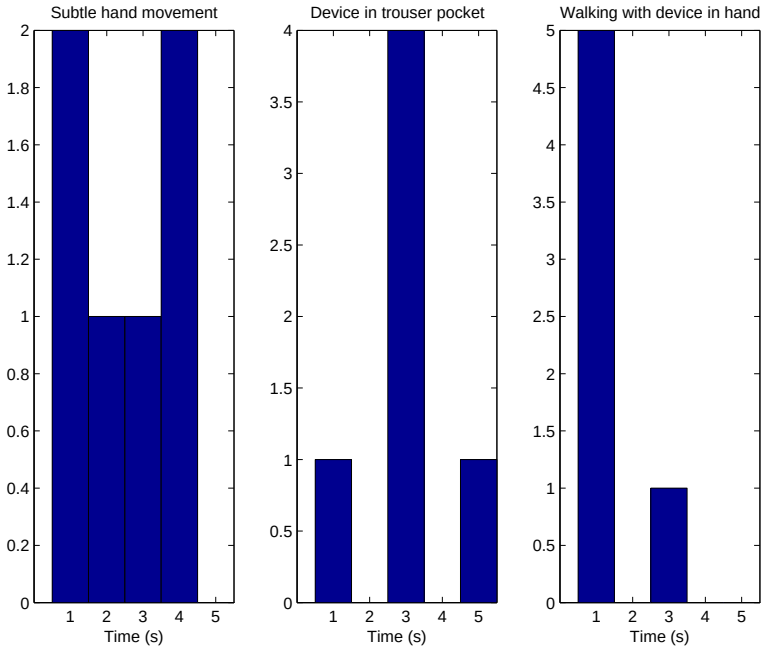


Figure 5.21: Histogram shows the distribution of time needed for user matching. From left to right: (1) subtle hand movement (2) with the device in the user's trouser pocket (3) walking with the device held in the hand.

The histogram in Figure 5.21 shows the distribution of time needed for user matching in all 3 contexts.

5.8.3 Conclusion

In this experiment, we collected walking samples from 6 users and matched the hand position sensed by the Kinect with the acceleration measured by the mobile inertial sensors. Experimental results show that our method achieves successful matches for 6 users. The conventional acceleration comparison method also works in this situation as there were large and vigorous hand movements.

Our method is superior to the conventional acceleration comparison method in the situation when the user performs less vigorous and visible movements with the device held in the hand (the subtle hand movement experiment).

5.9 Conclusions

This chapter presents a sensor fusion system for user matching and identification in a proxemic interaction environment by fusing the joint positions sensed by the Kinect sensor (and the inferred positions) and the accelerations measured by the mobile inertial sensors.

The proposed transformed Gaussian Process prior model incorporates the low-sampling-rate measurements and the high-sampling-rate derivatives, and takes the different noise characteristics of these sensors into account. The area focussed on this chapter is that of user matching and identification, which was achieved by comparing the joint log-likelihoods given different combinations of the positions from a fixed base Kinect and accelerations from multiple mobile devices. The proposed sensor fusion method can significantly improve the context sensing capability of the system.

The performance of the proposed sensor fusion approach to user matching and identification was tested in three contexts (1) subtle hand movements (2) with the device in the user's trouser pocket and (3) walking with the device held in the hand, respectively. We compared our approach with the direct acceleration comparison method and summarised the comparison results in Table 5.3, Table 5.6 and Table 5.9. Experimental results show that our method achieves successful matches in all 3 contexts, including when there are only subtle hand movements, where the direct acceleration comparison method (equation (5.9)) fails to find a match. The direct acceleration comparison method works only when there are vigorous and visible movements, which are unique to the user. Thus, our method is more robust and is more applicable in comparison with the direct acceleration comparison method. We concluded that the matching of the Kinect skeleton and mobile device is feasible through the human everyday movements, e.g. the subtle hand movement, moving around in the room with a mobile device in the pocket or held in the hand.

In experiment 1 (subtle hand movements), we studied the performance of the proposed GP prior model for user matching and identification when the users performed subtle hand movements. The principled nature of the proposed GP prior model is that it can incorporate the low-sampling-rate measurements and the higher frequency second derivatives, taking account of the uncertainty of each sensor type. This is beneficial for subtle movements when the user interacts with a mobile device. In experiment 2 (mobile device in the user's trouser pocket), we investigated the user matching problem when the mobile device was in the user's trouser pocket. It was a realistic scenario. The inferred position of the phone was based on the locations of the knee and the hip. Although the Kinect cannot detect the pocket position, we can infer this position based on the positions of relevant skeleton joints. The GP framework provides a consistent mechanism for performing inference in such situations.

In experiment 3 (walking with the device held in the hand), we can see that user 2 requires a bit longer time (3s) for matching. This is due to the fact that the walking pattern of user 1 and that of user 2 are very similar, that is, their motion were nearly synchronized. The similarities include the position similarity and the acceleration similarity. This can be seen in Figure 5.20. Thus, it takes a longer time for the system to distinguish user 2 from other users. This indicates that it will take the system a longer time to match and identify two users when they move in the same way, that is, there will be similar motion patterns between

them. As long as enough data are available, the proposed log-likelihood comparison method can finally find a best acceleration match for each skeleton.

The sensor fusion approach-based proxemic-aware system enables the user to get a personalized service in a comfortable way in a proxemic interaction application. In our approach, the user does not need to perform any specific gestures. Besides, the user does not need to wear any additional sensor on the body. The system can match the user with the device unobtrusively when he/she subtly moves the device in the hand, moves or walks around with the device in the trouser pocket or held in the hand. Walking is intuitive and unobtrusive and it does not require the user's full attention. Thus, the matching of the Kinect skeleton and the mobile device is feasible through people's everyday movements. Our method can be used for user matching and identification in people's everyday lives.

This study can be of benefit to any context-aware interaction systems that explore the use of mobile inertial sensors and external position sensing device for user matching and identification. In the proposed scenario, e.g. the proxemic-aware system has the potential to be used in a family environment and provide personalized multimedia services and TV programs to each family member when the system matches a skeleton with a personal device through his/her everyday movements.

Chapter 6

Experiment – User Performance Improvement in Sensor Fusion System

In this chapter, we apply the Gaussian Process prior model-based sensor fusion system for a trajectory-based target acquisition task in a spatially aware display application. We discuss the challenges in target acquisition and the importance of visual feedback. Following this, we illustrate the design of the spatially aware display application in detail. Finally, we conduct a user study on a trajectory-based target acquisition task in the spatially aware display application. We implemented the real-time sensor fusion system by augmenting the Kinect with a Nokia N9. In the trajectory-based interaction experiment, each user performed target selection tasks following a trajectory in (a) the Kinect system and (b) the sensor fusion system. In comparison with the Kinect time-delay system, our system enables the user to perform the task more accurately and more quickly. The MSE of target selection was reduced by 38.3% and the average task completion time was reduced by 26.7%.

6.1 Introduction

In this chapter, we designed a spatially aware display application to test the proposed sensor fusion system. We built an experimental setup where users followed trajectories and performed target selection in this spatially aware display application. The GP prior model-based sensor fusion prediction facilitated the targeting action of the user. Experimental results show that the improved accuracy, and reduced delay from the sensor fusion system, compared to the filtered system means that users can acquire the target more rapidly, and with fewer errors. They also reported improved performance in subjective questions.

Position Uncertainty, Lag and Target Acquisition

The position uncertainty and the lag are two key problems in indoor location-aware applications, among which peephole interaction plays an important part nowadays. High precision and low latency play an important role in a spatially aware display for handheld devices (Rohs & Essl, 2006; Cao & Balakrishnan, 2006). However, the hand position uncertainty, i.e. hand jitter, is a common issue in a spatially aware display with a handheld device (Kaufmann & Ahlström, 2012). Pavlovych & Stuerzlinger (2009) studied the effects of spatial jitter and latency on human performance in target pointing tasks.

Target acquisition, in which lag and frame rate play important roles, has been thoroughly researched in numerous studies (Ware & Balakrishnan, 1994; Rohs & Oulasvirta, 2008; Rohs et al., 2011). Ware & Balakrishnan (1994) investigated the effects of lag and the frame rate on human performance in target selection in virtual reality displays and showed that lag is critical in such applications and is detrimental to performance. Rohs et al. (2011) analysed target acquisition with camera phones as magic lenses and as dynamic peephole displays, showing the performance could be adequately modelled with a modification of the standard Fitts' law.

Models of trajectory-based interaction have been investigated in (Accot & Zhai, 1997). Accot & Zhai (1999) investigated the human performance modelling in trajectory-based tunnel steering tasks.

Latency and jitter adversely affect human performance in 2D pointing tasks with stationary targets (Pavlovych & Gutwin, 2012). Human movement prediction can help cope with the latency in real-time tracking systems and gesture applications. Rusdorf & Brunnett (2005) proposed a prediction method for predicting the movements of the tracked objects in a table tennis application.

The lag influences the user performance and experience in human-computer interaction. In experiments, latency typically exhibits strong negative effects on user performance starting at about 100ms (MacKenzie & Ware, 1993; Hinckley & Wigdor, 2002). User experience is the broad array of outputs the user perceives and inputs the user gives when interacting with a user interface, as well as the higher level goals, cognitive states, emotions, and social interactions that these experiences support and engender (Hinckley & Wigdor, 2002). The NASA Task Load Index (Hart & Staveland, 1988) questionnaire has been widely used for gathering subjective assessment of usability of the interaction systems.

6.2 Background

6.2.1 Feedback Control System

The term "feedback" refers to a situation in which two (or more) dynamical systems are connected together such that each system influences the other and their dynamics are thus strongly coupled (Aström & Murray, 2010). The "control" in engineering is the use of algorithms and feedback in engineered systems. A system that involves a person controlling a machine is called manual control (Franklin et al., 2001). The study of manual control focuses on the human's ability to close-the-loop when a person is in control of a system, such as a vehicle or a machine (Jagacinski & Flach, 2011). The systems can be divided into open-loop systems and closed-loop systems. When the controlled output signal is measured and fed back for use in the control computation, the system is called closed-loop or feedback control (Franklin et al., 2001).

Feedback plays an important role in control system design. It is essential for the control of any system subject to uncertainty. It allows a system to be insensitive both to external disturbances and to variations in its individual elements. Appropriate feedback can help the user better interact with a system. Without feedback, the system will be like an open-loop system and the user will be unaware of the interaction effects. Thus, feedback is of great importance for control system design and is beneficial for improving the user's performance.

Control theory can be linked to Fitts' law by viewing the pointing movements towards the target as a feedback control loop based on visual input and the limb as a control element. Dynamic system and manual control theory can be used as a theoretical framework for interaction design (Jagacinski & Flach, 2011). In a closed-loop system, the user interacts with the system through his movement. The system feedback enables the user to better control the handheld device, thus facilitates the user's pointing in a targeting task.

6.2.2 Visual Feedback

Visual feedback plays an important role in human-computer interaction (HCI) systems. Feedback is a word widely used to denote different meanings in several academic areas, including engineering, economics, biology, mathematical models or biological systems, formal logic, and social science (Richardson, 1999). In HCI, feedback is essential as any treatment of input devices without regard to the corresponding visual feedback is like trying to use a pencil without paper (Hinckley & Wigdor, 2002). An interaction system needs to provide a way for users to accomplish tasks by combining input with appropriate feedback.

Visual feedback proves useful for gesture interaction (Bau & Mackay, 2008). Bennett et al.

(2011) investigated the predictive gesture entry and non-predictive gesture entry, and concluded that predictive gestures with visual feedback lead to enhanced performance.

Visual feedback helps a user control the movements. The visual feedback control of hand movements was investigated in (Saunders & Knill, 2004). Saunders & Knill (2003) concluded that visual feedback from the hand contributes to on-line control of reaching throughout the full extent of the movement, even for relatively fast movements.

The effects of feedback on targeting performance have also been investigated. It has been demonstrated that feedback helps improve the user's targeting performance (Akamatsu et al., 1995; Mould & Gutwin, 2004).

The real-time feedback plays an important role in assessing and improving the user performance (Velloso et al., 2013). The Microsoft Kinect has been used for motion modelling and analysis. Alexiadis et al. (2011) used the Kinect technology for a project that provided real-time evaluation of dancers with visual feedback, and also supported evaluation of one dancer's performance against another. Velloso et al. (2013) proposed a system that extracted a movement model from a trainer, assessed the performance of other users and provided real-time feedback on how to improve their performance.

In our work, the visual feedback provided by the mobile phone helps the user to control the phone and the hand movement for reaching the visual targets located in the virtual information space embedded in the physical environment. We treat this visual feedback as an input of the human subsystem, which is a part of the overall system.

6.3 Augmenting the Kinect System with Mobile Device in Spatially Aware Display

In this chapter, we conducted a user study to test the performance of the GP prior model-based sensor fusion system. We augmented the Kinect system with a mobile device (N9) and developed a spatially aware display application.

6.3.1 System Overview

The equipment includes the Microsoft Kinect and Nokia N9. Details about the sampling rate of the Kinect and the drivers can be found in section 3.3. The sampling rate of the inertial sensor embedded in the N9 is 90Hz.

Sensor data should be converted into the same coordinate system before sensor fusion (Titterton et al., 2004). The acceleration sensed by the mobile inertial sensors and expressed in

the body frame needs to be converted to the acceleration in the Kinect frame. We gave a detailed description on the coordinate system transformation method in section 3.3.2. In the spatially aware display application, we also have the N9 phone image frame. For this image frame (i), the top left corner is $(0, 0)$(pixels) in the landscape mode. This can be seen in Figure 6.4. The slider on the axis of the phone image coordinate system indicates the location of the phone on the virtual information space. There is a mapping between the virtual information space and the Kinect skeleton position space. The hand position sensed by the Kinect should be converted to the coordinates defined on the virtual canvas. The virtual information defined on the 2D canvas is mapped to the coordinates (pixels) on the phone image frame. In this way, the virtual information displayed on the phone screen will change accordingly when the user moves the device (N9) held in the hand.

We define a 2D virtual space (2 m $\times$ 1 m). The phone is like a movable window on this canvas, which is defined in pixels (20000×10000). The part of canvas displayed on the phone screen is determined by the hand position in the Kinect frame.

6.3.2 Augmenting the Kinect System with a Mobile Device (N9)

The system architecture of our spatially aware display application is shown in Figure 6.1. Our design focuses on a 2D (the vertical XY plane) version of a spatially aware display. We aim to test whether the GP prior model-based sensor fusion system improves user performance by designing a trajectory-based target acquisition task in the spatially aware display application.

An example application is shown in Figure 6.2. A user is exploring the digital information stored in the physical space. By moving his phone to different locations following a trajectory, he can perform a target selection task. Imagine this is a virtual bookshelf application (Norrie et al., 2013). We store different digital books in different targets' locations. The user can search and browse those digital books by category. Since accurate positioning and feedback is critical to this application, we need to deal with the Kinect position jitter. The lags and overshoot problems occur when we apply a position-only Kalman filter for filtering the fast jitter on the hand movement data. Our novel GP sensor fusion method helps reduce the lags and deal with the overshoot problems when the user tries to find the target by moving the hand quickly.

The Nokia N9 is a phone with 3.9 inches display (480 pixels $\times$ 854 pixels) (size 48 mm $\times$ 86 mm). It is equipped with a $3-$axis accelerometer, which can be used to estimate the hand acceleration. The Kinect is put on a table. The user stands in front of the Kinect with the N9 held in the right hand, and is directly facing the XYplane, i.e. the vertical interaction

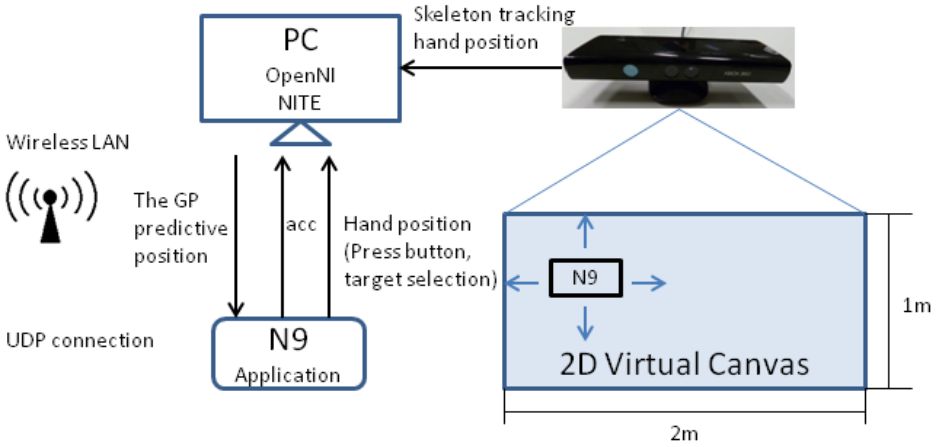


Figure 6.1: System architecture. A Wireless LAN is used for UDP connection. The OpenNI and NITE middleware are used. The Kinect senses the hand position and sends it to the PC. The accelerometer data from the phone is also sent to the PC. Our novel GP sensor fusion model is applied for fusing the position and the acceleration. The GP predictive position is sent to the phone. The phone is a movable window on the 2D virtual canvas, on which we put a pre-designed trajectory and 6 targets. When the virtual button on the phone screen is pressed, the target on the canvas is selected and the current hand position is sent back to the PC.

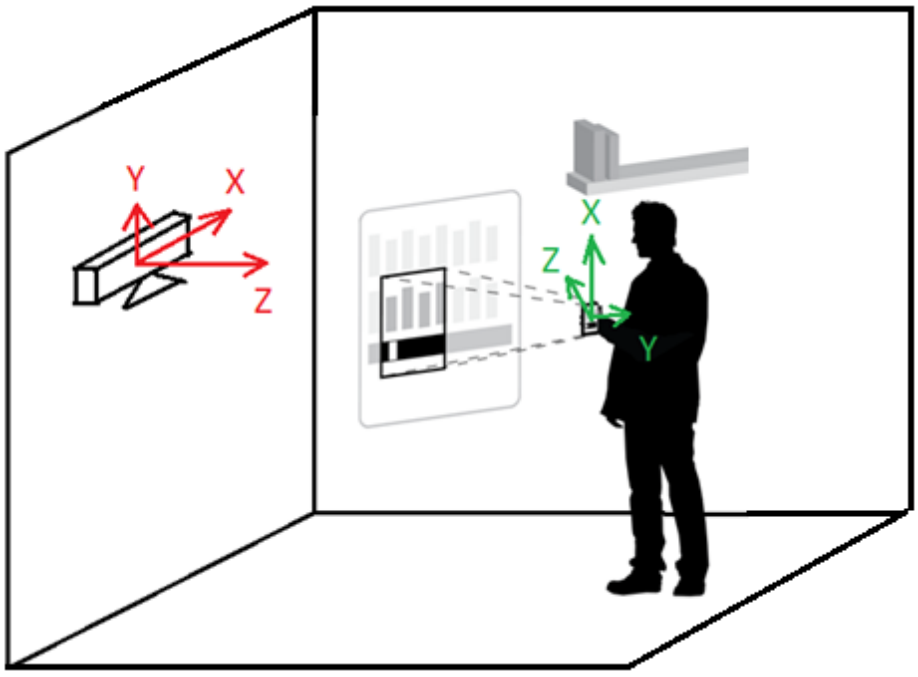


Figure 6.2: Spatially aware display application. A phone user performs a trajectory-based target selection task in 2D space.

plane.[1] The information (the trajectory and the targets) is spread out on a flat virtual space. The phone acts as a movable window (size 48 mm $\times$ 86 mm) on this much larger 2D virtual

[1]In this implementation, a fixed rotation matrix between the phone body frame and the Kinect frame is assumed.

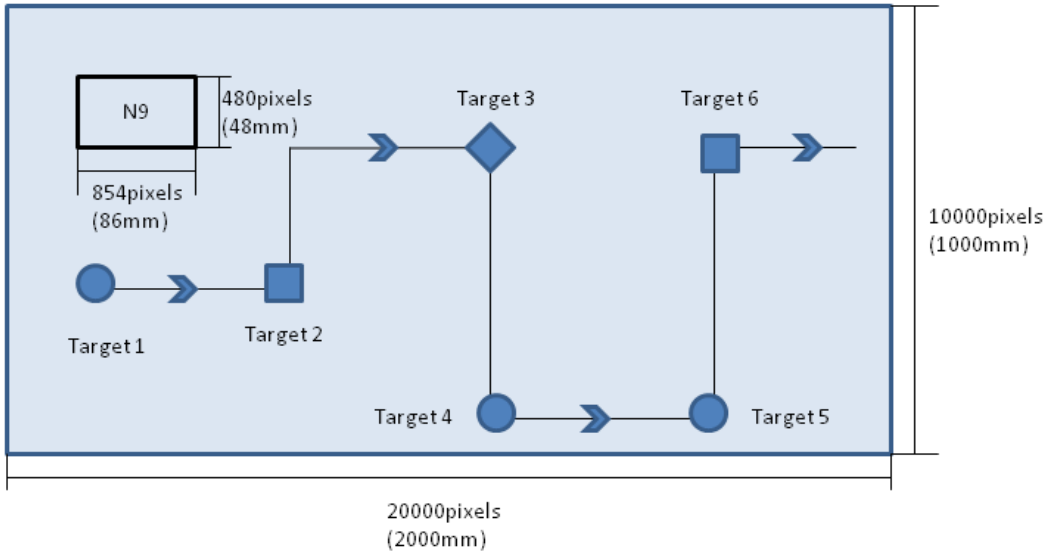


Figure 6.3: 2D virtual canvas design. The canvas covers a 2 m $\times$ 1 m area in the Kinect XY plane. N9 is a phone with 3.9 inches display (480 pixels $\times$ 854 pixels) (size 48 mm $\times$ 86 mm). Thus, when the size of the canvas is expressed in pixels, it is 20000 pixels $\times$ 10000 pixels. We use the straight line and square wave for modelling the trajectory, on which 6 targets are located.

canvas. The 2D canvas covers a 2 m $\times$ 1 m area.

When the user moves the hand in the 2D plane in front of the Kinect, he/she tries to explore the digital information stored in the physical space. The trajectory and the targets are located on the virtual canvas, which is shown in Figure 6.3. There is a mapping between the 2D real world space (mm) and the 2D virtual canvas (pixels). In our application, 1 mm $\times$ 1 mm = 10 pixels $\times$ 10 pixels. Along the $x-$axis, the range is $(-1000, 1000)$ mm, whereas $(0, 1000)$ mm for the $y-$axis.

The 2D plane is like a big virtual canvas, and the phone screen is a small movable window, through which we can see a part of the virtual canvas. No visual information is present in the real world outside the device's display. The hand position (x, y) indicates the phone position. When the phone moves, the corresponding part of the canvas will be displayed on the phone screen. When the user's hand is moved to the location where the trajectory is stored in the physical space, the mapped trajectory will be displayed on the phone screen. When the user moves the hand, the Kinect and the accelerometer sense this. The predictive position from the GP sensor fusion is sent to the phone to update the display on the phone screen. The user needs to move the hand along the trajectory. When a target appears on the screen, the user performs target selection. A selection occurs when the virtual button on the phone screen is tapped. On the N9, we designed a square virtual button and put it at the right side of the screen, as shown in Figure 6.4. As soon as the button is pressed, we record this hand position, where the user believes the target is located. Whenever the user presses the button,

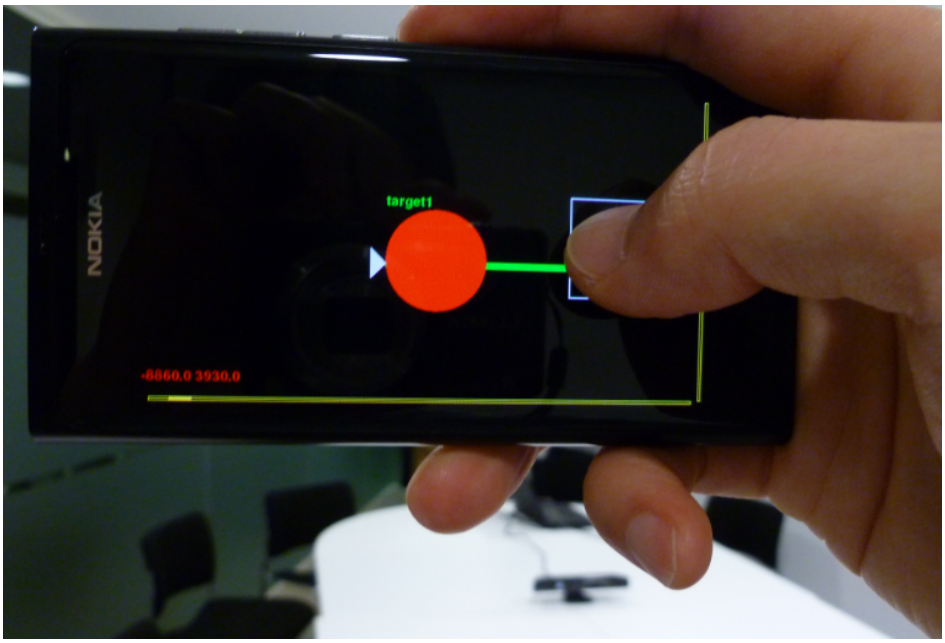


Figure 6.4: The interface on the Nokia N9 phone screen in the spatially aware display application. A user was performing the trajectory-based target selection task. The first target was shown on the screen. The square box on the right of the screen is the virtual button. When the button is pressed, the target is selected. Meanwhile, the visual feedback (the color of the button changes) is provided for the user during the target selection task.

the phone will send a signal and the PC will record the current hand position.

In the augmented system, a UDP connection is applied for sending and receiving data. A Wireless LAN is used for connecting the computer running the sensor fusion system and the N9 phone. The hand tracking positions sensed by the Kinect are sent to the N9 via WiFi. Meanwhile, the accelerometer data from the N9 is also sent to the computer via WiFi. The hand position measurements and the accelerations are fused with the proposed GP prior model-based sensor fusion approach for hand position prediction. The data transmission between the phone and the PC includes three parts:

1. The phone transmits the accelerometer data to the PC.
2. Sensor fusion with our novel GP prior model on the PC. The PC sends the GPs predictive position (x, y)mm to the phone.
3. The phone sends a signal to the PC when the user presses the virtual button to select the target.

This can be seen in Figure 6.1.

Trajectory Design Tool

In order to simplify the process of trajectory design, we designed an application that can simulate the Kinect position output. To draw a trajectory, simply click and drag the mouse pointer. The position data can be saved and used for trajectory design in the spatially aware display application. In addition, this application can also be used to simulate the Kinect position outputs. Simply click and drag the mouse pointer. It will trace the position of the mouse and send it to the phone application.

6.4 Experiment: User Study – Trajectory-based Target Acquisition Task

The user study aims to test our sensor fusion system when the user performs a 2D trajectory-based target selection task in a spatially aware display application. The application diagram is shown in Figure 6.2.

Experimental results show that our system helps the user perform the task more quickly and more accurately in comparison with the conventional Kinect system. The real-time visual feedback shown on the phone screen guides the user's hand to quickly locate the desired targets along the pre-designed trajectory.

6.4.1 Participants and Apparatus

There were 12 participants in total (6 male, 6 female). They were aged between 20 and 35 years (mean age 28). Participants were recruited by email, and some volunteered from the academic community in our school. The task was performed on a Nokia N9, which is a phone with 3.9 inches display (480 pixels $\times$ 854 pixels or 48 mm $\times$ 86 mm).

6.4.2 Data Collection and Analysis

We aim at analysing the accuracy of target selection and the task completion time. In the task, we recorded the hand position sensed by the Kinect and the hand acceleration measured by the Nokia N9. When the participant performed the target selection task, the hand position was recorded. We analysed the accuracy of target selection. Besides, we measured and analysed the task completion time. Following the experiment, the participants completed the NASA Task Load Index (Hart & Staveland, 1988) questionnaire, which gathered subjective assessment of usability of the system.

6.4.3 Experiment Design

The Kinect was put in a fixed location on a table, as shown in Figure 6.4. The participant stood at a distance of 2m from the Kinect. They were instructed that there were 6 targets on the vertical virtual interaction space. The task was to find the first target, select it and then move the phone along the virtual trajectory to find the next target and select it. The participant kept moving the phone along the virtual trajectory to find and select the remaining targets. When they finished the selection of the sixth target, the experiment was done. The participants were instructed to interact with the system in a comfortable way. Then they were instructed to perform the trajectory-based target selection task as accurately and quickly as possible. Each participant performed the task in (1) the Kinect system (2) the sensor fusion system. After each session, the user completed the questionnaire. The users were not informed which system they were using. Task 1 and task 2 were denoted on the questionnaire.

At the beginning of the experiment, the user stood in front of the Kinect with a mobile device (Nokia N9) held in the hand and was directly facing the XY plane, i.e. the vertical interaction plane. Once skeleton tracking locked on, the user moved his hand following the pre-designed trajectory, which was only shown on the phone screen. There was no visual information outside the phone. Whenever a target appeared on the trajectory, the user selected it by pressing the virtual button on the phone screen, as shown in Figure 6.4. Meanwhile, this position was recorded and sent back to the PC. It was compared with the ground truth data (we know the real position for the targets) for error rate analysis. The diagram of the system design is shown in Figure 6.1.

Trajectory Design

We used a combination of a straight line and a square wave curve for modelling the trajectory for the target selection task. Six targets were located along the trajectory. The pre-designed trajectory and the targets are shown in Figure 6.3. The arrows on the trajectory indicate the moving direction.

Data in Spatially Aware Display

The Kinect senses the position of the hand. Meanwhile, the N9 accelerometer is used to measure the hand acceleration. The accelerometer embedded in the N9 was calibrated before the experiment started. The hyperparameters for GPs are set to $v_0 = 5.66 \times 10^4$, $\omega_1 = 4.19$, $\sigma_y^2 = 64$ and $\sigma_a^2 = 100^2$.

The phone sends the acceleration to the PC. When the PC receives the Kinect position and the acceleration sensed by the inertial sensor, our novel Gaussian Process model is applied for sensor fusion. The GP sensor fusion results (the predictive position, i.e. (x, y) mm coordinates) are sent to the phone for updating the canvas display. The average time needed for this process was estimated to be 0.017s. In our sensor fusion system, the predictive hand position is also treated as the position of the screen centre. The digital content (e.g. a part of the trajectory) located in this area will be displayed on the screen.

We compared our system with the conventional Kinect system, in which a single rate Kalman filter (Feng & Murray-Smith, 2014) was applied for filtering the noisy position measurements. The filtered position was sent to the phone for updating the canvas display. We compared this Kinect system with our sensor fusion system.

6.4.4 Experimental Results

Accuracy of Target Selection

We compared the target selection position with the ground truth data, i.e. the real target position defined on the virtual canvas. In order to compare the accuracy of target selection in two systems, we calculated the Mean Square Error (MSE).

The comparison results are shown in Figure 6.5. The MSE of target selection in the Kinect system is 3.7263×10^5 pixel^2 (standard deviation SD 2.1096×10^5). For the sensor fusion system, it is 2.2975×10^5 pixel^2 (standard deviation SD 1.2452×10^5). The MSE is reduced by 38.3%.

Results were analysed using a repeated measures Analysis of Variance (ANOVA) (Albert & Tullis, 2013). The sensor fusion system has a statistically significant effect on the target selection accuracy, $F(1, 11) = 10.86$, $p = 0.0071$.

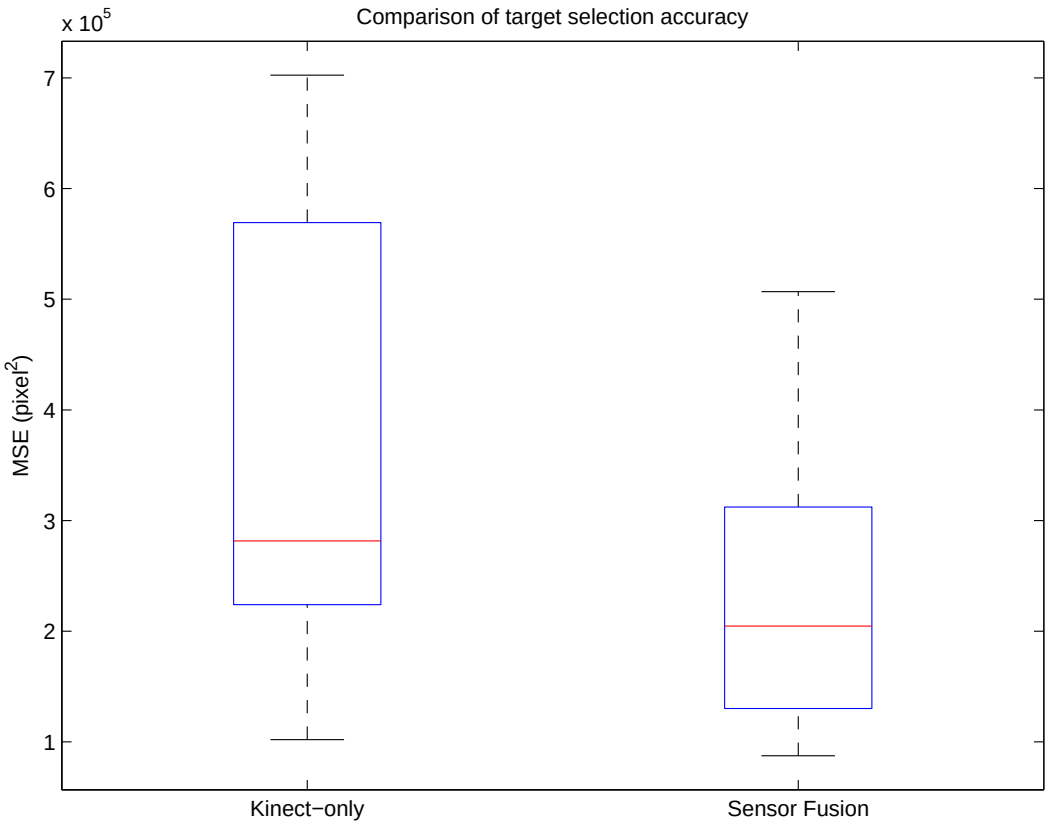


Figure 6.5: Comparison of target selection accuracy. Plots show the MSE of target selection. Left column: the Kinect system. Right column: GPs sensor fusion system. It can be seen that the target selection error is reduced by the sensor fusion.

Task Completion Time

The task completion time for our sensor fusion system (mean=32.41s, SD=12.04s) is shorter than that for the Kinect system (mean=44.21s, SD=14.77s). The average task completion time is reduced by 26.7%. A comparison of the average task completion time is shown in Figure 6.6.

Results were analysed using a repeated measures Analysis of Variance (ANOVA). The GPs sensor fusion system has a statistically significant effect on the task completion time, $F(1, 11) = 12.05$, $p = 0.0052$.

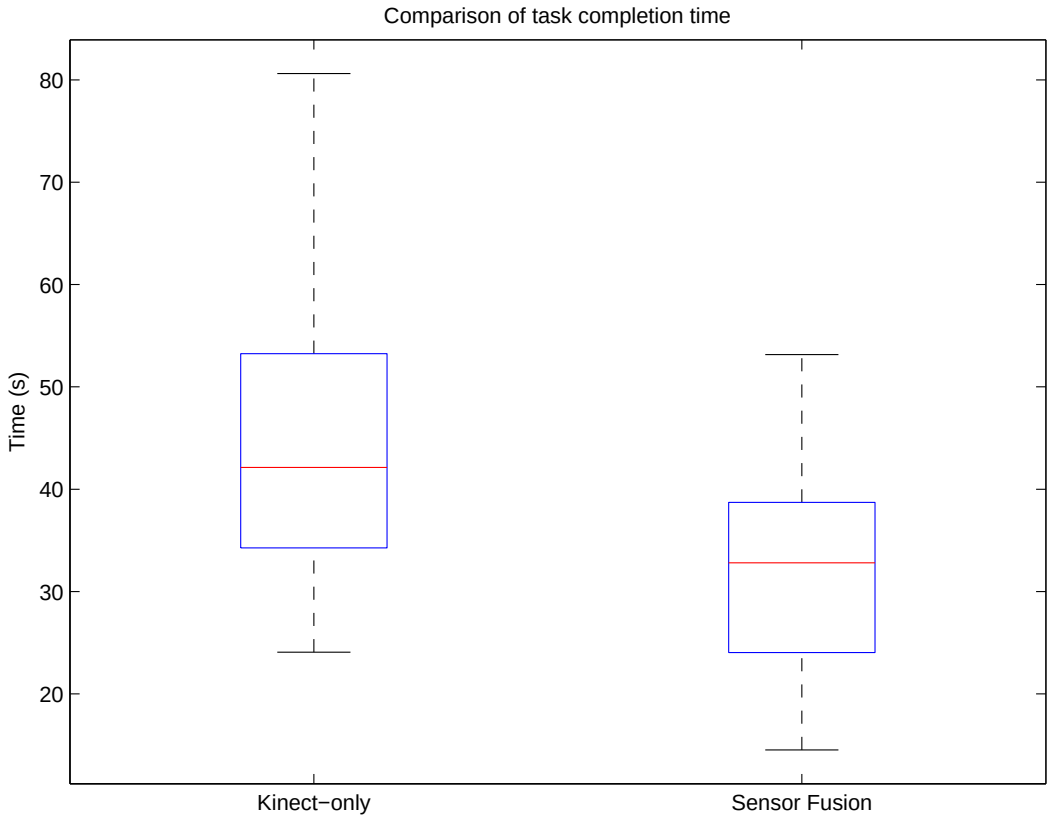


Figure 6.6: Comparison of task completion time. Left column: the Kinect system. Right column: GPs sensor fusion system. It can be seen that the average task completion time is reduced by the sensor fusion.

Table 6.1: The NASA Task Load Index

Scale	Scores for Different Systems			
	Kinect System		Sensor Fusion System	
	Mean	SD	Mean	SD
Mental Demand	7.17	3.64	5.08	2.78
Physical Demand	7.75	3.11	7.17	3.54
Temporal Demand	11.25	3.47	9.92	4.01
Performance	6.50	4.06	5.25	3.08
Effort	10.92	4.66	9.33	4.38
Frustration	8.17	4.37	6	4.11

Questionnaire

Following each session of the experiment, each participant was asked to complete the NASA Task Load Index questionnaire. For each scale, the line is divided into 20 intervals. From left (low) to right (high), scores range from 0 to 20 (Stanton et al., 2013). The lower score indicates a better performance. The conventional Kinect system obtained a score of 619, whereas our sensor fusion system obtained a score of 513. The subjective load varied in line with the objective measures of speed and accuracy.

We have two systems and need to do a paired sample test. Results were analysed using a Wilcoxon signed-rank test. We get the following results: (1) The mental demand, $p = 0.0137$. (2) The physical demand, $p = 0.0898$. (3) The temporal demand, $p = 0.0508$. (4) The performance, $p = 0.0249$. (5) The effort, $p = 0.1611$. (6) The frustration, $p = 0.0195$. It can be seen that the GPs sensor fusion system has a statistically significant effect on the mental demand, the temporal demand, the performance and the frustration.

For each scale, we also calculated the mean score and the standard deviation. The results are shown in Table 6.1. We can see that the average subjective assessment of usability of our sensor fusion system is better than that of the Kinect system. The histogram comparison results are shown in Figure 6.7. The Boxplot shows the comparison results of the NASA Task Load Index for the Kinect system and the sensor fusion system in Figure 6.8. The lower score of each scale indicates a better performance of the system. It can be seen that the sensor fusion system outperforms the Kinect system in the subjective assessment of usability of the system.

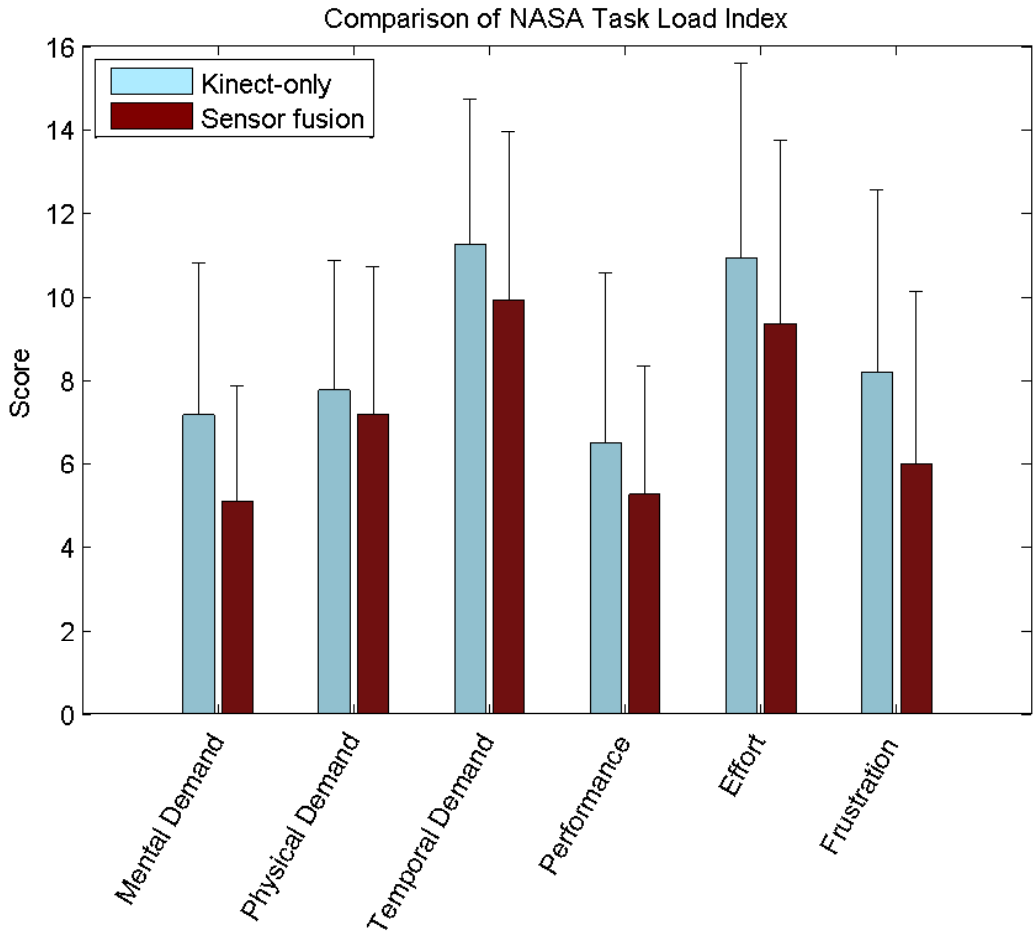


Figure 6.7: Plots show the mean+σ. For each scale, the comparison results of the NASA Task Load Index are shown. The lower score indicates a better performance. (the Kinect system and the sensor fusion system)

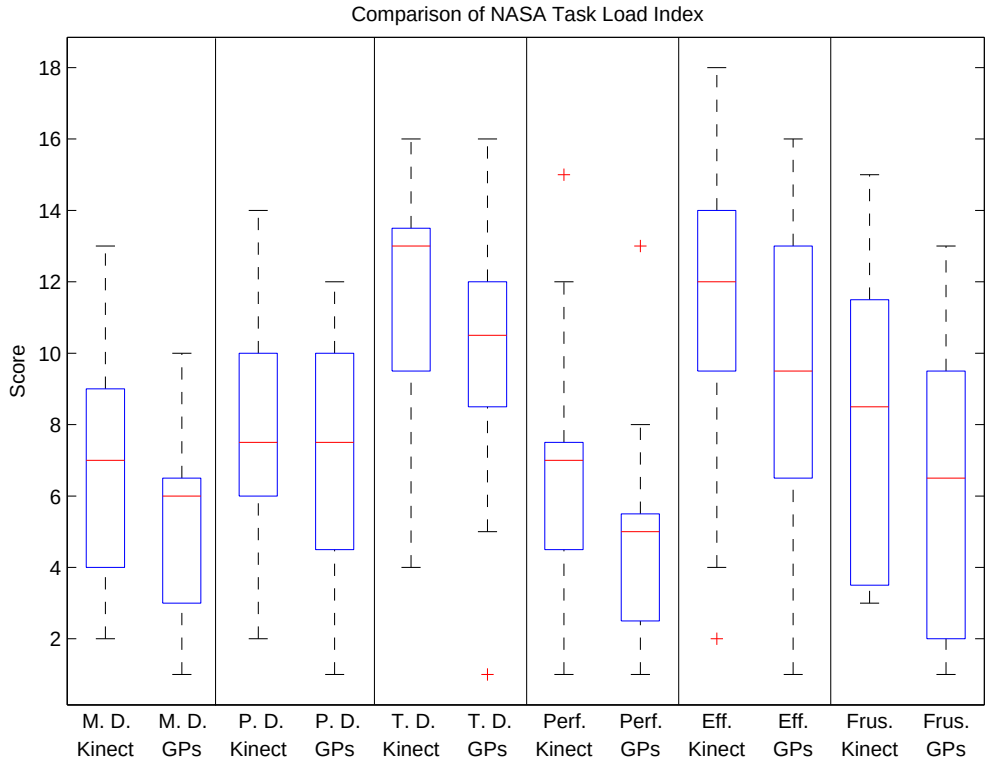


Figure 6.8: The Boxplot shows the comparison results of the NASA Task Load Index for the Kinect system and the sensor fusion system. The 6 scales along the $x-$axis are (1) Mental Demand (2) Physical Demand (3) Temporal Demand (4) Performance (5) Effort (6) Frustration. The "Kinect" (along the $x-$axis) represents the Kinect system. The "GPs" (along the $x-$axis) represents the sensor fusion system. A lower score indicates a better performance.

6.4.5 Conclusion

We conducted a user study on a trajectory-based target acquisition task in a spatially aware display application. We implemented the real-time sensor fusion system by augmenting the Kinect with a Nokia N9. In the trajectory-based interaction experiment, each user performed target selection tasks following a trajectory in (a) the Kinect system and (b) the sensor fusion system. We analysed the accuracy of target selection and task completion time, respectively. Experimental results show that our system enables the user to perform the task more accurately and more quickly in comparison with the Kinect system. The target selection error and the task completion time are both reduced by the GP sensor fusion. Moreover, we used the NASA Task Load Index for subjective assessment of usability of the sensor fusion system. Experimental results show that the GPs sensor fusion system has a statistically significant effect on the mental demand, the temporal demand, the performance and the frustration. The participants reported improved performance in our system.

6.5 Conclusions

This chapter explores the benefits to users of a Kinect-based sensor fusion system for improving the accuracy of joints position estimation and reducing the lag by fusing the Kinect sensor and the built-in inertial sensors in a mobile device. We conducted an experiment to test the GP prior model-based sensor fusion system. We built a spatially aware display application for user study. The user performed trajectory-based target acquisition tasks in two systems: (1) the Kinect system; (2) the proposed sensor fusion system. We found that the GP prior model-based sensor fusion can help users perform the tasks more quickly and more accurately in the spatially aware display application.

Firstly, we discussed the effects of position uncertainty and the lag on human performance in target acquisition. We gave a brief introduction to the feedback control system. We described the visual feedback and emphasized the importance of visual feedback in human movement control applications. After that, we presented the system overview and gave a detailed description on the design of the spatially aware display application. We augmented the Kinect system with a mobile phone (N9). We designed a 2D virtual canvas, where the trajectory and the targets were located. The mapping between the virtual information space and the Kinect space was discussed. We designed an interface on the N9 screen, where the user could perform target selection. We illustrated the data transmission between the PC and the phone. The Kinect sensed the user's hand position and the phone measured the hand acceleration. The position data and the acceleration data were fused for hand position prediction using the proposed GP prior model-based sensor fusion approach. The predictive position was sent to the phone for updating the screen display.

We conducted a user study to test the performance of the proposed sensor fusion system in this spatially aware display application. Experimental results show that the users can acquire the target more rapidly, and with fewer errors. The average task completion time was reduced by 26.7% and the MSE of target selection was reduced by 38.3%. The participants also reported improved performance in subjective questions. The average subjective assessment of usability of our sensor fusion system is better than that of the Kinect system.

Chapter 7

Conclusions

This thesis presents a multi-rate sensor fusion system composed of a Microsoft Kinect and mobile inertial sensors. We describe the use of Bayesian nonparametric models, i.e. transformations of Gaussian Process (GP) priors, to improve the context sensing capability of a system composed of a diverse set of sensors. These sensors are used to measure the typical human movements, which are relatively continuous and smooth, in people's everyday lives. The proposed GP prior model provides a principled mechanism for incorporating the low-sampling-rate position measurements and the high-sampling-rate derivatives in multi-rate sensor fusion which takes account of the uncertainty of each sensor type.

The Gaussian Process prior model-based sensor fusion approach is of great benefit for exploring the use of multiple complementary sensors for proxemic-aware sensing applications. We explore the complementary properties of the Kinect sensor and mobile inertial sensors, and apply the GP prior model for two applications.

1. User matching and identification, i.e. identify individual users, by matching the observed Kinect skeletons with the sensed inertial data from their mobile devices.

2. Skeleton joint position stabilisation and lag reduction. As an example, the Kinect was augmented with a mobile device (N9) in a spatially aware display application. We investigated the performance of the proposed GP prior model-based sensor fusion system in this application.

In this thesis, we present the literature review in Chapter 2, where we introduce the context-aware sensing and the multisensor data fusion approaches. In Chapter 3, we describe the multi-rate sensors-based Kalman filter method for fusing the Kinect sensor and the mobile inertial sensors. We then give a detailed description on the GP prior model-based sensor fusion approach in Chapter 4. After that, we present the two applications in Chapter 5 and Chapter 6, respectively.

7.1 Sensor Fusion with Multi-rate Sensors-based Kalman Filter

In Chapter 3, we apply a multi-rate sensors-based Kalman filter to explore the complementary properties of the Kinect and the mobile inertial sensors. The Kinect can be applied for skeleton tracking, which gives the skeleton joints positions. Meanwhile, the built-in inertial sensors in the mobile device sense the skeleton joint motion. When the mobile device is held in the hand, the hand acceleration can be estimated through inertial sensor fusion. The complementary properties of these sensors were introduced in section 1.2.2.

We explore the use of mobile inertial sensors to estimate the skeleton joint acceleration in the Kinect coordinate system. To fuse the position measurements sensed by the Kinect and the acceleration measured by the mobile inertial sensors, we need to estimate the acceleration through inertial sensor fusion first. This was described in section 3.4. In order to convert the linear acceleration from the body frame to the Kinect frame, we proposed a coordinate system transformation method in section 3.3.2.

The sensor fusion with the multi-rate Kalman filter helps improve the system state estimation. As discussed in section 3.5, we compared our method with a single rate Kalman filter and presented a clear illustration of using the proposed method to improve the estimation of position, velocity and acceleration. We conclude that the sensor fusion helps improve the accuracy of the system state estimation including the position, the velocity and the acceleration.

The sensor fusion with the multi-rate Kalman filter helps increase the stability of the position sensed by the Kinect (see Figure 3.16). The increased stability of hand position gives the user a smoother experience, thus helps the user better interact in the information space embedded in a physical environment. This is important for the interaction systems that aim to fuse mobile inertial sensors and the external position sensing device for indoor augmented reality (AR) and other location-aware sensing applications.

The sensor fusion with the multi-rate Kalman filter also helps improve the accuracy of velocity and acceleration estimation by fusing a position sensing device (Kinect) and the inertial sensors equipped in a mobile device. We proposed a robust method for estimating the velocity and acceleration of the skeleton joint in the Kinect space (see Figure 3.19 and Figure 3.20).

This robust method is of great use for any interaction system that requires velocity and acceleration sensing. It is not feasible to get robust estimation of velocity and acceleration with only a Kinect sensor or only mobile inertial sensors. This can benefit the pointing task in HCI. Gallo & Minutolo (2012) discussed the importance of reducing the effects of spatial jitters, which were due to the noise in the device signal and the noise from hand tremor,

and described a velocity-oriented precision enhancing technique for remote pointing. Besides, our method is also beneficial for solving the target overshooting problem, which is a key characteristic of peephole pointing (Kaufmann & Ahlström, 2012). This proves that our work in Chapter 3 is beneficial for improving the accuracy of pointing using a mobile device as a pointer in a room-sized environment by fusing the Kinect sensor and the mobile inertial sensors for performance improvement.

7.2 The Sensor Fusion System

In Chapter 4, we present the novel Gaussian Process prior model-based sensor fusion system. This model incorporates the low-sampling-rate measurements and the high-sampling-rate derivatives, and takes the uncertainty of each sensor type and the Kinect latency into account. We gave an introduction to the sensor fusion system, and discussed that the Kinect could be augmented with mobile devices equipped with inertial sensors, including SK7 and the mobile phone. Firstly, the GP prior model-based sensor fusion approach can be used for user matching and identification. Secondly, this approach is very beneficial for improving the usability of an indoor location-aware application by increasing the stability of the position and reducing the lag.

Following this, we presented the problem statement for this dynamic system modelling, as discussed in section 4.3.1. In this closed-loop human-computer interaction system, the human motion is observable by multiple sensors. Considering the joint position uncertainty with the skeleton tracking and the time delay issue in the Kinect system, we presented a nonlinear dynamic system modelling problem due to the complexity of human motion. We described how to fuse the low-sampling-rate position and the higher frequency acceleration with the novel GP prior model, which is an autoregressive Gaussian Process (ARGP) model. After that, we gave a detailed description on how to use transformations of GP priors to fuse the measurements from different sensors. We also presented an alternative view of the multi-rate sensor fusion problem in the Kalman filter framework to help the readers better understand this concept in section 4.4.

We conducted an experiment to test the GP prior model-based sensor fusion system in section 4.5. Experimental results show that the sensor fusion helps improve the accuracy of position estimation, and reduce the lag. The system response time is reduced by 0.11s (see Figure 4.8). This indicates that the high-sampling-rate accelerations can be used to compensate for the effects of the position uncertainty and the latency in a conventional Kinect system.

The GP prior model-based sensor fusion approach is very beneficial for improving the usability of a location-aware application by increasing the stability of the position and reducing

the lag. We have discussed how to apply this model for position prediction in section 4.3.2. Firstly, the sensor fusion can improve the quality of inferred joint positions, as the high-sampling-rate acceleration signal can augment the low-sampling-rate, noisy and delayed position measurements. It can also help to reduce the lag, as the inertial sensing has a lower latency than the position sensed by the Kinect.

In the following chapters, we investigated the performance of the GP prior model-based sensor fusion system. The two key aspects of the GP prior model is (1) the joint log-likelihood of the low-sampling-rate positions sensed by the Kinect and the high-sampling-rate accelerations measured by mobile inertial sensors; (2) the GP sensor fusion for position prediction. The two aspects correspond with the first and second application, respectively.

7.3 First Application – User Matching and Identification

The first application is *user matching and identification*, which was described in Chapter 5. We apply the GP prior model to identify individual users, by matching the observed Kinect skeletons with the sensed inertial data from their mobile devices using the GP-based sensor fusion algorithm.

In Chapter 5, we achieved user matching by comparing the GP log-likelihoods of the position and the acceleration. We presented the problem statement in section 5.3.1 and discussed how to compute the log-likelihood of a combination of the low-sampling-rate positions and the higher frequency accelerations in section 5.3.2. By comparing the log-likelihoods of matching a particular user's skeleton with multiple time-series of acceleration signals sensed by the mobile devices, we can identify which device this user's skeleton is associated with, that is, which device this user carries. We do this for all the skeleton position signals, respectively, allowing us to infer a match for each user.

We investigate the feasibility and usability of the GP prior model-based sensor fusion approach to user matching and identification. We conducted three experiments and investigated the performance of the proposed GP prior model in these situations: (1) subtle hand movement (section 5.6) (2) with a mobile device in the user's trousers pocket (section 5.7) (3) walking with a mobile device held in the hand (section 5.8). We compared our work with the state-of-the-art work presented in the literature and demonstrated that our method achieves successful matches in all 3 contexts, including when there are only subtle hand movements, where the direct acceleration comparison method fails to find a match. The study shows that the GP prior model-based user matching approach is more robust and is more applicable. The three experiments involve a variety of people's everyday movements. The experiments in all

3 contexts indicate that it is feasible to achieve user matching through people's everyday movements in a Kinect-augmented room.

This study can be of benefit to any proxemic interaction or context-aware systems that explore the use of mobile inertial sensors and external position sensing device for user matching and identification. User matching and identification plays a crucial role in providing user-specific information and services in a proxemic interaction system. For instance, the system has the potential to be used in a family environment to provide personalized multimedia services and TV programs to each family member when the system matches a skeleton with a personal device through his/her everyday movements.

Moreover, the proposed GP log-likelihood comparison method is also beneficial for locating the inertial sensors on the human body. The Kinect skeleton tracking provides the 3D coordinates of each joint. By matching each skeleton joint position measurements with the accelerations sensed by a mobile device, we can find the best match of the joint and the device. For example, we can determine whether the device is associated with the hand, i.e. held in the hand, or attached on the hip. However, it may be difficult when we want to determine whether the device is attached on the left hip or the right hip when the user walks. Thus, the matching of the joints and the inertial sensors on the human body will be difficult in this case.

7.4 Second Application – Position Stabilisation and Lag Reduction

Besides user matching, a second application is *position stabilisation and lag reduction* in a Kinect-based location-aware sensing application. In Chapter 6, we augmented the Kinect with a mobile device and developed a spatially aware display application to investigate the performance of the proposed sensor fusion system.

We conducted a user study to investigate the performance of the novel GP prior model-based sensor fusion system. We implemented the real-time sensor fusion system by augmenting the Kinect with a Nokia N9, as discussed in section 6.3.2. In the experiment, each user performed the trajectory-based target selection tasks following a trajectory in (a) the Kinect system and (b) the sensor fusion system. Experimental results show that the improved accuracy, and reduced delay from the sensor fusion system, compared to the filtered system means that users can acquire the target more rapidly, and with fewer errors. In comparison with the Kinect system, our system enables the user to perform the trajectory-based target acquisition tasks more accurately and more quickly in the spatially aware display application. The MSE of target selection was reduced by 38.3% and the average task completion

time was reduced by 26.7%. The results show significant decreases in the error rate of target selection and task completion time in the GP prior model-based sensor fusion system. The participants also reported improved performance in subjective questions. The results were presented in section 6.4.4.

This type of sensor fusion system is very useful for improving the usability of an indoor location-aware sensing application system. By improving the accuracy of skeleton joint position estimation and reducing the lag of the system, we can significantly improve the usability of a proxemic interaction system. In this spatially aware display application, the GP prediction facilitated the user's targeting. We conclude that the enhanced position tracking can improve a user's interaction experience in a Kinect-based location-aware sensing application. For instance, the improved position tracking can facilitate a user's browsing in a digital book library (Norrie et al., 2013).

This research can also benefit the techniques that explore the use of the Kinect as an input device to enhance the capabilities of the mobile device. For instance, the ShoeSense system proposed in (Bailly et al., 2012) provided a novel modality for interacting with mobile devices, e.g. 3D control on mobile devices. No matter the Kinect is mounted onto the shoe of a user or put in a fixed location in a room, the stabilised hand position tracking and the faster system response enable the user to better perform mid-air input gestures, thus enhance the user's interaction.

This research is also beneficial for the pointing tasks in an indoor environment. The combination of position sensing devices and mobile inertial sensors for enhanced pointing has been studied in the literature. Rahman et al. (2010) proposed a sensor fusion approach that combined the data from multiple infrared (IR) sensors and mobile phone accelerometer to locate the position and targeting orientation of a user's mobile device. Nickel & Stiefelhagen (2003) studied the pointing gesture based on 3D tracking of face, hands and head orientation. Imagine a user moves the arm towards a target in front of the Kinect. Given the skeleton joint positions, we can determine the forearm orientation. However, due to the position jitter, there will be error. In order to better estimate the pointing direction, we can augment the Kinect with a mobile device equipped with inertial sensors to facilitate a user's pointing in a proxemic interaction system.

Moreover, with the advent of embedded projector being used in a mobile device, our method is also beneficial for peephole pointing (Fitzmaurice, 1993; Kratz et al., 2012b), which is a promising interaction technique for visualizing the virtual information space embedded in a physical environment. This is also a potential application that could benefit from our work.

7.5 Combination of Two Applications in Proxemic Interaction

The fusion of a Kinect sensor and inertial sensors equipped in a mobile device can significantly improve the context sensing capability of a system. The fusion can be used for user matching and identification. Moreover, the fusion can also help stabilise the skeleton joint position and reduce the lag of the conventional Kinect system. The former enables us to know who the user is and the latter significantly improves the usability of a location-aware sensing application, such as a spatially aware display application. Combination of the two applications brings an enhanced proxemic interaction in a Kinect-augmented room.

The combination of *(1) user matching and identification* and *(2) improved position tracking and reduced lag in the Kinect skeleton tracking* is beneficial for the research on tabletops systems and interactive surfaces, where the robust tracking of user input with high precision and low latency needs to be attributed to individual users (Kratz & Rohs, 2009b). A spatially tracked mobile device on the interactive surface provides a means for input and allows the surface to imply the user identity from the device's identity. Imagine multiple users are using a multi-touch interactive tabletop in a Kinect-augmented room. Each user is holding a personal device in the hand. Each input action can be attributed to a particular user when the sensor fusion system matches this particular user's skeleton with the personal device. The increased stability of the hand position tracking with the proposed GP prior model-based sensor fusion approach can provide a reliable input action on the tabletop, thus enhance the user's interaction.

This type of sensor fusion system is of great importance for proxemic interactions. Imagine the proposed scenario in section 1.1 is a family environment. The members can be identified through their movements in everyday lives. Personalised content will be displayed when they approach the surface on the wall. Meanwhile, there is an indoor location-aware sensing application, such as a spatially aware display application. The user matching enables the system to use the acceleration sensed by a particular user's phone to compensate for the effects of position uncertainty (jitter) and the lag in this particular user's skeleton tracking sensed by the conventional Kinect system, giving the user a smoother, more responsive experience.

Therefore, the two applications, including (1) user matching and identification, and (2) position stabilisation and lag reduction, can be combined seamlessly in a proxemic interaction system. The proposed GP prior model-based sensor fusion system is of great benefit for improving the context sensing capability and increasing the usability of such context-aware sensing applications.

Appendix A

Acronyms

List of Abbreviations and Acronyms

Notation	Description	Page List
2D	Two-dimensional	23, 141
3D	Three-dimensional	7, 48
AHRS	Attitude and Heading Reference System	20, 21
ANOVA	ANalysis Of VAriance	147
API	Application Programming Interface	73
AR	Augmented Reality	1, 16, 70, 75
ARGP	AutoRegressive Gaussian Process	80, 94, 157
DOF	Degree-Of-Freedom	27
EKF	Extended Kalman Filter	8, 34, 72
GP	Gaussian Process	5, 8, 35, 36
GP-LVM	Gaussian Process Latent Variable Model	35
GPDM	Gaussian Process Dynamical Model	35

Notation	Description	Page List
GPR	Gaussian Process Regression	36
GPS	Global Positioning System	7, 19, 21
GPs	Gaussian Processes	1
HCI	Human-Computer Interaction	6, 22, 31
IMU	Inertial Measurement Unit	20
INS	Inertial Navigation System	20, 21
IR	InfraRed	19, 25
KF	Kalman Filter	7
KKF	Kinematic Kalman Filter	44
LED	Light-Emitting Diode	25
MEMS	Micro-ElectroMechanical System	26
MSE	Mean Square Error	147, 148, 154, 159
NED	Earth's North-East-Down	47
NITE	Natural Interaction Technology for End-user	46
NUI	Natural User Interface	17, 29
OpenNI	Open Natural Interaction	46, 73
RFID	Radio-Frequency IDentification	19
RMSE	Root Mean Square Error	97
SDK	Software Development Kit	73

Notation	Description	Page List
SHAKE	Sensing Hardware Accessory for Kinaesthetic Expression	46
UbiComp	Ubiquitous Computing	13
UDP	User Datagram Protocol	77, 142
UKF	Unscented Kalman Filter	8, 34, 72
VR	Virtual Reality	18
WLAN	Wireless Local Area Network	19, 77

Bibliography

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless networks*, *3*, 421–433.

Abowd, G. D., Ebling, M., Hung, G., Lei, H., & Gellersen, H.-W. (2002). Context-aware computing. *IEEE Pervasive Computing*, *1*, 22–23.

Accot, J., & Zhai, S. (1997). Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 295–302). ACM.

Accot, J., & Zhai, S. (1999). Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 466–472). ACM.

Ackad, C., Clayphan, A., Maldonado, R. M., & Kay, J. (2012). Seamless and continuous user identification for interactive tabletops using personal device handshaking and body tracking. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 1775–1780). ACM.

Aggarwal, J. K., & Cai, Q. (1997). Human motion analysis: A review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE* (pp. 90–102). IEEE.

Akamatsu, M., MacKenzie, I. S., & Hasbroucq, T. (1995). A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, *38*, 816–827.

Albert, W., & Tullis, T. (2013). *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

Alexiadis, D. S., Kelly, P., Daras, P., O'Connor, N. E., Boubekeur, T., & Moussa, M. B. (2011). Evaluating a dancer's performance using Kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia* (pp. 659–662). ACM.

Armesto, L., Tornero, J., & Vincze, M. (2007). Fast ego-motion estimation with multi-rate fusion of inertial and vision. *The International Journal of Robotics Research*, *26*, 577–589.

Ashby, W. R. (1957). *An Introduction to Cybernetics*. Chapman & Hall Ltd.

Aström, K. J., & Murray, R. M. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.

Azimi, M. (2012). Skeletal joint smoothing white paper. `http://msdn.microsoft.com/en-us/library/jj131429.aspx`. [Accessed: August 2014].

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, *21*, 34–47.

Azuma, R. T. et al. (1997). A survey of augmented reality. *Presence*, *6*, 355–385.

Bailly, G., Müller, J., Rohs, M., Wigdor, D., & Kratz, S. (2012). ShoeSense: a new perspective on gestural interaction and wearable applications. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (pp. 1239–1248). ACM.

Ballendat, T., Marquardt, N., & Greenberg, S. (2010). Proxemic interaction: designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces* (pp. 121–130). ACM.

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Barnard, L., Yi, J. S., Jacko, J. A., & Sears, A. (2005). An empirical comparison of use-in-motion evaluation scenarios for mobile computing devices. *International Journal of Human-Computer Studies*, *62*, 487–520.

Bau, O., & Mackay, W. E. (2008). OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (pp. 37–46). ACM.

Bennett, M., McCarthy, K., OModhrain, S., & Smyth, B. (2011). SimpleFlow: enhancing gestural interaction with gesture prediction, abbreviation and autocompletion. In *Human-Computer Interaction–INTERACT 2011* (pp. 591–608). Springer.

Bier, E. A., Stone, M. C., Pier, K., Buxton, W., & DeRose, T. D. (1993). Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (pp. 73–80). ACM.

Bilandzic, M., & Foth, M. (2012). A review of locative media, mobile and embodied spatial interaction. *International Journal of Human-Computer Studies*, *70*, 66–71.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Blažica, B., Vladušič, D., & Mladenić, D. (2013). MTi: A method for user identification for multitouch displays. *International Journal of Human-Computer Studies*, *71*, 691–702.

Bobick, A. F. (1997). Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *352*, 1257–1265.

Brown, R. G., Hwang, P. Y. et al. (1992). *Introduction to random signals and applied Kalman filtering* volume 3. John Wiley & Sons New York.

Cao, X., & Balakrishnan, R. (2006). Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proceedings of the 19th annual ACM symposium on User interface software and technology* (pp. 225–234). ACM.

Cao, X., Li, J. J., & Balakrishnan, R. (2008). Peephole pointing: modeling acquisition of dynamically revealed targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1699–1708). ACM.

Caron, F., Duflos, E., Pomorski, D., & Vanheeghe, P. (2006). GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. *Information Fusion*, *7*, 221–230.

Caruso, M. (1997). Applications of magnetoresistive sensors in navigation systems. *SAE transactions*, *106*, 1092–1098.

Casiez, G., Roussel, N., & Vogel, D. (2012). 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (pp. 2527–2530). ACM.

Chang, Y.-J., Chen, S.-F., & Huang, J.-D. (2011). A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, *32*, 2566–2570.

Chellappa, R., Roy-Chowdhury, A. K., & Kale, A. (2007). Human identification using gait and face. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (pp. 1–2). IEEE.

Conner, B., & Holden, L. (1997). Providing a low latency user experience in a high latency application. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (pp. 45–ff). ACM.

Conti, M., Zachia-Zlatea, I., & Crispo, B. (2011). Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security* (pp. 249–259). ACM.

Corke, P., Lobo, J., & Dias, J. (2007). An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, *26*, 519–535.

Crossan, A., McGill, M., Brewster, S., & Murray-Smith, R. (2009). Head tilting for interaction in mobile contexts. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 6:1–6:10). ACM.

Dabrowski, J., & Munson, E. V. (2011). 40 years of searching for the best computer system response time. *Interacting with Computers*, *23*, 555–564.

Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, *5*, 4–7.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, *16*, 97–166.

Dix, A. (2004). *Human computer interaction*. Pearson Education.

Ek, C. H., Torr, P. H., & Lawrence, N. D. (2008). Gaussian Process latent variable models for human pose estimation. In *Machine learning for multimodal interaction* (pp. 132–143). Springer.

El-Beltagy, M. A., & Wright, W. A. (2001). Gaussian Processes for model fusion. In *Artificial Neural Networks ICANN 2001* (pp. 376–383). Springer.

Erol, A., Bebis, G., Nicolescu, M., Boyle, R., & Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, *108*, 52–73.

Farella, E., OModhrain, S., Benini, L., & Riccó, B. (2006). Gesture signature for ambient intelligence applications: a feasibility study. *Pervasive Computing*, (pp. 288–304).

Feng, S., & Murray-Smith, R. (2014). Fusing Kinect sensor and inertial sensors with multi-rate Kalman filter. In *Data Fusion Target Tracking 2014: Algorithms and Applications (DF TT 2014), IET Conference on* (pp. 1–8).

Feng, S., Murray-Smith, R., & Ramsay, A. (2014). Position stabilisation and lag reduction with Gaussian Processes in sensor fusion system for user performance improvement, . Submitted.

Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, *36*, 39–49.

Foxlin, E. (1996). Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996* (pp. 185–194). IEEE.

Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, *25*, 38–46.

Fraden, J. (2004). *Handbook of modern sensors: physics, designs, and applications*. Springer.

Franklin, G. F., Powell, D. J., & Emami-Naeini, A. (2001). *Feedback Control of Dynamic Systems*. (4th ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.

Friedmann, M., Starner, T., & Pentland, A. (1992). Device synchronization using an optimal linear filter. In *Proceedings of the 1992 symposium on Interactive 3D graphics* (pp. 57–62). ACM.

Froehlich, P., Baillie, L., & Simon, R. (2008). FEATURE Realizing the vision of mobile spatial interaction. *interactions*, *15*, 15–18.

Fröhlich, P., Simon, R., Baillie, L., Roberts, J., & Murray-Smith, R. (2007). Mobile spatial interaction. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* CHI EA '07 (pp. 2841–2844). ACM.

Funk, M., Rozinat, A., Karapanos, E., Alves de Medeiros, A., & Koca, A. (2010). In situ evaluation of recommender systems: Framework and instrumentation. *International journal of human-computer studies*, *68*, 525–547.

Gallo, L., & Minutolo, A. (2012). Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique. *International Journal of Human-Computer Studies*, *70*, 287–300.

Gemeiner, P., Einramhof, P., & Vincze, M. (2007). Simultaneous motion and structure estimation by fusion of inertial and vision data. *The International Journal of Robotics Research*, *26*, 591–605.

Girard, A. (2004). *Approximate Methods for Propagation of Uncertainty with Gaussian Process Models*. PhD thesis, Department of Computing Science, University of Glasgow.

Girard, A., Rasmussen, C. E., Candela, J. Q., & Murray-Smith, R. (2003a). Gaussian Process priors with uncertain inputs – application to multiple-step ahead time series forecasting. In S. T. S. Becker, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15* (pp. 529–536). Cambridge, MA: MIT Press.

Girard, A., Rasmussen, C. E., Candela, J. Q., & Murray-Smith, R. (2003b). Gaussian Process priors with uncertain inputs-application to multiple-step ahead time series forecasting. *Advances in neural information processing systems*, (pp. 545–552).

Girolami, M. (2006). Bayesian data fusion with Gaussian Process priors: An application to protein fold recognition. In *Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology, PMSB*.

Gopalakrishnan, A., Kaisare, N. S., & Narasimhan, S. (2011). Incorporating delayed and infrequent measurements in extended Kalman filter based nonlinear state estimation. *Journal of Process Control*, *21*, 119–129.

Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., & Wang, M. (2011). Proxemic interactions: The new ubicomp? *interactions*, *18*, 42–50.

Grewal, M., Weill, L., Weill, L., & Andrews, A. (2007). *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Blackwell.

Grochow, K., Martin, S. L., Hertzmann, A., & Popović, Z. (2004). Style-based inverse kinematics. In *ACM SIGGRAPH 2004 Papers* SIGGRAPH '04 (pp. 522–531). ACM.

Guna, J., Humar, I., & Pogacnik, M. (2012). Intuitive gesture based user identification system. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on* (pp. 629–633). IEEE.

Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, *85*, 6–23.

Hall, E. T., & Hall, E. T. (1969). *The hidden dimension* volume 1990. Anchor Books New York.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, *1*, 139–183.

Hartikainen, J., & Sarkka, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian Process regression models. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on* (pp. 379–384). IEEE.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized Additive Models* volume 43. CRC Press.

Hazas, M., Scott, J., & Krumm, J. (2004). Location-aware computing comes of age. *Computer*, *37*, 95–97.

Hightower, J., & Borriello, G. (2001). A survey and taxonomy of location systems for ubiquitous computing. *IEEE computer*, *34*, 57–66.

Hinckley, K., & Wigdor, D. (2002). Input technologies and techniques. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, (pp. 151–168).

Hol, J., Schön, T., Luinge, H., Slycke, P., & Gustafsson, F. (2007). Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing*, *2*, 149–160.

Höllerer, T., & Feiner, S. (2004). Mobile augmented reality. *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK*, *21*.

Jagacinski, R. J., & Flach, J. (2011). *Control theory for humans: Quantitative approaches to modeling performance*. Taylor & Francis.

Jaimes, A., & Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, *108*, 116–134.

Jeon, S., & Tomizuka, M. (2007). Benefits of acceleration measurement in velocity estimation and motion control. *Control Engineering Practice*, *15*, 325–332.

Jeon, S., Tomizuka, M., & Katou, T. (2009). Kinematic Kalman filter (KKF) for robot end-effector sensing. *Journal of Dynamic Systems, Measurement, and Control*, *131*, 021010.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*, 401–422.

Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995* (pp. 1628–1632). IEEE volume 3.

Kadouche, R., Pigot, H., Abdulrazaka, B., & Giroux, S. (2010). Support vector machines for inhabitant identification in smart houses. *Ubiquitous Intelligence and Computing*, (pp. 83–95).

Kale, A., Sundaresan, A., Rajagopalan, A., Cuntoor, N. P., Roy-Chowdhury, A. K., Kruger, V., & Chellappa, R. (2004). Identification of humans using gait. *Image Processing, IEEE Transactions on*, *13*, 1163–1173.

Kalman, R. E. et al. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*, 35–45.

Kaufmann, B., & Ahlström, D. (2012). Revisiting peephole pointing: a study of target acquisition with a handheld projector. In *MobileHCI 2012* (pp. 211–220). ACM.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2011). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, .

Khoury, H. M., & Kamat, V. R. (2009). Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, *18*, 444–457.

Klette, R., & Tee, G. (2008). *Understanding human motion: A historic review*. Springer.

Ko, J., & Fox, D. (2009). GP-BayesFilters: Bayesian filtering using Gaussian Process prediction and observation models. *Autonomous Robots*, *27*, 75–90.

Ko, J., Klein, D. J., Fox, D., & Haehnel, D. (2007). GP-UKF: Unscented Kalman filters with Gaussian Process prediction and observation models. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (pp. 1901–1907). IEEE.

Kocijan, J., Girard, A., Banko, B., & Murray-Smith, R. (2005). Dynamic systems identification with Gaussian Processes. *Mathematical and Computer Modelling of Dynamical Systems*, *11*, 411–424.

Kocijan, J., Murray-Smith, R., Rasmussen, C. E., & Girard, A. (2004). Gaussian Process model based predictive control. In *American Control Conference, 2004. Proceedings of the 2004* (pp. 2214–2219). IEEE volume 3.

Kratz, S., & Rohs, M. (2009a). HoverFlow: Expanding the design space of around-device interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* MobileHCI '09 (pp. 4:1–4:8). ACM.

Kratz, S., & Rohs, M. (2009b). Unobtrusive tabletops: Linking personal devices with regular tables. In *Workshop Multitouch and Surface Computing at CHI'09*.

Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G., & Nischt, M. (2012a). PalmSpace: Continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* AVI '12 (pp. 181–188). ACM.

Kratz, S., Rohs, M., Reitberger, F., & Moldenhauer, J. (2012b). Attjector: an attention-following wearable projector. In *Kinect Workshop at Pervasive 2012*.

Krumm, J. (2009). *Ubiquitous computing fundamentals*. CRC Press.

Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, *48*, 140–150.

Larson, H., & Shubert, B. (1979). *Probabilistic Models in Engineering Sciences*. Probabilistic models in engineering sciences. Wiley.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian Process latent variable models. *The Journal of Machine Learning Research*, *6*, 1783–1816.

Lawrence, N. D. (2003). Gaussian Process latent variable models for visualisation of high dimensional data. In *NIPS* (p. 5). volume 2.

Lawrence, N. D., & Moore, A. J. (2007). Hierarchical Gaussian Process Latent Variable Models. In *Proceedings of the 24th International Conference on Machine Learning* ICML '07 (pp. 481–488). ACM.

Lee, D.-J., & Tomizuka, M. (2003). Multirate optimal state estimation with sensor fusion. In *American Control Conference, 2003. Proceedings of the 2003* (pp. 2887–2892). IEEE volume 4.

Leith, D. J., Heidl, M., & Ringwood, J. V. (2004). Gaussian Process prior models for electrical load forecasting. In *Probabilistic Methods Applied to Power Systems, 2004 International Conference on* (pp. 112–117). IEEE.

Li, F. C. Y., Dearman, D., & Truong, K. N. (2009). Virtual shelves: interactions with orientation aware devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (pp. 125–128). ACM.

Liang, J., Shaw, C., & Green, M. (1991). On temporal-spatial realism in the virtual reality environment. In *Proceedings of the 4th annual ACM symposium on User interface software and technology* (pp. 19–25). ACM.

Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *37*, 1067–1080.

Livingston, M. A., Sebastian, J., Ai, Z., & Decker, J. W. (2012). Performance measurements for the Microsoft Kinect skeleton. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE* (pp. 119–120). IEEE.

Llinas, J., Hall, D. L., & Liggins, M. E. (2009). *Handbook of Multisensor Data Fusion: Theory and Practice*. CRC Press.

Lu, X., Zhang, H., Wang, W., & Teo, K.-L. (2005). Kalman filtering for multiple time-delay systems. *Automatica*, *41*, 1455–1461.

Luinge, H. J. (2002). *Inertial Sensing of Human Movement*. PhD thesis, Enschede.

Luo, R. C., Chang, C. C., & Lai, C. C. (2011). Multisensor fusion and integration: theories, applications, and its perspectives. *Sensors Journal, IEEE*, *11*, 3122–3138.

MacKay, D. J. (1998). Introduction to Gaussian Processes. *NATO ASI Series F Computer and Systems Sciences*, *168*, 133–166.

MacKenzie, I. S., & Ware, C. (1993). Lag as a determinant of human performance in interactive systems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 488–493). ACM.

Madgwick, S. O., Harrison, A. J., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on* (pp. 1–7). IEEE.

Marquardt, N., Diaz-Marino, R., Boring, S., & Greenberg, S. (2011). The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 315–326). ACM.

Mehra, S., Werkhoven, P., & Worring, M. (2006). Navigating on handheld displays: Dynamic versus static peephole navigation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *13*, 448–457.

Moeslund, T. B., & Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, *81*, 231–268.

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, *104*, 90–126.

Morganti, E., Angelini, L., Adami, A., Lalanne, D., Lorenzelli, L., & Mugellini, E. (2012). A smart watch with embedded sensors to recognize objects, grasps and forearm gestures. *Procedia Engineering*, *41*, 1169–1175.

Motion, L. (2014). Leap Motion Controller. `https://developer.leapmotion.com/`. [Accessed: September 2014].

Mould, D., & Gutwin, C. (2004). The effects of feedback on targeting with multiple moving targets. In *Proceedings of Graphics Interface 2004* (pp. 25–32). Canadian Human-Computer Communications Society.

Murray-Smith, R., Johansen, T., & Shorten, R. (1999). On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In P. M. Frank (Ed.), *5th European Control Conference*. Springer.

Murray-Smith, R., & Pearlmutter, B. A. (2005). Transformations of Gaussian Process priors. In *Proceedings of the First International Conference on Deterministic and Statistical Methods in Machine Learning* (pp. 110–123). Berlin, Heidelberg: Springer-Verlag.

Nickel, K., & Stiefelhagen, R. (2003). Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on Multimodal interfaces* (pp. 140–146). ACM.

Niwa, S., Masuda, T., & Sezaki, Y. (1999). Kalman filter with time-variable gain for a multisensor fusion system. In *Multisensor Fusion and Integration for Intelligent Systems, 1999. MFI'99. Proceedings. 1999 IEEE/SICE/RSJ International Conference on* (pp. 56–61). IEEE.

Norrie, L., Koelle, M., Murray-Smith, R., & Kranz, M. (2013). Putting books back on the shelf: Situated interactions with digital book collections on smartphones. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* MUM '13 (pp. 44:1–44:2). ACM.

Norrie, L., & Murray-Smith, R. (2011). Virtual sensors: rapid prototyping of ubiquitous interaction with a mobile phone and a Kinect. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* MobileHCI '11 (pp. 25–28). ACM.

Oh, J., Jung, Y., Cho, Y., Hahm, C., Sin, H., & Lee, J. (2012). Hands-up: motion recognition using Kinect and a ceiling to improve the convenience of human life. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 1655–1660). ACM.

O'Hagan, A., & Kingman, J. F. C. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, *40*, 1–42.

Olguín-Olguín, D., & Pentland, A. (2010). Sensor-based organisational design and engineering. *International Journal of Organisational Design and Engineering*, *1*, 69–97.

Olwal, A., & Feiner, S. (2009). Spatially aware handhelds for high-precision tangible interaction with large displays. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (pp. 181–188). ACM.

OpenNI (2014). OpenNI. `http://www.openni.org/`. [Accessed: January 2014].

Ozyagcilar, T. (2012). Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. *Freescale semiconductor, AN*, *4248*.

Panger, G. (2012). Kinect in the kitchen: testing depth camera interactions in practical home environments. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 1985–1990). ACM.

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. S. (2007). Human computing and machine understanding of human behavior: a survey. In *Artifical Intelligence for Human Computing* (pp. 47–71). Springer.

Pantic, M., & Rothkrantz, L. J. (2003). Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, *91*, 1370–1390.

Park, S., Chung, K., & Jayaraman, S. (2014). Chapter 1.1 - wearables: Fundamentals, advancements, and a roadmap for the future. In E. Sazonov, & M. R. Neuman (Eds.), *Wearable Sensors* (pp. 1 – 23). Oxford: Academic Press.

Paton, M. A., & Ware, C. (1994). Passive force feedback for velocity control. In *Conference companion on Human factors in computing systems* (pp. 255–256). ACM.

Pavlovych, A., & Gutwin, C. (2012). Assessing target acquisition and tracking performance for complex moving targets in the presence of latency and jitter. In *Proceedings of Graphics Interface 2012* GI '12 (pp. 109–116). Canadian Information Processing Society.

Pavlovych, A., & Stuerzlinger, W. (2009). The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 187–196). ACM.

Pedley, M. (2013). Tilt sensing using a three-axis accelerometer. *Freescale Semiconductor Application Note*, .

Peñarrocha, I., Sanchis, R., & Romero, J. A. (2012). State estimator for multisensor systems with irregular sampling and time-varying delays. *International Journal of Systems Science*, *43*, 1441–1453.

Pérez-Cruz, F., Van Vaerenbergh, S., Murillo-Fuentes, J. J., Lázaro-Gredilla, M., & Santamaria, I. (2013). Gaussian Processes for nonlinear signal processing: An overview of recent advances. *Signal Processing Magazine, IEEE*, *30*, 40–50.

Picard, R. W. (2010). Emotion research by the people, for the people. *Emotion Review*, *2*, 250–254.

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer vision and image understanding*, *108*, 4–18.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Addison-Wesley Longman Ltd.

Qi, H., & Moore, J. B. (2002). Direct Kalman filtering approach for GPS/INS integration. *Aerospace and Electronic Systems, IEEE Transactions on*, *38*, 687–693.

Rahman, A. S. M. M., Hossain, M. A., & Saddik, A. E. (2010). Spatial-geometric approach to physical mobile interaction based on accelerometer and IR sensory data fusion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, *6*, 28.

Ramakers, R., Vanacken, D., Luyten, K., Coninx, K., & Schöning, J. (2012). Carpus: a non-intrusive user identification technique for interactive surfaces. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* UIST '12 (pp. 35–44). ACM.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Reece, S., & Roberts, S. (2010). An introduction to Gaussian Processes for the Kalman filter expert. In *Information Fusion (FUSION), 2010 13th Conference on* (pp. 1–9). IEEE.

Rehg, J., & Kanade, T. (1994). Digiteyes: Vision-based hand tracking for human-computer interaction. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on* (pp. 16–22). IEEE.

Richardson, G. P. (1999). *Feedback thought in social science and systems theory*. Pegasus Communications, Inc.

Robinson, S., Jones, M., Williamson, J., Murray-Smith, R., Eslambolchilar, P., & Lindborg, M. (2012). Navigation your way: from spontaneous independent exploration to dynamic social journeys. *Personal and Ubiquitous Computing*, *16*, 973–985.

Roetenberg, D. (2006). *Inertial and magnetic sensing of human motion*. University of Twente.

Roetenberg, D., Luinge, H., & Slycke, P. (2009). Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors. Xsens Technologies white paper.

Rofouei, M., Wilson, A., Brush, A., & Tansley, S. (2012). Your phone or mine?: fusing body, touch and device sensing for multi-user device-display interaction. In *Proceedings of the*

2012 ACM annual conference on Human Factors in Computing Systems (pp. 1915–1918). ACM.

Rohs, M., & Essl, G. (2006). Which one is better?: information navigation techniques for spatially aware handheld displays. In *Proceedings of the 8th international conference on Multimodal interfaces* (pp. 100–107). ACM.

Rohs, M., & Oulasvirta, A. (2008). Target acquisition with camera phones when used as magic lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1409–1418). ACM.

Rohs, M., Oulasvirta, A., & Suomalainen, T. (2011). Interaction with magic lenses: real-world validation of a Fitts' law model. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2725–2728). ACM.

Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural computation*, *11*, 305–345.

Rusdorf, S., & Brunnett, G. (2005). Real time tracking of high speed movements in the context of a table tennis application. In *Proceedings of the ACM symposium on Virtual reality software and technology* (pp. 192–200). ACM.

de Sá, M., & Churchill, E. (2012). Mobile augmented reality: Exploring design and prototyping techniques. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* MobileHCI '12 (pp. 221–230). ACM.

Salber, D., Dey, A. K., & Abowd, G. D. (1999). The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 434–441). ACM.

Särkkä, S., Solin, A., & Hartikainen, J. (2013). Spatio-temporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, *30*, 51–61.

Sato, Y., Saito, M., & Koike, H. (2001). Real-time input of 3D pose and gestures of a user's hand and its applications for HCI. In *Virtual Reality, 2001. Proceedings. IEEE* (pp. 79–86). IEEE.

Saunders, J. A., & Knill, D. C. (2003). Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental Brain Research*, *152*, 341–352.

Saunders, J. A., & Knill, D. C. (2004). Visual feedback control of hand movements. *The Journal of neuroscience*, *24*, 3223–3234.

Schmidt, A. (2000). Implicit human computer interaction through context. *Personal Technologies*, *4*, 191–199.

Schssel, F., Honold, F., & Weber, M. (2013). Adaptive multimodal HCI with uncertain data by collaborative fission and fusion. In C. Stephanidis (Ed.), *HCI International 2013 - Posters Extended Abstracts* (pp. 372–375). Springer Berlin Heidelberg volume 373 of *Communications in Computer and Information Science*.

Shahidi, S., Mazrooei, P., Esfahani, N., & Saraee, M. (2010). Proximity user identification using correlogram. *Intelligent Information Processing V*, (pp. 343–351).

Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface*. (4th ed.). Pearson Addison Wesley, USA.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., & Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*, 116–124.

Siciliano, B., & Khatib, O. (2008). *Springer handbook of robotics*. Springer.

Smith, D., & Singh, S. (2006). Approaches to multisensor data fusion in target tracking: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, *18*, 1696–1710.

Smyth, A., & Wu, M. (2007). Multi-rate Kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring. *Mechanical systems and signal processing*, *21*, 706–723.

Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., & Rasmussen, C. E. (2003). Derivative observations in Gaussian Process models of dynamic systems. In S. T. S. Becker, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15* (pp. 1033–1040). Cambridge, MA: MIT Press.

Stanton, N. A., Walker, G. H. et al. (2013). *Human factors methods: a practical guide for engineering and design*. Ashgate Publishing, Ltd.

Strachan, S., & Murray-Smith, R. (2009). Bearing-based selection in mobile spatial interaction. *Personal Ubiquitous Comput.*, *13*, 265–280.

Strachan, S., Murray-Smith, R., & O'Modhrain, S. (2007). BodySpace: inferring body pose for natural control of a music player. In *CHI'07 extended abstracts on Human factors in computing systems* (pp. 2001–2006). ACM.

Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, *45*, 52–57.

Thrun, S., Burgard, W., Fox, D. et al. (2005). *Probabilistic robotics* volume 1. MIT press Cambridge.

Titterton, D., Weston, J., & of Electrical Engineers, I. (2004). *Strapdown Inertial Navigation Technology, 2nd Edition*. IEE radar, sonar, navigation, and avionics series. Institution of Engineering and Technology.

Tuck, K. (2007). Tilt sensing using linear accelerometers. *Freescale Semiconductor Application Note AN3107*, .

Turner, R. D. (2012). *Gaussian Processes for State Space Models and Change Point Detection*. PhD thesis, University of Cambridge.

Urtasun, R., Fleet, D. J., & Fua, P. (2006). 3D people tracking with Gaussian Process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (pp. 238–245). IEEE volume 1.

Vasudevan, S. (2012). Data fusion with Gaussian Processes. *Robotics and Autonomous Systems*, *60*, 1528–1544.

Velloso, E., Bulling, A., & Gellersen, H. (2013). Motionma: Motion modelling and analysis by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1309–1318). ACM.

Vera, L., Gimeno, J., Coma, I., & Fernández, M. (2011). Augmented mirror: interactive augmented reality system based on Kinect. In *Human-Computer Interaction–INTERACT 2011* (pp. 483–486). Springer.

Vogel, D., & Balakrishnan, R. (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (pp. 137–146). ACM.

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

Wagner, J., Nancel, M., Gustafson, S. G., Huot, S., & Mackay, W. E. (2013). Body-centric design space for multi-surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1299–1308). ACM.

Wan, E. A., & Van Der Merwe, R. (2001). The unscented Kalman filter. *Kalman filtering and neural networks*, (pp. 221–280).

Wang, C., Chen, W., & Tomizuka, M. (2012). Robot end-effector sensing with position sensitive detector and inertial sensors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (pp. 5252–5257). IEEE.

Wang, J. M., Fleet, D. J., & Hertzmann, A. (2008). Gaussian Process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*, 283–298.

Ware, C., & Balakrishnan, R. (1994). Reaching for objects in VR displays: lag and frame rate. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *1*, 331–356.

Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, *7*.

Welch, G., & Bishop, G. (1997). SCAAT: Incremental tracking with incomplete information. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH '97 (pp. 333–344). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Wikipedia (2014). Kinect. `http://en.wikipedia.org/wiki/Kinect`. [Accessed: April 2014].

Williams, C. K. (1998). Prediction with Gaussian Processes: From linear regression to linear prediction and beyond. In *Learning in graphical models* (pp. 599–621). Springer.

Williams, C. K., & Rasmussen, C. E. (1996). Gaussian Processes for Regression. In *Advances in Neural Information Processing Systems 8* (pp. 514–520). MIT press.

Williamson, J. (2006). *Continuous Uncertain Interaction*. PhD thesis, Department of Computing Science, University of Glasgow.

Williamson, J., Murray-Smith, R., & Hughes, S. (2007). Shoogle: excitatory multimodal interaction on mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems* CHI '07 (pp. 121–124). ACM.

Wilson, A. D. (2007). Sensor-and recognition-based input for interaction. In A. Sears, & J. Jacko (Eds.), *The Human Computer Interaction Handbook*. Taylor & Francis.

Woodman, O., & Harle, R. (2008). Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 114–123). ACM.

Yee, K.-P. (2003). Peephole displays: pen interaction on spatially aware handheld computers. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1–8). ACM.

Zhu, R., & Zhou, Z. (2004). A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *12*, 295–302.

Index