Abstract

Gestures have a lot of potentials as a natural interaction method that can enrich the interaction between humans and ubiquitous environments. The lack of ordinary devices in ubiquitous environments like the keyboard and mouse make researchers work on utilizing hand gestures for interaction. However, hand gestures could change according to the situation it is performed. This situation could include the variability of location to do gestures and the position of doing the gestures, device interacting with, application to control and user preferences. Furthermore, social environment of people and activity could change the gesture shapes. In this research, we study the effect of context parameters on the gesture shapes and developed a system that can support people by appropriate gesture profiles according to their context. We conducted experiments to see the effect of changing position on the gesture shapes. We also applied some social parameters on the subjects while they were doing the experiments. The results show us that when subjects change their situation, most of the subjects change some of the gestures they do not like or fail to perform. Moreover, when subjects adjusted their gestures they can interact more accurate with their environment.

Everyday objects exist around the user as pens, bottles, books, headphones, balls and cellular phones. Augmenting everyday objects with sensors have a potential to convert objects into input devices. We think that allowing users to do gestures in ubiquitous environments through objects could inspire people to develop their own smart appliances. Hence, we extended our work to include the objects around the user as one of the context parameters that can change the gesture shapes. People can stick sensor to the objects then a profile will be downloaded to the user according to his context and the object he selected for interaction. We conducted an experiment to see the effect of objects on the gesture shapes and also the system proposed profiles compared to predefined profiles. The results showed that there is some variability between the objects and their usage to control devices. The more functions to control, the more people, are not satisfied by the predefined gestures. We found that, objects gestures are faster and less in errors than hand gestures, if and only if appropriate objects were select. Based on users feedback that showed an agreement on using objects around them enrich their interaction with applications like using balls, pens and headphones.

A hand gesture that is sensitive to angles of a hand wrist is called “Tilt gestures”. Tilting gestures were studied as an interaction method in ubiquitous environments where it is difficult to have ordinary input devices. By combining tilt gestures with objects, users can control applications and devices. We proposed a system that captures people context and provides them tilt gestures for objects they interact with. We discuss the system settings, gesture recognition and methods for mapping directions using a 3D accelerometer to capture the tilting gestures. We explored application fields and areas were tilting gestures could be useful by applying it to map/web browsers, media players, PTZ camera protector and special Japanese text tool. We present the mapping techniques for each application commands and tilt gestures based on objects they use for interaction. We addressed the classification of suitable applications for tilting interfaces as direct mapping, tilt sequence and mouse movements. An experiment to evaluate the interaction with a menu representing GUI items was conducted. The results show that tilt interface was close to mouse in accuracy and speed for up to 8 targets. Even when the number of targets increased up to 16 targets tilt gestures still perform much better than hand gestures. Furthermore, we found that tilt gestures were easy to memorize by most of the users. Some appropriate objects that were carried by hands pretended to have good accuracy results compared to objects carried by fingers only.

Acknowledgments

I am heartily thankful to Professor Jiro Tanaka, my thesis supervisor, for his many valuable suggestions, precise directions, and kind encouragement. I also thank Professor Shin Takahashi, also my thesis supervisor, for his useful suggestions and kind advice. I also greatly thank Professor Kazuo Misue and Professor Buntarou Shizuki for their kind advice and suggestions regarding this research. I would like to thank Professor Yukio Fukui, Professor Hideaki Kuzuoka and Professor Hotaka Takizawa the other members of my thesis committee, for their many useful comments on this research. I am also grateful to all the members of the IPLAB, Interactive Programming Laboratory, University of Tsukuba, for giving me many opportunities to discuss my research with them and helping me in my experiments.

I would like to acknowledge the financial support provided by Helwan University, Ministry of Higher Education, Egypt. I would like to express my sincere gratitude to my former supervisors at Helwan University, Prof. Mostafa Sami. Without their support, I could not get this opportunity to study in Japan. I would like also to thank Dr. Omar Hamdy and Dr. Waleed Youssef for their continuous encouragement.

I would like to thank my wife (Dr. Ghada Khoriba) for her continuous support, my son Omar and my sister (Dr. Hanan Atia). Finally, I extend my sincere gratitude to the soul of my parents, Mohamed Ezzat and Wafaa Elzorkani for all the efforts they did for me and their wishes and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Mark Weiser in 1988 has defined the term “ubiquitous computing” as a method of enhancing computer use by making many computers available throughout the physical environment, but make them effectively invisible to the user [1]. Since then, the enormous development of wearable devices and small sensors have affected the field of human computer interaction in ubiquitous environments. Sensors now are embedded in most of the new communicating devices as cellular phones, PDA, digital cameras and notebooks. This age of digital life inspired many researchers to develop new approaches to replace the normal WIMP (windows, icons, menus and pointing devices) interfaces. People like to use the normal and intuitive interfaces to interact with their environments. However, in many situations it is difficult to use the normal ways because of people different situations. Such situations could include activities, social implications and locations they are interacting inside. The early evolution of a “computer walls” to “calm computing” concept discussed by Brown et al. [2] as they show the main aims of the area of “ubi-com”. The main aim is to allow people to focus on the human to human communication and disappearing into the configurations of computers allowing people to focus on the big images of their goals.

So conceptually computers, sensors and intelligent systems will be embedded in every object around us, including homes, work locations, tables and even body parts in an unobtrusive way. This can make humans living simpler through the managing and controlling of information in easy ways.

1.1 Context aware environments

Nowadays, the term context aware environments become popular, context of users consider many parameters like the location, identity, time and activity [3]. In this work, we give focus to some parameters of context awareness.

First, location aware systems that have been discussed in the form of indoor locations like museums using RFID (radio frequency identification) in the work of Kuo et al. [4] and Tsai et al. [5]. Location is one of the most important parameters for context awareness as services that can be delivered to users can be extremely different from location to another [6]. Context aware web services have been presented by Markus et al. [7], they infer the user location using GPS and preserve user details and devices preference. They adapt the web service according to user location.

Second, the device the user is interacting with like smart devices, notebooks, personal computers and projected displays. Lai et al. [8] presented a cross-heterogeneous remote control system for home devices to relieve the problem of excessive remote controllers. Jea et al. [9] proposed a method to control the accessing of shared public devices through the usage of different context parameters. If the system can know which device the user is using, then It can automatically change the interface according to the carried device. This idea was presented by the work of Artail et al. [10].

Third, the applications existing in the user environment early presented by the work of Schilit et al. [11] and surveyed by Baldauf et al. [12]. An application that can take advantage of location aware systems by controlling data to display or application configuration has been presented by Hess et al. [13]. Owen et al. [14] showed the usage of location aware and user preferences in an E-Learning domain and suggest the usage of web services. Another application for supporting reminders called “CyberMinder” presented by Dey et al. [15], this application can remind people about their valuable reminders according to their context. Furthermore, Budzik et al. [16] presented a system which gathers contextual information in the documents to pro-actively recommend related documents from distributed repositories upon context. Joanna et al. [17] studied the customization of application per user preferences.

Forth, the objects around the people are good candidates for interactions as they are always existing and many people are carrying several objects with them, so they represent candidates for enriching interaction in the people environment. Such objects can enrich the user environment through interaction models scenarios to form a term of smart objects [18]. Jakob [19] studied the context awareness of application that can be applied in hospital systems. They use RFID (radio frequency identifiers) to objects to form smart bed and smart pill container.

Fifth, the user himself, personalization in the context aware environments was presented by Zimmermann et al. [20]. Indoor location awareness has been studied by Carlos et al. [21] by using Bluetooth infrastructure to locate users inside ubiquitous museum. In their work they preserve the user preferences and graphical user interface preferences into centralized data repository. The social awareness could be crucial as to know which users are collaborating with each other in environments as presented by Idrus et al. [22]. Personalization was discussed in means of behaviour of people and addressed by Oulasvirta et al. [23]. Kappel et al. [24] showed the customization of ubiquitous web applications as a survey study on the main parameters affecting customization and personalization of applications.

1.2 Interaction in ubiquitous environments

Interaction in ubiquitous environments can sustain from the absence of conventional input devices like mouse and keyboards. Hence, gestures are one of the means to interact in ubiquitous environments as they represent a natural way of communication between people. From this point, researchers in this area aim to invent tools, devices and interfaces that can facilitate retrieval and commanding of information anywhere and anytime. Figure 1.1 shows a sample of ubiquitous computing applications and concepts. Camera techniques to select some items from a projected display was presented in the work of Kim et al. [25] in Figure 1.1 (a). Nakamura et al. [26] used a LED to control menu items and browsing images / Internet on big display screen shown in Figure 1.1 (b). Wong et al. [27] used everyday objects to link them with files and documents for easy retrieval of information in Figure 1.1 (c). Higuchi et a. [28] used ART toolkit to perform gestures to preserve people privacy in locations with PTZ (pan, tilt, zoom) camera installed in Figure 1.1 (d). Nagao et al. [29] showed a mirror appliance for displaying clothes recommendations based on the weather temperature shown in Figure 1.1 (e). Hand gestures can be captured in 3D space through the usage of special interacting surfaces like cylinders presented by work of Naito et al. [30] shown in Figure 1.1 (f).

Figure 1.1: Ubiquitous computing concept and application sample.

Conventional methods also require a stationary interaction situation. If people like to use keyboard or mouse, their hands should be fixed on some surface to input some data. People who used to move frequently and ride transportation also have some difficulties to have a stationary position to enter data. The use of hand gestures also brought some challenges to the area of gesture interaction. First, the gestures could not be convenient to all situations specially those of public spaces. Some gestures have social threaten implications, which cannot be accepted by other persons in the people context. Second, some capturing hand gesture techniques using a camera involve a lot of privacy issues, as people do not want to feel monitored by some obtrusiveness systems involving in their personal life's. Third, using hand gestures for a long time periods could make hand tremor and pain for users. This pain makes people fail to practice or hate to use some gestures.

1.3 Gesture types

We classified human gestures into three main types, hand gestures, object gestures and tilt gestures. Figure 1.2 shows the three types of gestures. Hand gestures shown in Figure 1.2 (a) are gestures done by the movement of four elements of the human body (arm, elbow, forearm and wrist). The four elements can work together or separately to form a complete gesture. Hand gestures are well known in expressing commands or actions. For example, the American water ski association [31] in 1964, has devised a set of hand signals to facilitate the communication between the skier and boat driver. This signals are now universally used to command the boat driver.

Object gestures are depending on moving objects carried on hands, head or legs. Figure 1.2 (b) shows a sample of object gestures that can be done by mug cup, doors and table tennis rackets. Such gesture people normally do as opening or closing doors. The movement of the door is called an object gesture. Objects by their nature have always fixed attributes, like functionality, shape and other parameters. This functionality is a natural behavior of an object, and it sometimes accompanied with some movements. Those movements are done naturally by people and done in many situations intentionally or unintentionally. Intentioned gestures are those that people did using objects to achieve some goal, like moving a pen to write something on the board. However, the unintended gestures are the shakes of the pen that is done while writing on the board. The unintended gestures done by people using everyday objects are candidates for further research to understand their feasibility for interaction usage in ubiquitous environments.

Tilt gestures are a special case of hand gestures as they depend on the movement of the hand wrist only. Tilt gestures can be used to express directions as in Figure 1.2 (c), directions for left, normal, down and vertical directions. Moreover, tilt gestures are useful for some types of objects, such as objects that can be easy carried by hands and have cubed or spherical shapes like balls, boxes and bottles. Objects and tilt gestures have a lot of common areas that both share and affect each other. A gesture done while tilting tennis table racket to spin the ball must use a tilt gesture to detect the spinning angle.

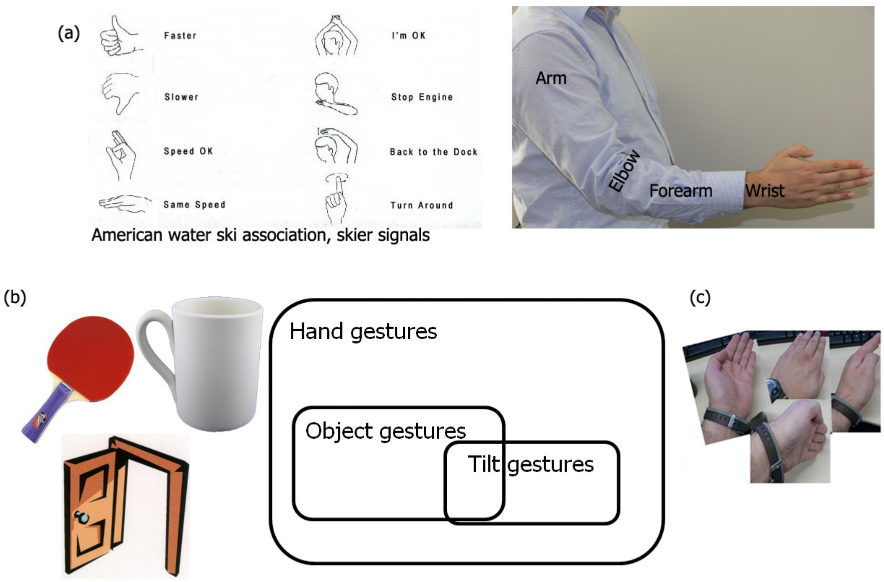


Figure 1.2: (a) Hand gestures, (b) Object gestures and (c) Tilt gestures.

More gestures can be done by human gestures like the neck, head and legs as they are also parts of the human body and can be used for activity recognition [32] and [33]. However, those gestures sometimes are difficult to use for interaction because of either, they have less space to move or they are not practical for everyday interaction.

1.4 Goal and approach

The main aim of this work is to study the effect of people context on gesture shapes and profiles. Hence, we want to offer people sufficient gesture profile for their context environment. We want also to study if the hand, object and tilt gestures can enrich the interaction between humans and computers. The effect of using gestures to control devices in the people context can make them think about new smart appliances and applications.

1.5 Dissertation organization

The presented dissertation is structured as follows. Chapter 1 is an introduction that defines the scope of the dissertation. Chapter 2 presents the using of hand gestures for interaction in ubiquitous environments. Chapter 3 explains the usage of objects to interact in ubiquitous environments. Chapter 4 describes our proposed system for capturing and interacting with tilt gestures. Chapter 5 concludes the dissertation and presents future work.

Chapter 2

Interaction with hand gestures in ubiquitous environments

Human hand gestures are an intuitive way for humans to express their feelings and to interact with several objects in daily life. Hand gestures can be used as a tool for interacting with different applications like media player, presentation viewer...etc. Interaction can be done through predefined hand gestures. However, it is difficult to fit all users and users must adjust some of their hand gestures. The interaction can be affected negatively by some parameters like users preferences, interaction device, or application or combination of those parameters. Another parameters that can have a direct impact on users so that they change their hand gesture pattern are position and social awareness.

First, the position of the user while customizing his hand gesture. A predefined hand gesture, such as moving the elbow towards the body in a standing position, will be difficult to do when the user is sitting down on a sofa or sitting at a table because of the limited space. Second, is the social implication of the hand gesture, some hand gestures may scare, annoy, or disturb other people.

For example, user A is sitting on a sofa beside person B. If user A moves his or her arm towards person B (someone A knows), Person B might become annoyed and interpret the gesture as a minor interruption. A more complex reaction might occur if there is no relationship between people such as in a public space. Such a gesture may scare the other person. The main idea behind this research is to provide users with enough appropriate hand gesture profiles for any given situation.

2.1 Related work

The study of gesture recognition with a presentation viewer application was shown in [34]. They show an active region for starting and ending gesture interaction. Also, they point out that gestures can be useful in crowded or noisy situations, such as in a stock exchange or manufacturing environment. Head and hand gestures have been used for limited interactions as demonstrated in by Keates et al. [35]. They discussed the problem of learning gestures and showed the importance of customization. Kurze et al. [36] presented personalization of multi-modal applications as a design approach. They focus on implicit and explicit customization of systems according to a user's preferences. Kawsar et al. [37] presented customizing the proactive applications preferences in a ubiquitous environment. They present customization in many levels of artifact, action, interaction, and timing preferences.

There are many studies that focus on reusing current applications and interfaces in context-aware environments. Nakajima [38] proposed a model for displaying a GUI on multiple output devices. He/She also developed a way for selecting devices according to users location. Ronkainen et al. [39] studied the usability of hand gestures in different ubiquitous environments. They conducted a survey on the social implications of hand gestures in public spaces. Moreover, they present a tap gesture for interacting with mobile devices as a type of socially acceptable hand gesture. They point out that there are gestures that are perceived as being threatening in public spaces. Rico et al. [40] studied the various difference of accepting gestures by conducting Internet survey to stand on the difference of accepting gesture shapes. Their results do not show clear image on what parameters could affect the gesture shapes. However, they point out the variability of accepting gestures according to subject level of experience.

The information technology revolution in the last ten years had a direct impact on the size and performance of wireless sensors. Some of these sensors have been used for inferring the context of user and providing ubiquitous services. Recently, there have been many multifunction sensors that have been embedded in small devices [41][42]. In this research, we used the common Wii remote [43] and WAA006 [44] sensors as an interacting device in ubiquitous environments. One of the advantages of using 3D accelerometers to capture the hand gestures is that they do not require special environment setup, only Bluetooth connection. Furthermore, the users are not required to face camera or some setup, so they have more freedom in doing gestures in different positions.

2.2 Motivations of hand gestures

One of the main challenges of interaction in a ubiquitous environment is the use of hand gestures for interacting with day to day applications. It is difficult to have a universal gesture shapes that can fit everybody, because gestures are a product of users imaginations and experiences levels. Kela et al. [45] showed from their study on user gesture types for a VCR controlling tasks that users can have different gestures for each task, which lead to the importance of personalizing the gestures. The study also concludes that gesture commands can assist users in natural interaction, especially for commands of spatial association. Furthermore, some guidelines were designed by Zoltan et al. [46] to produce better gesture profiles. They pointed out the ubiquity of hand gestures as gestures profiles should be available everywhere around the user. The unobtrusiveness of gestures so people should not feel that systems capturing gestures are interrupting their interaction with the environment. Adaptivity or adjusting gestures that users do not like while interaction and limiting or expanding functions associated with gestures. Finally, simplicity of interaction interface and minimum number of training gestures for interaction as people get bored from long training for recognizing their gestures.

A typical scenario for how gestures can be changed according to user's context is for a user in the train, and he wants to interact with his PDA to play next song. He/she has limited gestures because the train is crowded and not all gestures will be convenient for people around. The same user goes to the train station, and he wants to interact with kiosk and public display. In this case, he/she has more enough space to do more gestures with different shapes. We investigated the effect of context parameters on the gesture shapes while users are interacting in ubiquitous environments. Furthermore, we studied the effect of social parameters while interacting.

2.3 Proposed UbiGesture system

We presented a system called ubiquitous gestures "UbiGesture" as an architecture for users who frequently change locations while interacting in ubiquitous environments. The

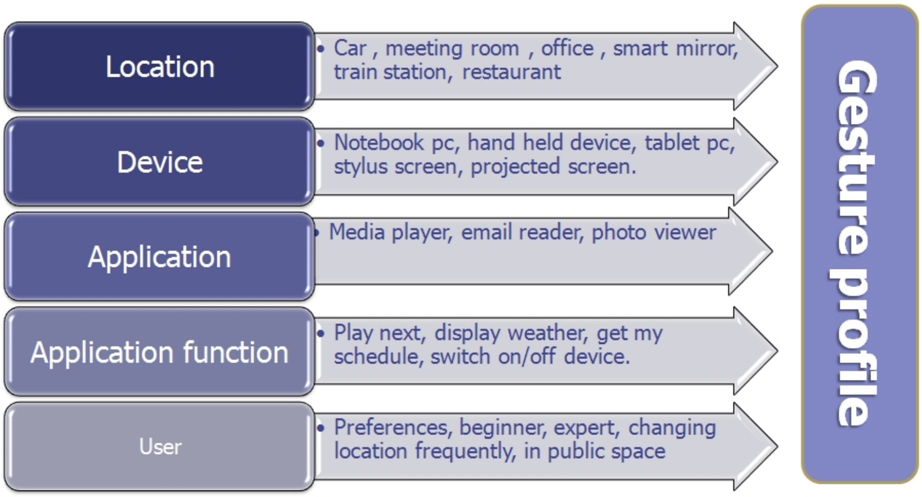


Figure 2.1: Gesture profile and context parameters

architecture enables applications to be operated by using hand gestures. UbiGesture is based on combining user preferences, location, input/output devices, applications, and hand gestures into a single gesture profile.

A gesture profile is defined by user's context parameters shown in Figure 2.1. UbiGesture packs user context and his gestures, save it to the network, then whenever the user changes his context it will download appropriate profile.

The location is important as it represents the different of the social environment for people for example, being in office is different from being in public transportation or plane. Thus, we think it is necessary to keep tracking of people location outdoor or indoor. Devices like hand-held devices or large display projected screens could also affect the gestures of user. We think that gestures help users to execute commands faster with large display screen where it is difficult to reach menu items. Furthermore, gestures could also be helpful to execute commands on small size devices while using a touch screen is hard as menu items are small to select. We think also that applications like media players, email reader and photo viewers can have different gesture shapes. It was shown by Dachselt et al. [47] as they used a "throw and tilt gestures" to make natural interaction with big display screen. Application functions represent the core of hand gesture systems as they are the direct mapping of hand gestures to real commands that will be executed. People do not like to use all the application functions all the time. However, they like to personalize those functions according to their context. In the car while driving, they want to answer or reject

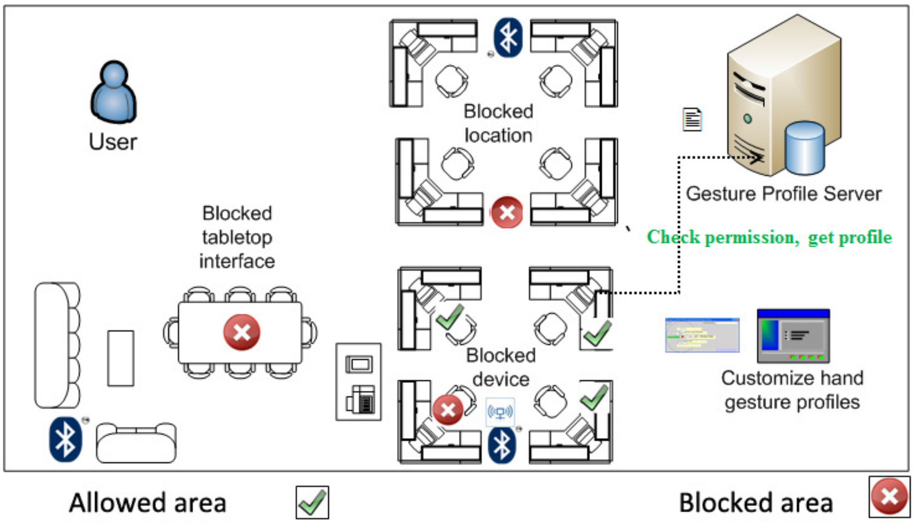


Figure 2.2: UbiGesture system overview and Bluetooth infrastructure in the office prototype

calls and check traffic jam of streets. However, if they are not driving they might like to check email, messages, and schedules. Finally, hand gestures are sometimes a product of people imaginations and can have a variety of pattern shapes. Some parameters like people experience level or background could affect the gesture shapes. People who used to play entertainment games using Wii remote controllers, PS move would like to use the hand gesture to control all devices around with innovative gestures. Conversely, users with no experience would like to use predefined profiles and adjust a couple of gestures they do not like.

2.4 System architecture

UbiGesture is used for profiling hand gestures of users depending on the context parameters they exist in. In this research, an inexpensive Bluetooth infrastructure was used because this technology is now embedded in most hand held devices. Sanchez et al. [48] used this technique for commercial advertising based on location. Figure 2.2 shows an overview of the UbiGesture system. In each location, a server is installed to manage the identities of the users entering locations. We assume that every user is carrying some Bluetooth device with him, so the server captures the mac address and identifies users. The server then determine what devices the user is allowed to access in this area. Some areas will be blocked for user and other's will be allowed. After that, the server will check what

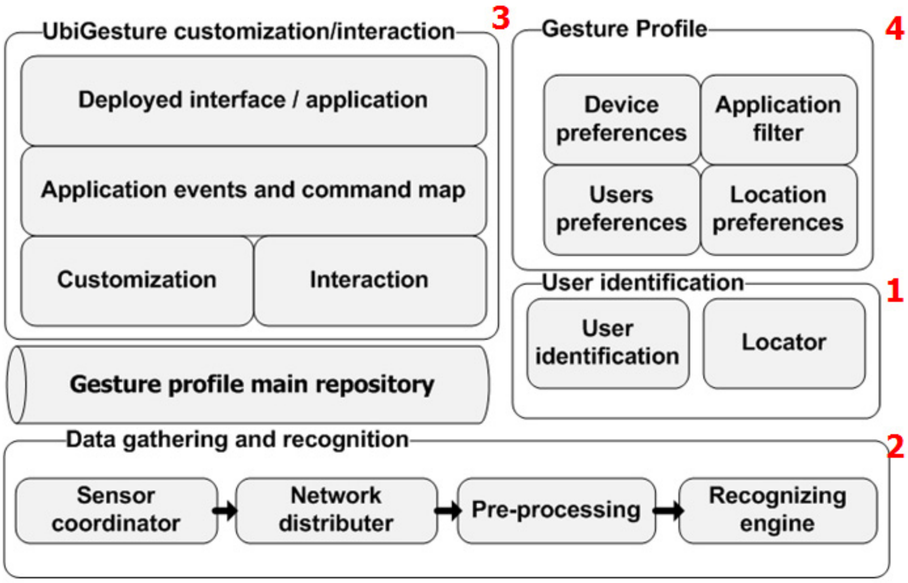


Figure 2.3: UbiGesture System architecture

device the user intended to interact with and connect to a server to download his/her profile. Furthermore, an application to customize gestures will be displayed on the interacted device, in case he/she wants to adjust their gesture profile.

There is a need for different level of users to operate the UbiGesture system (administrator, developer, and regular users). An administrator's main role is managing resources and granting permissions to each user. A developer enables an application to be customized by defining keyboard shortcuts or application programming interfaces (APIs). A regular user has to define his or her gesture profile and UbiGesture stores it over the network storage.

UbiGesture was built using four main modules each module is responsible for some functions. Figure 2.3 shows the system architecture modules of UbiGesture. The first module is for user locating and identifying. It is also responsible for checking permissions of users in a certain location. The second module is responsible for collecting data from sensors, pre-process it and send it to recognition engine that compares it with template of gestures. The third module is responsible for executing commands matched by the gesture performed by the user. It is also responsible to manipulate and update stored hand gestures. Furthermore, it captures video feedback for the user doing the gesture and show this video to the users in order to memorize gestures they have performed. Finally, the gesture profile itself which is stored in a central data store accessed through network storage. This module

Figure 2.4: Wii remote used for capturing gestures

collect all the context parameters (location, device, application, application functions and user) and store it for further retrieval.

2.4.1 Wireless Sensor

The revolution in the information technology in the last ten years has directly impact on the size and performance of wireless sensors. Some of these sensors used for inferring the context of the user and then can provide ubiquitous services. The problem of the small size sensors is its battery life in addition to its expensive price. In this research we used Wii [43] remote controller for capturing hand gestures. Figure 2.4 shows Wii remote controller. Wii remote is the primary controller for Nintendo's Wii console, it cost is about 40 US dollars. The main advantage of Wii remote is its embedded 3D accelerometer and optical sensor that can be used for motion and position tracking. Wii remote has been used before in applied virtual reality applications by Sreeram et al. [49], with volumetric medical data by Luigi et al. [50]. The Wii remote acceleration range is ± 3 G and has been used in many studies for hand-gesture recognition [51] [52]. The Wii remote controller has been suggested as an interaction device for entertainment purpose and some music and art applications [53] [54].

2.4.2 Identifying users

UbiGesture assumes that there is a Bluetooth location manager server called "Locator" in each location, which is a low-specification desktop PC with a Bluetooth dongle. Arjan et al. [55] used Bluetooth infrastructure and presence server for location and presence aware services. The locator hosts an application that reads a user's identification by continuously searching for a user's device's media access control (MAC) address within a 30 second interval. Each user may have one or more identification devices such as a hand held mobile device or a Bluetooth sensor. The locator grants access to users for some devices in one location, and it logs the user's entrance and exit time from the location. Whenever users exist in location covered area, it flags a message for all the permitted devices to download the user gesture profile.

User can have access to more than one device in one location. Therefore user identification is required before a device can be used. Interaction with devices is usually initialized using a keyboard, mouse, or the stylus and touch panel. We believe that hand gestures could be similar to someone's unique hand-written signature, by which a user can access a device by entering their device hand gesture signature. If the user wants to use a plasma display, he/she must enter hand gesture for starting this device; another gesture will be required for a large screen display and so on. If there are two or more users in one place and both have been granted access to the same device, the User identification module locks the device to be used with only a single user.

2.4.3 Data gathering and sensors used

The user has to press the Wii remote's 'A' button to start recording a gesture and release the button to stop the recording. Hand gesture h can be represented as a sequence of accelerometer sensor readings. A reading at time t can be represented as $A_t = x_t, y_t, z_t$. A sensor coordinator module will connects the Wii remote as a human-computer interface device with the locator. The locator module receives readings from the Wii remote and sends them using Transmission Control Protocol/Internet Protocol (Tcp/Ip) to the appropriate device "Network distributer". On each device, there is an application for receiving the data and checking data appropriate receiving order.

The collected data pass into four levels of preprocessing for the gathered signal shown in Figure 2.5 to obtain better recognition of the hand gesture. The system first, calculates the magnitude of the gestures as sum squares of 3D vector. Second,applying moving average to smooth the curves over the signal. Third, apply a non-linear quantization on a scale of 10 values, so signal become more easy to recognize. Forth, If the gesture is too long / short then it must have the same size in order to compare it with stored gestures, so we applied interpolation extrapolation of signal. Finally, the signal will a have fixed width and be

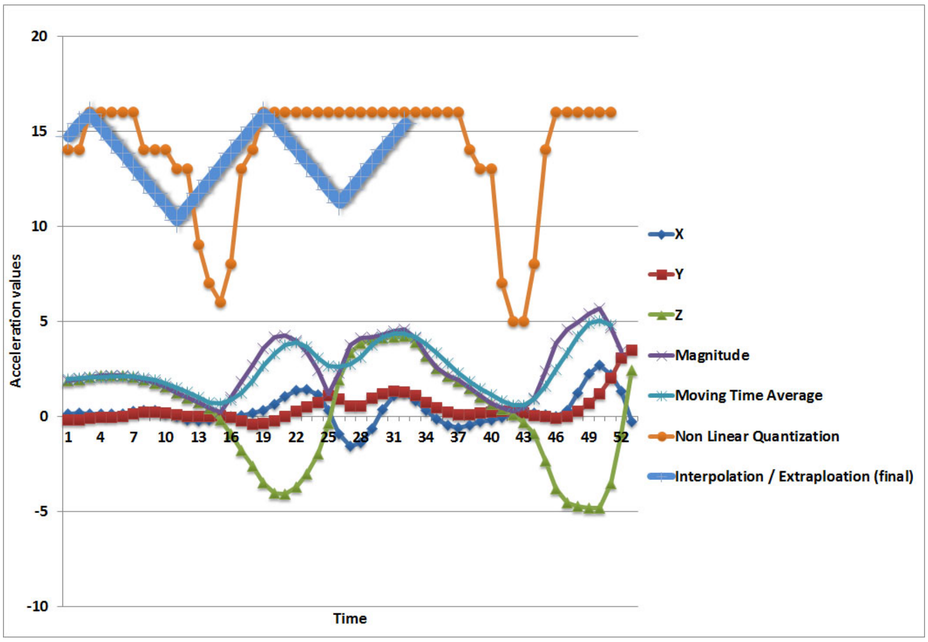


Figure 2.5: Analysis of a gesture that entails the shaking of the hand

ready for applying a recognition engine. The final signal can be shown in Figure 2.5 as the signal with bold line. The preprocessing steps was used in similar research of Mantyjarvi et al. [56] and Jang et al. [57].

The core of the recognition engine was built using a DP-Matching algorithm [58] [59] [60] for recognizing hand gestures. The cost function of the algorithm has been calculated as Euclidean distances between two 3D vectors, as shown in Equation 2.1, where $p_{x,y,z}$ is one of the accelerometer readings of a user's hand gesture and $q_{x,y,z}$ is the accelerometer readings inside stored hand gestures template.

$$Cost = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \tag{2.1}$$

It can be seen in algorithm 1 that the system calculates the minimum value between the users hand gesture and all stored template gestures. The recognized value should be less than some threshold value. The threshold value depends on the sensitivity of the application. An example for this value, when the user uses hand gesture to show some private information, in this case the threshold value must always be below 30 edit distance. On the other hand, applications for controlling media player could have less than 70 edit distance value.

Algorithm 1 *Recognize Hand Gesture h*

```
Procedure RecognizeGesture (Gesture h)
List=AllGestures.size
 for {k=0 to AllGestures.count}
   List[k]=EditDistance(AllGestures[k],h)
return min(List)
Procedure EditDistance (Gesture,h)
N = h.size
M = Gesture.size
for {i=1 to N}
  for {j=1 to M}
      D(i,j)= min (D(i-1,j)+1 ,
                   D(i,j-1)+1 ,
                   D(i-1,j-1)+Cost)
return D(i,j)
```

2.4.4 Customization interface

The adaptation of recognition engine with respect to users hand gestures in addition to prediction of the commands of the user depending on his heuristics have been studied by Xiang et al. [61]. However, the mapping of commands into functions have been included in the prototype system. Developers have to deep understand the system in order to get usage of their system.

We have designed prototype application for adjusting hand gestures and creating gesture profiles. The command map translates the gesture commands into application understandable events. In the current version of UbiGesture, a hand gesture is translated into keyboard-shortcut sequences. Each keyboard shortcut sequence is mapped to an interface or application function. The application developer has to list all application functions and features in the main system repository. The interface extracts the features and exposes them for user customization. The customization interface always compares the entered gesture with the stored gestures to avoid conflicts and alarming the user. The interaction interface compares the user's captured hand gesture with all the templates, returns the minimum distance, and compares it with the threshold value. If the minimum distance is below the threshold value, it will execute the command; otherwise, it will ignore the gesture.

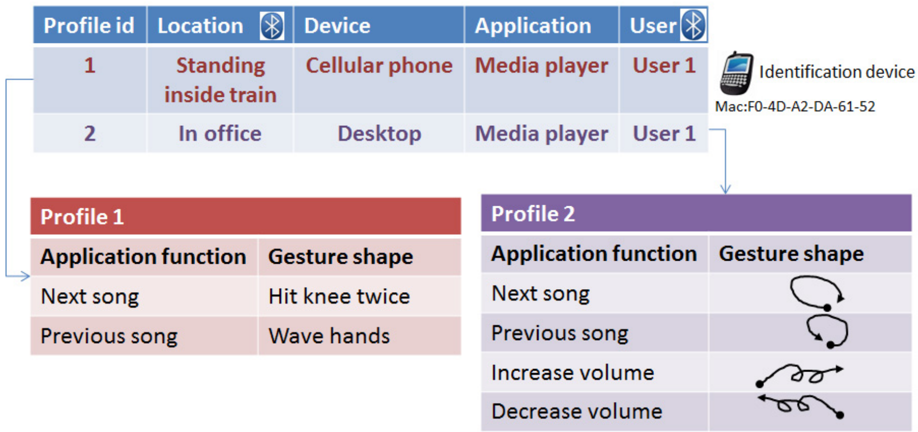


Figure 2.6: Gesture profile details.

2.4.5 Gesture profile

A gesture profile is recorded in the main repository of UbiGesture. This record is composed of the user, location, device, application, application functions and recorded hand gesture. The gesture profile is the key record in this research, and whenever the user changes his or her context, the appropriate gesture profile will be downloaded on the device of interaction. From this profile, the system extracts the rest of information such as the user's preferences and hand gestures. The profile is stored in the network for further access from different locations.

The details for the gesture profile can be shown in Figure 2.6. There exists 2 profiles for one user in different locations. The presence of users in locations is been tracked by Bluetooth. Each user has an identification device, which can be any device that have a Bluetooth connection and mac address. If "user 1" changed his location to "inside train", then profile 1 will be downloaded. If he changed his location to "in office", then profile 2 will be downloaded to him. The difference between profile 1 and 2 is that gestures inside train must be natural and accepted by people around the user. Such gestures like hitting knee twice, waving had wrist and tapping gestures are useful for this situations. In this situations, people do not have much space to make big gesture shapes. Moreover, there could be also difference in the number of application functions to control the application. In profile 2, people in the office might have more space to define more gestures to command their applications.

2.5 Evaluation experiments

We want to study if users like to adjust predefined gesture profiles. Then we want to study if user situation changed, does user wants to change gestures. In addition, we studied the effect of imposing position and social parameters on users interacting with predefined hand gestures.

A test bed was established using two positions in one location in standing and sitting positions. The room has a large display screen and a three-seater sofa. Subjects used their cellular phones as an identification device. The subjects interact with windows media player that has 9 functions (Play, Pause, Mute, Stop, Close, Increase volume, Decrease volume, Next, previous). Figure 2.7 shows the media player 9 functions and the predefined gestures. Predefined hand gestures for those functions were set to cover basic shapes (Play, pause, next, previous) and complex shapes (Mute, stop, close, increase and decrease the volume). These different shapes enable users to imagine what hand gesture they can adjust using their hands.

The basic hand gesture shapes were selected to comply with the study done by Kela et al. [45] for controlling VCR player. The complex gestures were selected to have a symmetric shapes like the increase and decrease volume. Other gestures have direct map to English letters like the mute mapped to letter 'M' and stop was mapper to letter 'S'. The close gesture was mapped to the normal wrong mark. We think that this gesture shapes are common to use and easy to remember by the majority of people.

The subjects subscribed in the experiments were 20 volunteers. All volunteers were between the ages of 22 and 35 with an average of 25, and all were computer science specialists. Subjects were 18 males and 2 females and one user was left handed user. The subjects were divided into 2 groups for doing the experiments, first group did the experiment in a standing position then in a sitting position. The second group, did the experiment from a sitting position then standing position. We need to know if there was a learning effect difference between the sequence of doing the gestures. At the end, of the experiments we conducted an interview with the users to get some feedback on the way they configure their gestures. In addition, how they think about the parameters affecting their gesture shapes.

Function	Gesture	Function	Gesture
1-Play		6-Increase volume	
2-Pause		7-Decrease volume	
3-Mute		8-Next song	
4-Stop		9-Previous song	
5-Close			

Figure 2.7: Media player 9 functions and predefined gestures.

2.5.1 Setup

The administrator granted permission for volunteers to access the location, access the big display screen in the room, and run the media player on the device. The developer made all the applications accessible by keyboard shortcuts. The Administrator stored the application functions and their corresponding keyboard shortcuts in the main repository. The volunteers were asked to perform four set of positions (standing, adjusted standing, sitting, adjusted sitting). In the standing and sitting positions, the subjects used a predefined profile. The subjects have to enter the nine gestures in each position four times, the first session is used for training only. The subjects were asked to evaluate the satisfaction of the predefined gestures using Likert scale of 5 values. Then, subjects were asked to adjust the gestures they hate or not satisfied with. After that, they used the adjusted profile in standing and sitting positions, then we measured the accuracy of interaction. During the experiments,

(a) (b)

Figure 2.8: User positions (a) sitting position, (b) standing position.

verbal guides were given to beginner subjects when they asked to adjust their gestures. There was a three seconds pause between each two consecutive gestures. This pause time will allow users to return their hands back to the starting position of doing gestures. If the subject enters wrong gesture shape, it is considered as an error and requested to repeat doing the gesture. We put a border between each sofa seat to simulate the presence of another person. Figure 2.8 shows users in standing and sitting positions. We illustrated to the volunteers that they should not touch this border during the experiments and we counted the number of times they touch the sofa border.

2.5.2 Results

In order to know the learning effect between the sequence of doing the experiments between standing and sitting positions, We conducted a one way ANOVA to test for the difference among the number of modified gestures. There was no significant difference between the two groups with $F(1, 16) = 0.43, P > 0.5, F_{crit} = 4.5$. The results show us that people like to change the gestures they are not satisfied with. Figure 2.9 shows that the more the users are not satisfied by the gestures the more they like to change gesture shapes.

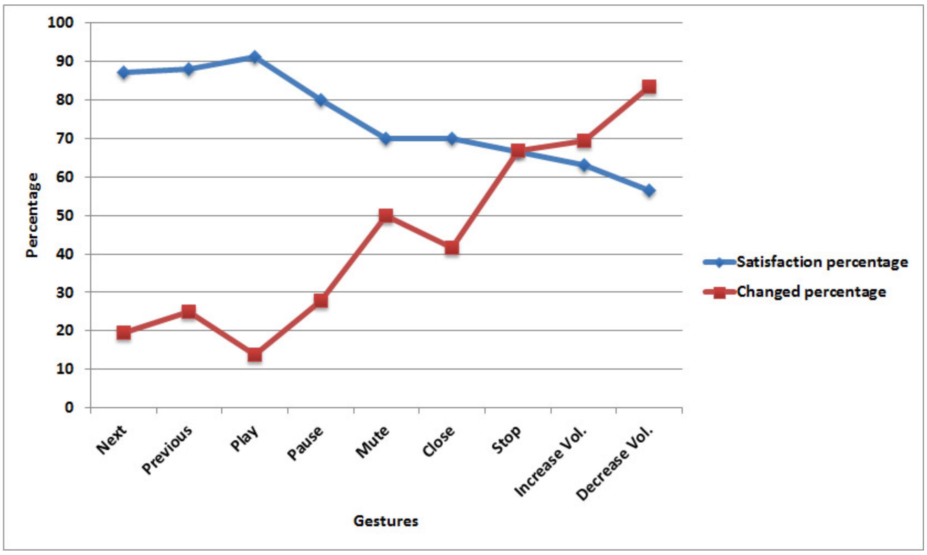


Figure 2.9: The more the users are not satisfied with gesture shape the more they adjust.

There was a strong negative correlation coefficient between the satisfaction and adjusting rate of -0.96.

We found that the more the shape of the gesture became complex in shape like the mute, close and stop gestures the more people like to change them. Also, increase and decrease volume gestures were originally done in big shapes and with fast speed which was difficult to achieve their similar patterns by users.

The results show that adjusting gestures help people to interact more accurate with the media player application. Figure 2.10 shows the average number of error difference between the four positions. Standing adjusted gestures had the minimum number of errors as it was easy for the users to perform their gestures using all spatial space around them.

In sitting position people changed basic stroke gestures more than in standing position. A one way ANOVA to test for the difference among the number of gestures modified for standing and sitting. There was a significant difference between standing and sitting, $F(1,6) = 28.14, P < 0.5, F_{crit} = 5.9$. Figure 2.11 shows that users prefer to change the basic stroke gestures in sitting positions, because they hit the sofa border many times or it was difficult for them to move their hand to down directions.

When the users adjusted the complex shape gestures in sitting position the accuracy of interaction increased. A one way ANOVA to test for the difference among the average number of errors for sitting and sitting adjusted positions. There was a significant difference

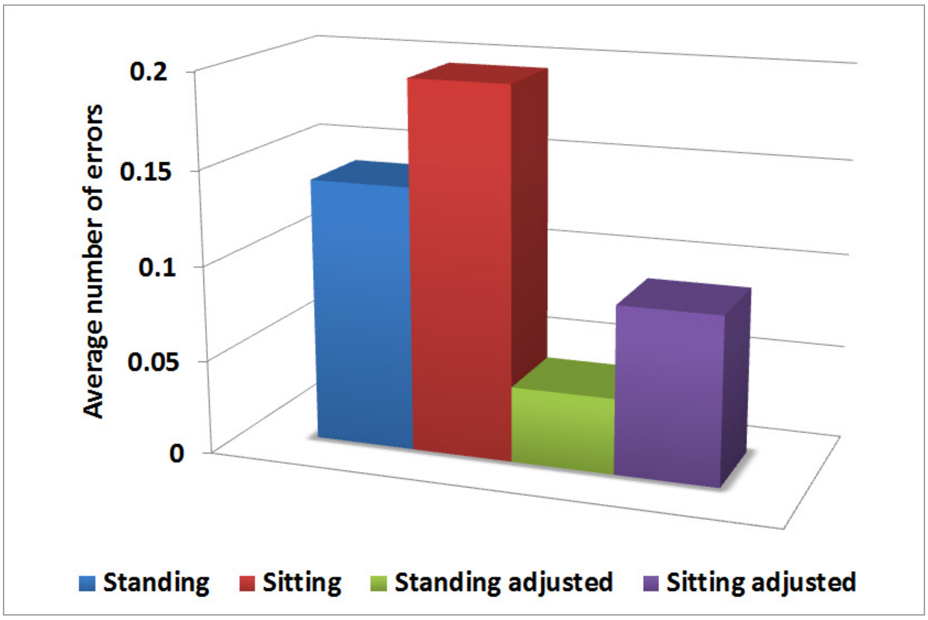


Figure 2.10: The average number of error difference between the four positions.

between sitting and sitting adjusted, $F(1,8) = 11.52, P < 0.5, F_{crit} = 5.3$. Figure 2.12 shows that when subjects adjusted their gestures in sitting position the number of errors decreased.

2.6 Discussions

It was found that most of the users at least adjusted two gestures in standing or sitting positions. After finishing the experiments, we interviewed the users to know how they think about the relation between context and gesture shapes. We asked them to fill in forms with Likert scale for how they think about parameters affecting the gesture shapes. Table 2.1 shows the parameters we ask about and the average values for the answerers we collected.

First, we asked the subjects to evaluate the relation between the gesture shapes and locations. We found that 90% of subjects agree that gestures and locations are highly related to each other. Second, 96% of the subjects agree that downloading adjusted profiles to them according to their situation has a lot of potentials to enhance their interaction with ubiquitous environments. Third, 83% of the subjects think that standing and sitting position could change their gesture shapes because the difference in spatial space or the directions of the gestures that could not be achieved by sitting positions. Finally, 90% of

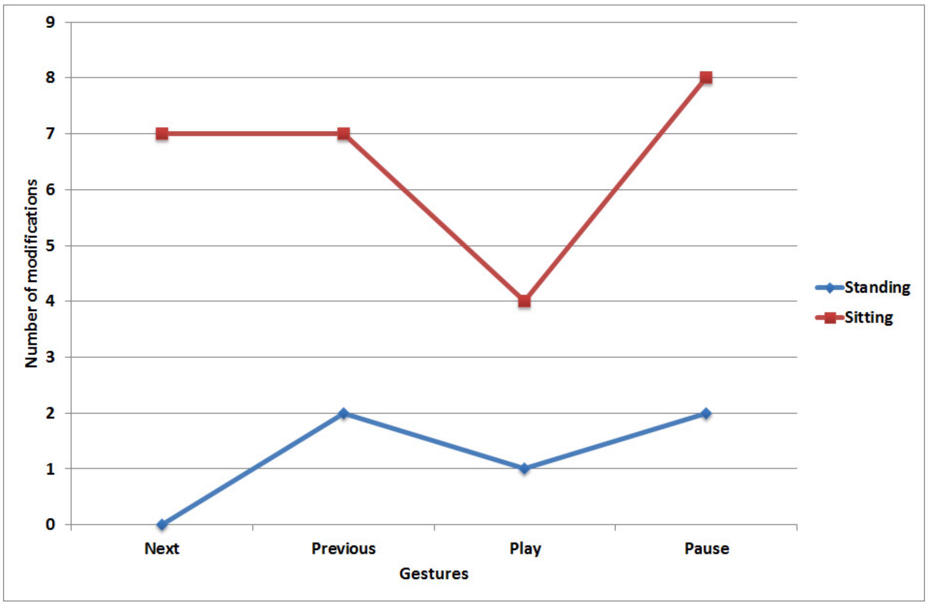


Figure 2.11: Subjects modified the basic gestures more in sitting position.

Table 2.1: Users feedback

Parameter	Average
Do you agree that gesture shapes is related to location	90%
Do you like to adjust profile and download to your location	96%
How much is the importance of standing/sitting position	83%
How much is the importance of being in transportation	90%
How many times subject touch sofa border	6

the subjects agree that a situation where it is in public transportation or busy in activity could highly change the gesture shapes. We also observed that users touched the sofa border on an average of six times in the sitting position. Subjects comment that if the person sitting beside them is person they know like freinds, then gestures could be fun, otherwise it will be disturbing action.

The subjects tried to avoid touching the sofa border nearest to them, so they failed in completing the gesture. When the subjects adjusted these gestures, the error rate and time reduced, and they were satisfied with their gestures. Subjects pointed out that there could be other factors that can affect the gesture shapes in locations like being in relaxed mode or serious mode, more position parameters like sleeping laying, health of user, activity, time of doing gestures and age of users. We think that location, device, application and user context

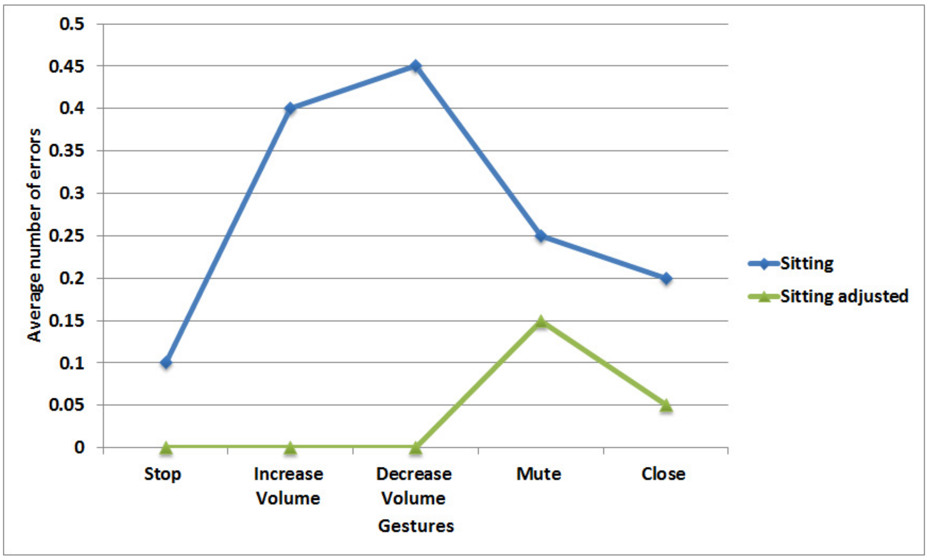


Figure 2.12: Sitting adjusted position reduced the number of errors.

parameters can affect the gesture shapes of the user as they represent user situation. The customization interface was easy for the subjects to use, as it used simple menu commands.

2.7 Summary

We developed a system called UbiGesture system, which profile user gestures according to context parameters. The system will download the gesture profiles for the users according to their context parameters. People like to customize their hand gestures according to their situation. We conducted experiments regrading the user behavior regarding the change in the situation in means of position of doing gestures standing and sitting positions. Furthermore, the social implication of gesture and its effect on the accuracy of interaction was studies. The results show us that the context parameters play an important role in changing the gesture shape of the users. The results also show us the when users adjusted their profile, they can interact more precisely and naturally.

Chapter 3

Smart gesture sticker: Everyday objects for interaction

3.1 Introduction

Ubiquitous environments are now very popular interaction environments as it was envisioned by Weiser [1]. The emerge development of devices, and wireless sensors have directly affected the techniques for interaction in such environments. Interaction in the ubiquitous environment can suffer from the nonexistence of conventional input device like keyboard and mouse. Moreover, sometimes people are not hands free to interact with both their hands. People could be involved in another activity, or they are holding some objects in their hands. Such objects they carry are candidates for several interaction possibilities [62].

Interaction with everyday objects is one of the challenging research areas in smart ubiquitous environments. Recently, smart applications that depend on user's activity recognition directly use everyday objects. Objects have a direct impact on the interaction in smart environments according to their shapes [63]. There are some gestures that done by users naturally while they hold objects unintentionally. Object gestures are defined as moving an object in the spatial space. In several cases, using hands to interact with devices is not applicable such as to check email while driving a car. The user is involved in another activity and his/her hands might not be free for interaction. It is not appropriate for a user to perform a big circle using object in hands in public areas. It will seem to others as someone is threatening them. Gesture by itself is a production of users imaginations, so

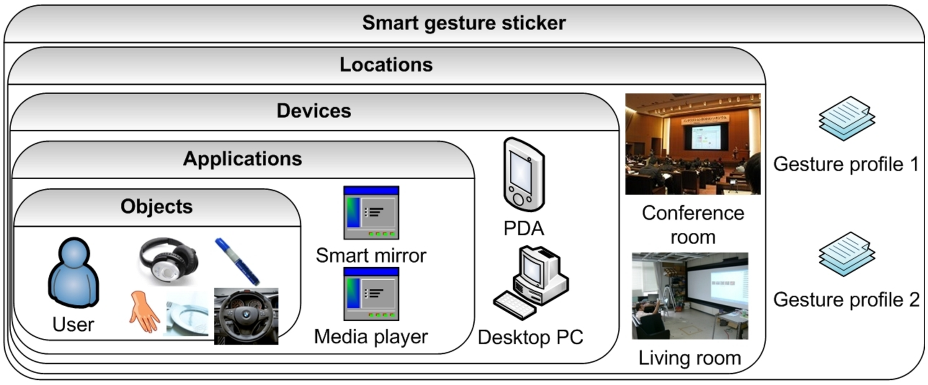


Figure 3.1: SGS main context parameters.

there are almost no standard gestures that could match the satisfaction of many users. In this research, target users those who are interacting with various objects to perform actions in different situations. We consider different factors for situations where the user can exist in ubiquitous environments.

A situation is defined as a combination of user's context parameters. The main context parameters can be shown in Figure 3.1. A gesture profile will combine location of the user like being in conference room or living room. Device the user interacting with like his PDA or desktop PC. The application the user is controlling like smart mirror, media player, or any other application running on the device. The profile will include the object the user is interacting with like headphones, car wheel drive, pen and dish. Finally, the user himself is the corner stone of gesture profiles including his preferences and experience level.

Sensors have been used to capture motions [64], physical body activities [65], emotions and capturing human life logs [66] [67]. The 3D accelerometer sensors as a motion capture sensor were used to capture motions and activity in [68]. In typical scenarios for object interaction in ubiquitous environments, users have to build predefined scenarios using various sensors and embed them in every object they define for interaction scenarios. We think that such approach requires the use of many sensors in every object around. It is sometimes difficult to achieve due to the huge number of everyday object people interacting with daily.

We propose a technique to use any object around users for devices and appliances interactions. The technique makes user interactions more natural and intuitive. We developed a system called "Smart Gesture Sticker" (SGS) to facilitate user interaction with everyday objects in ubiquitous environment. The system is designed to allow users to customize ges-

Figure 3.2: 3D accelerometer sensor.

tures for objects according to their context parameters. SGS packs the user's gestures and provides a customized interface in his/her environment. SGS recommends gesture profiles to people according tho their existing context parameters.

Gestures of users can be captured with many techniques such as using a camera [69]. The use of camera issue the privacy concerns from users about being captured. Consequently, we use a small attachable coin size wireless 3D accelerometer sensor [44]. Figure 3.2 shows the 3D accelerometer sensor used in this study. It is a coin size small sensor that can be attached easily to various objects. It has a built-in 3D accelerometer, angular rotation rate sensor and temperature sensor. The sensor is a 20 g weight with dimensions of 39mm (w) x 44mm (h) X 12mm (d) and acceleration range is ± 2 G. The sensor has been used to capture activity of the elderly person in the work done by Pouke et al. [70].

3.2 Related work

Hyper-objects was discussed from designing point of views and their effect on lifestyle patterns in the work by Mavrommati et al. [71]. They discussed the potential benefit of hyper-objects on the daily life of people. Kameas et al. [72] design an architecture that aims to provide a conceptual and technological framework for engaging and assisting ordinary people in configuring everyday objects, which are able to communicate, using wireless networks. However, the tangible objects were limited to activity recognition of users. They did not provide a way to profile the activities per the users and the object locations was fixed. Kawsar et al. [73] used objects augmented with sensors to provide value added services in the context aware environment. They show three types of user applications and how can objects be used to get context of the user. Chi et al. [74] classified the design choices of an object human interaction by surveying different object interaction research papers and projects. They show that the digital enhancement to object's functions must avoid any conflicts between its traditional function. In our research, we try to allow users to use any object around in different ubiquitous contexts with the ability to customize their gestures.

Hand gestures and object gesture are similar in their usage as both are a movement in the spatial space. However, some object gestures have some limitation for their movements as they have less space such as objects attached to heads, neck and legs. Some users prefer hand tangible devices like the smart glove [75] or using a handled device like the magic wand device [76] to control devices around them. We try to generalize the case of using objects and allow users to use any object to control devices in their context.

Context tracking and location aware services that can support the user in his/her daily activities had a lot of techniques and ideas that were discussed in [77] [78]. Situation dependent user profiles in context aware environments have been discussed in the literature from different aspects and points of view. Sutterer et al. [79] present arguments for structuring the user profile into situation-dependent sub-profiles. They extend their work in [80] to focus on the problem of profiling from the user point of view and his/her context by adding ontology reasoning to select the appropriate user profile.

Cheng at al. [81] used special tags for identifying objects and binding them to predefine limited gestures. Their study showed the importance of object interaction to minimize the time needed while interacting with multi-task operations. However, their system depends on special hardware setup based on camera, and the bounded applications are limited in functions and mapping operations. In this research, we used the shake and tape gestures to start and stop using object for interactions as mode gestures, and we used sensor threshold values for modeless interaction. Hendrik et al. [82] used the idle time of not interacting to capture user activity. However, they point out that implicit change from interaction to activity recognition is still need development and research. The identification of objects was done by Merrill [83] through the usage of infrared LED to identify products in shop or parts of a car engine.

3.3 Object gestures

In order to develop a system capable of capturing gestures of users in ubiquitous environments, we have conducted a detailed study about object gestures. We want to study how the object gestures can affect the interaction with the environment. SGS should support users in different situations by providing object gestures for the users anywhere. The use of different objects could inspires users for more intuitive interaction with their environment.

In this study, we choose objects with different shapes including the human body parts so we can study the most intuitive and suitable objects that can make users interact with applications fast and accurate. We asked five subjects to do this experiment, all subjects are aged from 25 to 33. At the beginning, of the experiments, we show the users different 10 objects and they choose three to use during the experiments. Objects are (user's hands, user's legs, stylus board pen, bottle, book, cellular phone, tooth brush, umbrella, handy fan and wireless headphones).

A total of three scenarios to be done, two scenarios to be done in the living room while watching movies using media player and using two different objects. The other scenario is done in the bath room while interaction with smart mirror to control media player. The media player has nine functions that can be customized (Play, Pause, Next, Previous, Increase volume, Decrease volume, Mute, Stop and Close). In the first two scenarios, the subjects switch between the two objects and determine the preferred scenario to play. In the third scenario, one profile will be downloaded automatically in their context.

First, we calculated the number of conflict gestures done by users while they customize their gestures. Second we measured the accuracy of interaction with predefined gestures and the three customized gestures scenarios. Third we get some feedback from users about appropriate usage of objects for gestures. If the users enter a gesture that is not adequate, it will be counted as an error.

The results show that the majority of users 76% create there hand gesture without having any conflicts, and 16% had at least one conflict gesture. The average time to create object gesture was nine seconds. Some subjects show more consumed time up to 86 seconds because they perform similar pattern gesture shapes that lead to conflicts.

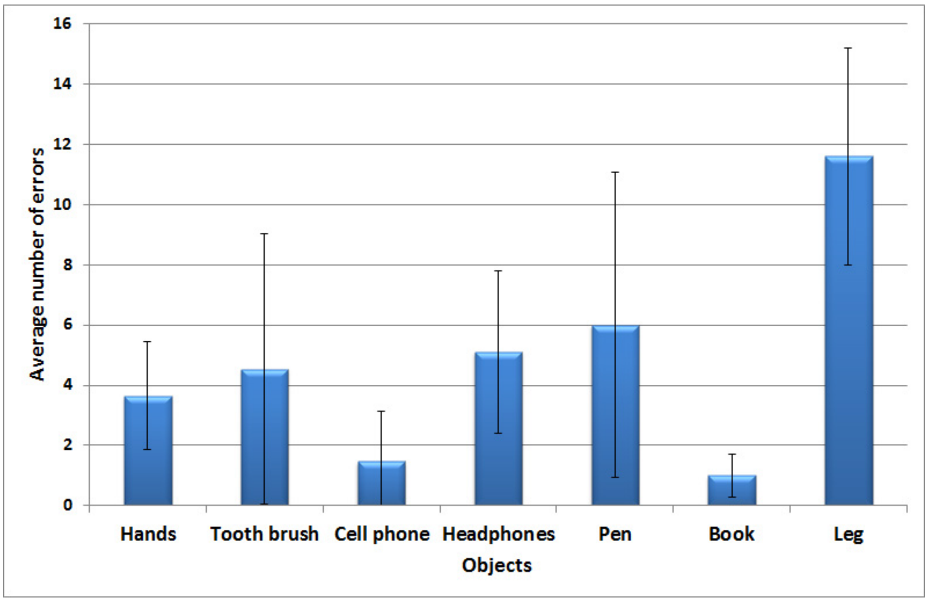


Figure 3.3: Object gesture results

The results show that when users get familiar with the system, they can generate gesture ideas such as using headphones to play and pause media player. They choose the gesture of taking off the headphone to pause the media player and putting up headphone to play action. They can create a smart headphone appliance through recognizing their current activities and command tasks towards those activities. Figure 3.3 shows that the there was a difference between the objects used in the experiments. A one way ANOVA to test for the difference among the objects. There was a significant difference between objects, $F(1, 28) = 4.3, P < 0.01, F_{crit} = 2.44$. Some objects like cellphones and the book were easy to control compared to legs which has a problem of space to make gestures. If users select some appropriate objects, they could have a less errors for the application they are interacting with. When people use some objects like the book and cellular phone they make natural gestures like opening book, moving cellular phone from left to right hand. This type of gestures are easy to perform and natural for use.

Comments and verbal feedback from users about SGS system was also recorded. One user tried to use his leg for interaction and defining all the functions. Since the gestures that can be done using the leg is limited, as the space for gesturing by leg is small. It was hard for the user to customize all the functions for the media player using his leg. Another user says that hitting and tapping gestures was easy to memorize. Subjects recommended that the strength of doing the gesture with the object can affect the level of doing a command such as increasing or decreasing volume. Users says that gestures need some time to be learned. However, after learning, the usage of gestures was very similar to real-life situations.

Customizing gestures were intuitive to use specially the scenario of using the cellular phone was easy to understand. We noticed that some users could make gestures like throwing in the air, which might have some social threaten if used in public space area. Some users use a headphone object on his/her head, trying to do similar gestures pattern. They fail as there is a limited space to do gestures with headphones fixed on head. Furthermore, SGS depend on threshold values to capture the gestures, so weak gestures are difficult to be captured.

We notice four main design principles for object gesture customization and interaction interface based on our study and users feedback. First, the gestures should be fitting all situations of people implicitly. People can change their location, device, application or object they are interacting with. The ubiquitous environment should maintain well understanding of the gestures. Second, is the intuitiveness and natural use of the gestures. People should interact with devices using gestures as they interact with everyday objects. Third, people do not prefer to consume much time to learn and get trained for the gestures. This can cause frustration and make them bored from using the system. Hence, there is a need for an easy way to customize interface that can depend on the minimal number of learning gestures and can support the ability to learn the gestures easily. Finally, the adjustment of a predefined profile are much easier for a beginner user than creating a profile from scratch.

3.4 Gesture stickers

Stickers are a way to add attributes to some object to enrich it's shape or value. Some stickers could have a universal meaning or they are empty. Empty stickers can have anything written on it giving a new meaning for the object it is attached too. There exist two types of stickers one is universal or have a predefined meaning that every one can understand. The other type is adjustable sticker or empty stickers which can be mapped to predefined profiles and adjustable profiles. Figure 3.4 shows two different types of stickers. We want to study if we can attach gesture profiles to objects so that they can give new meaning and functions to the object attached. We assume that people will attach some sensor to an

(a) (b)

Figure 3.4: Predefined fragile symbol sticker (a) and adjustable or empty stickers (b)

object they have in their environment, and the system embedded in their environment will load appropriate interaction gesture profile for that object.

3.5 Proposed SGS overview

SGS system is functioned by tracking users basic context parameters: location, devices, applications, objects and users in different ubiquitous environments based on UbiGesture infrastructure [84]. A subscribed user to the system enters some location and selects one of the devices he/she wants to interact with. User sticks the sensor to one of objects by binding it with double side tape or bounding piece of cloth. SGS shows an interface on the device that allows the user to choose the applications he is permitted to run on this device. After the user selects the application, SGS will loads the appropriate gesture profile for user in this ubiquitous environment. Users can create new profiles or they can adjust gestures they performed in a similar ubiquitous context.

Figure 3.5 shows SGS profiles parameters and SGS role. The profile of the user includes the device he is interacting with like big display screen or PDA and so on. Furthermore, the application he/she pretended to control like media players, TV controllers. The object he/she is using to control device is captured such as stick, headphones, or any object around including the human body parts like hands, legs and head. In addition to this, SGS will recommend the user object gesture adjustable profile based on his context and objects used. A typical gesture profile composed of profile ID, location name, device, application, object

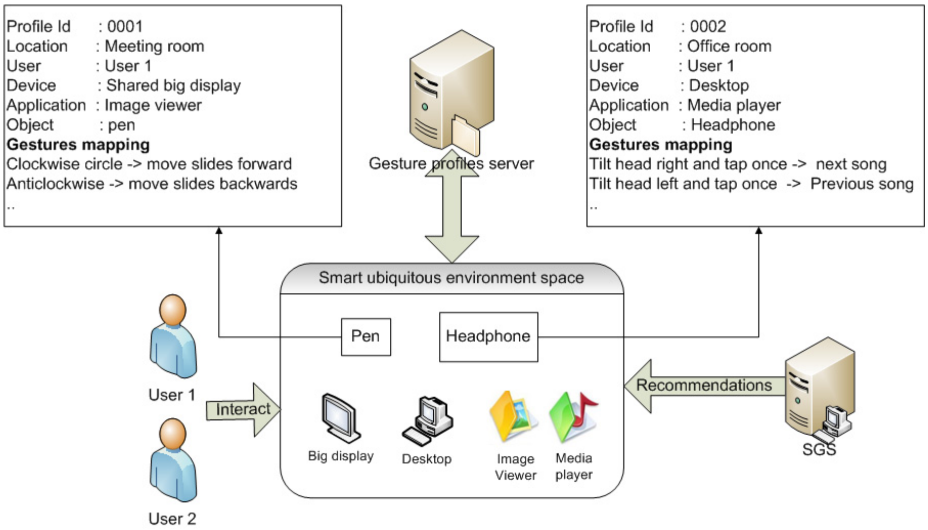


Figure 3.5: SGS profiles

and gestures attached to this object. Later, when people change their location, SGS will search it's repository to get them the most appropriate profile for their context.

3.5.1 Gesture recognition

The WAA006 sensor sends data as a pattern of signals that represent the movement of the sensor. The output of the accelerometer is a strip sequence of 3D points denoted by G, such that a point can be represented as $h_t\{a_x, a_y, a_z\}$. h_t is a time stamp generated automatically by the sensor and a_x, a_y, a_y are the readings recorded from the accelerometer at time t.

We used k-mean clustering algorithm with three clusters to compare the distance between the stored object gestures. Then, gesture is classified into one of the clusters. SGS applies DP-matching algorithm to get similar gestures. The cost function for DP-Matching has been calculated as the Euclidean distance between the two 3D vectors. SGS system calculates the minimum value between the user's object gesture and all stored template gestures within the cluster. Each object has a normal functionality rather than being used for interaction. The user has to decide when he wants to use object for interaction and when he wants to use it naturally. In this research, we asked the users to make four successive shake gestures to start using the object for interactions and once again to stop interacting.

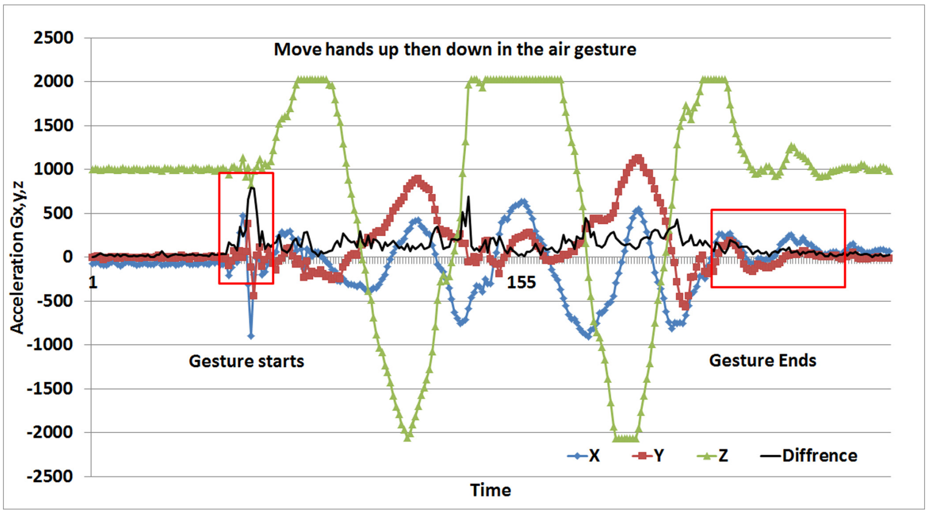


Figure 3.6: Capturing gestures using start and end thresholds for moving hand up and down in the air.

3.5.2 Capturing object gestures

There are various ways to capture the gestures for interaction using objects either. Some techniques depend on using explicit buttons in Jiayang et al. work [85] and work done by Daqing et al. [86]. Other methods can also depend on using direct methods to switch between object modes like using the camera in wong et al. [27], RFID tags in work done by Inomata et al. [87] and special tag readers in work of Ljungstrand et al. [88] and Tammara et al. [89].

The system will track all the gestures done by the user using the objects. SGS gestures with the stored template of gestures, until it finds a matching gesture with a similar pattern shape same as the work presented by Rehm et al. [90] and Kela et al. [45]. In order to achieve flawless capturing of gestures, the proposed system is monitoring all the accelerometer values over the three accelerometer axis x, y and z. Continuous recognition of users gestures with respect to their location has been studied by Daehwan et al. [91]. They used limited gestures to control turning on / off of light and curtains.

The start of gesture is captured by finding the Euclidean distance between two successive accelerometer 3D vector points. If the distance, between those two points exceeds a predefined threshold value of 120, and it stands for 0.4 seconds then it has a possibility to be a gesture. Hence, the system will start recording all points respectively. The minimum number of points to capture for a candidate gesture must be at least 50 points. If the

Euclidean distance, between two successive points became less than 40 and stands for 0.8 seconds then this means the gesture has been ended.

Figure 3.6 shows capturing gestures using a mode less technique for moving hand up and down in the air. The difference between two successive points of a gesture sequence can be shown as the difference curve on Figure 3.6. When the difference between the two points stands above a threshold value for a period of time, it means that this is a start of gesture to be captured. If the difference keep below a threshold value for an amount of time, this means that the gesture has been stopped. The left box represents the start of gesture and the right box represent the end of gesture.

3.5.3 Selecting appropriate profile

SGS keeps tracks of a shared network log repository that can be accessed in ubiquitous environment. SGS load users context parameters through this log. Any user may create a profile with respect to his/her context. If the user wants to create a profile, then SGS lists all objects existing in the user location and available applications to be associated with objects. SGS searches for the most appropriate profile for the application according to the attributes of the context parameter defining it. An example of objects existing in the same location that might have some common attributes is on the desk or in the office room. The pen and the pencil might be similar in their usage. The initial primary attribute values for those context parameters were defined based on observations from the primary study on object gestures. Table 3.1 shows the basic context parameters attributes.

Objects can have some parameters that can affect the gesture shapes like the size, length, weight and geometric shape of the object. Object size can affect the handling of the gesture either by both hands or one hand or even using fingers only. Applications parameters like media players, browsers, text editors cab have affect on the gesture shapes. Basic functions for browsing images, presentation slides, Internet browser tags can all have remarkably similar gestures. Moreover, device parameters like portable, screen size can directly affect the gesture shapes. Devices that large projected screen could have similar gestures cause the criteria of interaction and reaching items on the screen are the same. Location parameters like being in public, transportation and indoor locations can have a social influence on the gesture shapes. A gesture that is done in public space must be different from the one done in an indoor location as in booth situations social awareness is different. Finally, the user

Table 3.1: Context parameters sample

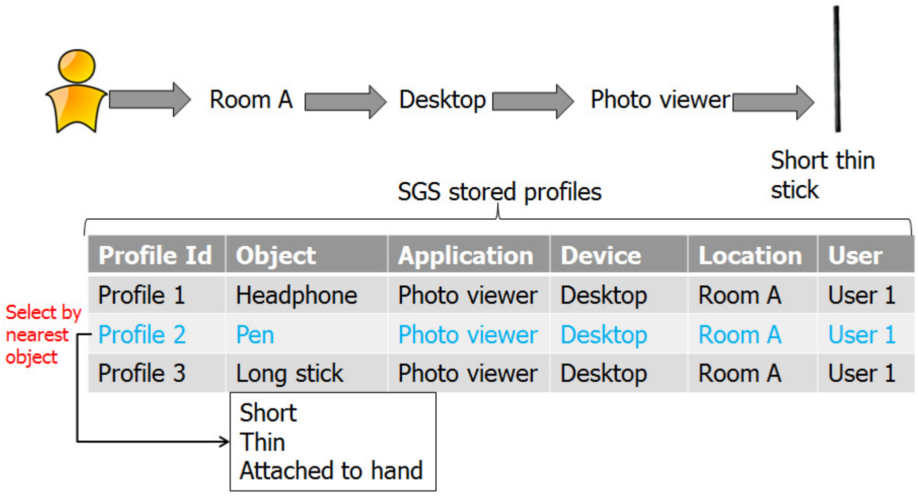
Parameter	**Sample**
Objects	Long, Short, Big, Small, Fixed on hand, Fixed on head, Sphere shape, Stick shape and Cone shape.
Application	Media players, Presentation viewers, Browsers,Text editors.
Device	Portable, Fixed, Small display, Big display, Large projected display, Speakers exist and Speakers not exist.
Location	Office, Car, Indoor general, Outdoor general, Restaurant, Mall and Train station.
User	Beginner, Moderate, Expert (to use gestures), Computer science specialist, Other specialty.

himself could have parameters like his experience level or using gestures in general, his field of specialty or education background and being left or right handed. Users gender also could affect the hand gesture shapes, as females intended to do gestures in weak and small size patterns. Also, the more the users have experience with doing gesture in playing games and so on, the more they like to have innovative gestures.

SGS will match the profiles with maximum similar attributes for context parameters by giving priorities for objects, device, location and user parameters respectively. We give users parameter lowest order for selecting profile to preserve the privacy of users. However, each object gestures have default gestures for applications to be selected at the initialization of the system.

Those attributes are changeable and each profile item can be classified by many attributes. A gesture profile example classified as (user is expert, object attached to hands, location is in public space, device is portable PDA and application is media player). An example of selecting profiles, if a user enters the room A and want to interact with desktop PC, and run photo viewer application using a short, thin stick as shown in Figure 3.7. The system has three stored gesture profiles for headphone, pen and long stick. The rest of the parameters are all the same. In this case, SGS will compare the objects by their parameters and find the nearest object parameter. In this case, SGS will load the gesture profile of pen because the pen is short and thin, so it is highly similar to the stick.

In the second example, SGS has three stored profiles for the same user in the same location with the same device, but having different applications and objects. If a user wants



Profile Id	Object	Application	Device	Location	User
Profile 1	Headphone	Photo viewer	Desktop	Room A	User 1
Profile 2	Pen	Photo viewer	Desktop	Room A	User 1
Profile 3	Long stick	Photo viewer	Desktop	Room A	User 1

Figure 3.7: Selecting profile using the nearest object.

to create a new profile for the short thin stick, then SGS will ignore using profile 1 shown in Figure 3.8. Profile 1 has a different object with many different attributes (headphone) and different application (media player). Profile 2 has a long stick which is thin and long and profile 3 has a board pen which is short and thick. SGS will find that both objects have equal attributes, then it will search for similar application. It will find that profile 3 used photo viewer which is similar to profile the user wants to create. Hence, it will select profile 3 because it has the nearest application.

3.6 Applications

SGS users create their own smart appliances by adjusting object gestures for context. We tested three main applications to be run by SGS. The first application is media player application. We used Microsoft windows media player. Second, is the photo viewer application that we developed to browse and apply effect on pictures. It has 15 functions (Start PTZ camera, Show image, Close application, Zoom out all pics, Zoom in all pics, Send picture to my home, Navigate right, Navigate left, Navigate down, Navigate up, Save captured image and close, Black white, Crop image, Take picture, Rotate image). Third, is PTZ camera protection application that allows the user to control the camera directions and make some reactions in case a person felt being watched. The PTZ camera protector application has nine functions (Tilt up, Pan right, Tilt down, Zoom out, Pan left, Redirect

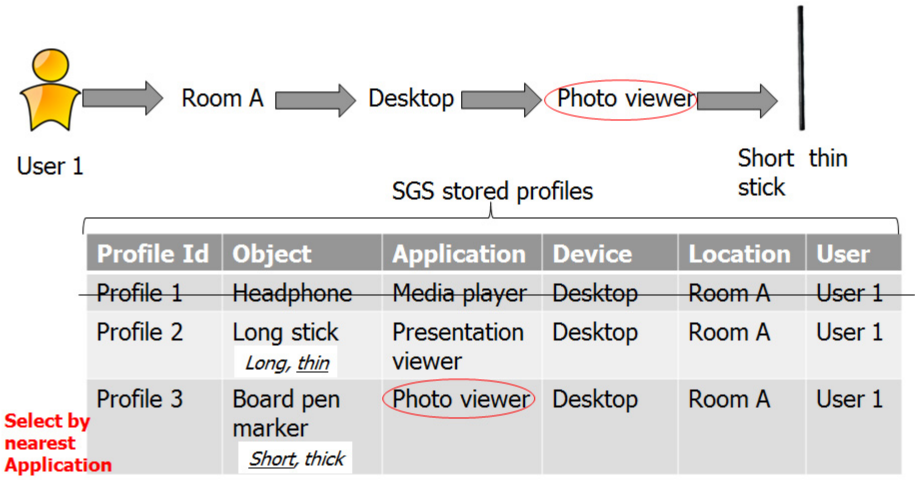


Figure 3.8: Selecting profile using the nearest application.

to board, Zoom in, Close, Redirect to desktop). We ask users to interact with the SGS and ask them to create their own gesture imaginations. Hence, we have classified the interaction with objects and applications into two smart scenarios.

3.6.1 Media browsers

Smart headphone: The user created intuitive gestures for controlling their appliance. Putting headphones on the head mapped to play music file. Putting off the headphone on the desk pause the music file. Tilting head to the right and tapping headphone increase volume. Figure 3.9 (a) shows a user using headphone and tilt and tap gesture to control media player.

Smart mobile phone: The user created in the train station. The user maps the gesture of play and pause media files as moving his mobile phone from the right hand to the left hand and vice versa. The user assigns the stop command as to put his mobile phone in his pocket. Figure 3.9 (b) shows a user using a mobile phone in simulated train station seat.

Smart hands: The user maps gestures to control a photo viewer application. The user mapped the rotate image command as a capital 'R' shape gesture, and the black and white filter into 'B' shape gestures. Figure 3.9 (c) shows a user interacting with his hands.

Figure 3.9: (a) Smart headphone (b) Smart phone (c) Smart hands (d) Smart book

3.6.2 Device controller

Smart book: The user used a book on the table to control PTZ camera protector application. The user maps the waving gesture as a blocking command for PTZ camera and is redirecting it to a white-board area with some message. Circle clockwise and anticlockwise to zoom in and out respectively. Figure 3.9 (d) shows a sensor attached to book to control camera protector application.

3.7 Evaluation experiments

3.7.1 Experiment 1

We conducted an experiment to compare between the SGS selected profiles and the adjusted profiles by users in means of speed and accuracy. In addition, we studied which parameters affecting the object gesture profiles and its effect on the accuracy of interaction. We asked six subjects to do the experiment. All subjects aged from 25 to 32. We configure three different situations. The first scenario is in the living room while watching photo picture viewer application. The second scenario is in the office room while sitting on the desk and using a PTZ camera protection application. The third scenario is in the outdoor location like a train station and using media player.

In this experiment, each user creates a new profile using the customization interface. The user first selects one of the objects to use for his/her scenario, and then the user asked to select an application to run. SGS will copy the appropriate profile for the user. The user uses the SGS profile to interact with the application all functions one time. The user evaluates the SGS profile without any adjustment for gestures twice. Then we ask the users to adjust some of the gestures that they did not like then evaluate the profile again three times. In the experiment, each time a message appears to the user asking him to enter a gesture and is showing a recorded video of that gesture. If the user enters different gesture pattern, then it is counted an error. Time measured to complete the whole session and enter all the functions of the application.

3.7.2 Results

Initially users take time to learn about the SGS created profile gestures specially for the first scenario (photo viewer application). Figure 3.10 shows the comparison between the three scenarios in means of speed and accuracy. After subjects had adjusted their gestures in scenario 1 (Photo viewer), number of errors reduced relatively. It was observed that when the number of functions for application increased as the case of the photo viewer application, the number of errors was much higher than scenario 2 and 3. In scenario 2 and 3, we observed that the selected SGS profile speed Figure 3.10 (a) and accuracy Figure 3.10 (b) remains constant even after user adjusted some of gesture shapes. A one way ANOVA to test for the difference among the average number of errors between SGS profiles and adjusted profiles for scenario 1 (photo viewer) application. There was a significant difference between SGS and adjusted profiles, $F(1,28) = 6.22, P < 0.05, F_{crit} = 4.19$. A one way ANOVA to test for the difference among the average number of errors between SGS profiles and adjusted profiles for scenario 2 (camera protector) application. There was no significant difference between SGS and adjusted profiles, $F(1,16) = 1.8, P > 0.05, F_{crit} = 4.49$. A one way ANOVA to test for the difference among the average number of errors between SGS profiles and adjusted profiles for scenario 3 (media player) application. There was no significant difference between SGS and adjusted profiles, $F(1,16) = 1.9, P > 0.05, F_{crit} = 4.49$. Thus, we think that SGS can support users with appropriate gesture profiles within ten gesture shapes. This result slightly complies with the work done by Eamonn et al. [92] as they pointed that for an application that have a small functions there is no need for graphical representations of the gestures. Moreover, when users get trained to their gestures, a learning effect could be seen for speed and accuracy in scenario 1 and 2. Subjects in scenario 3 used headphones tapping gesture although it was new for users. They took time to adjust and understand how the tapping gesture works. Thats why in session 3, we got slightly increase in error rates.

3.7.3 Experiment 2

The main goal of the second experiment was to proof that when objects change, people like to change their gesture shapes. Moreover, we need to find a tendency between objects and gestures. We asked 12 subjects aged between 22 and 35 with an average of 27.5 to conduct the experiment. The average level of experience for using object gestures for the subjects were 45%. The subjects use an average of 45 object in everyday life. The experiment composed of using 4 objects (ball, headphone, stick and box), and two positions per object, first position using predefined profiles and the other position is using adjusted profiles, a total of 8 positions. Subjects asked to perform two sessions in each position. Figure 3.11 shows the objects used in the experiments. We selected those objects according to some criteria shown in Table 3.2. First object is called “ No object” which is performed using hand gestures only. The ball object is held by two hands and it is thick, similarly was the box object. The stick was held by one hand and it was described as thin object. The headphone has some common parameters with the ball and box objects as they are held by two hands. However, people normally use headphones by attaching it to their heads.

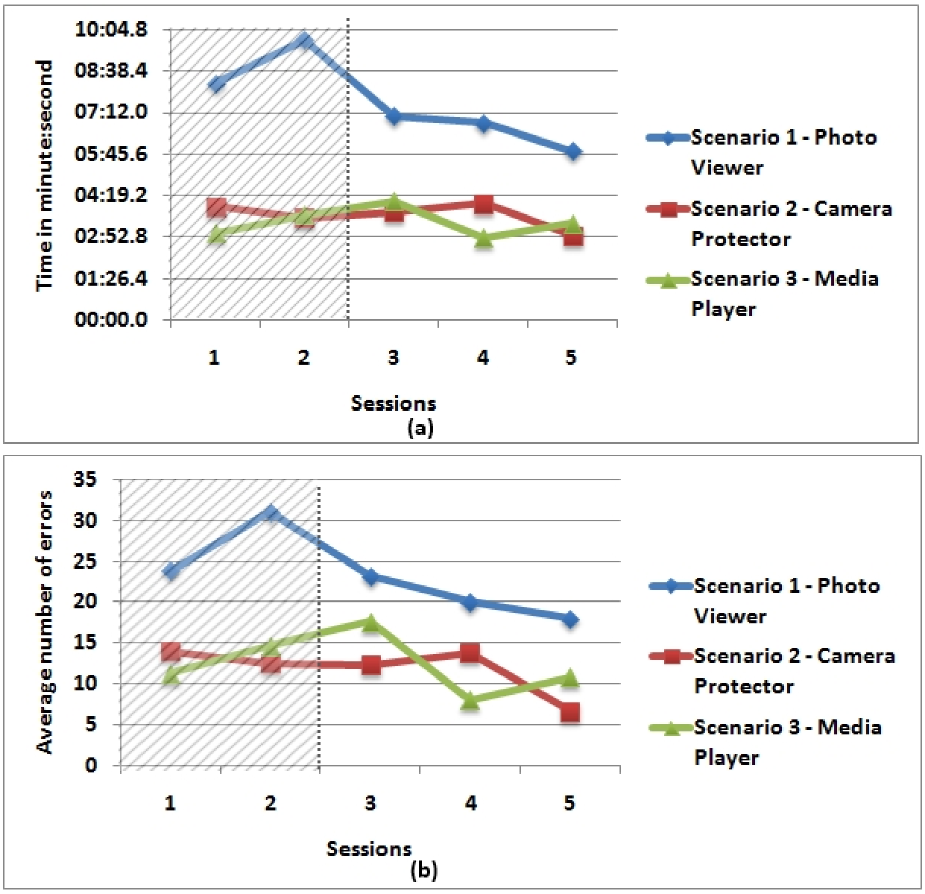


Figure 3.10: Average time to finish each session (a), Average number of errors per session (b), Shaded area for SGS object profile

Table 3.2: Objects used in the experiments parameter

Objects	1 hand	2 hands	Thin	Thick
No object	⊕			
Ball		⊕		⊕
Box		⊕		⊕
Stick	⊕		⊕	
Headphone		⊕		

We divided the subjects into two groups with different sequence of using objects called Graoup A and Group B. Figure 3.12 shows the difference between the two groups A and B and shows the profiles recommended by SGS for both groups. In the beginning of the experiments, all subjects started from a hand gesture profile defined in Figure 2.7 by one of the authors of the thesis.

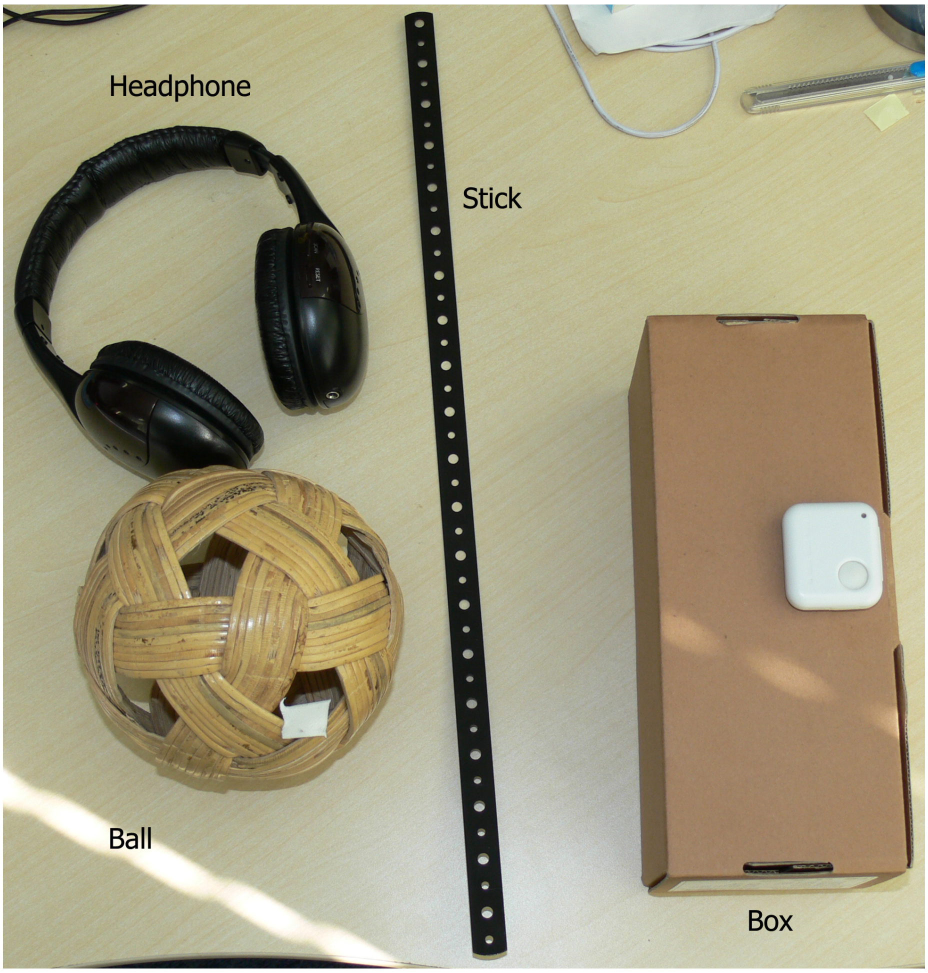


Figure 3.11: Objects used for the experiments.

Each subject requested to adjust the gestures that he did not like by creating them by himself. Some guide lines for creating gestures using different objects were given to subjects to assist them while doing the experiments. The adjusted profiles for the ball in group A or the headphone in group B will be used for the other objects in the experiment. In group A, the headphone gesture profile will load the same gesture profile of ball because SGS will compare the attributes of headphone, ball and box and it will find that ball and box share the same attributes with headphone as shown in Table 3.2. In Group B SGS selected the ball profile for the box cause it shares more attributes than headphone.

The subjects have to fill satisfaction questionnaire for all the gestures they used on a Likert scale of 5 points, and then they were interviewed for some feedback.

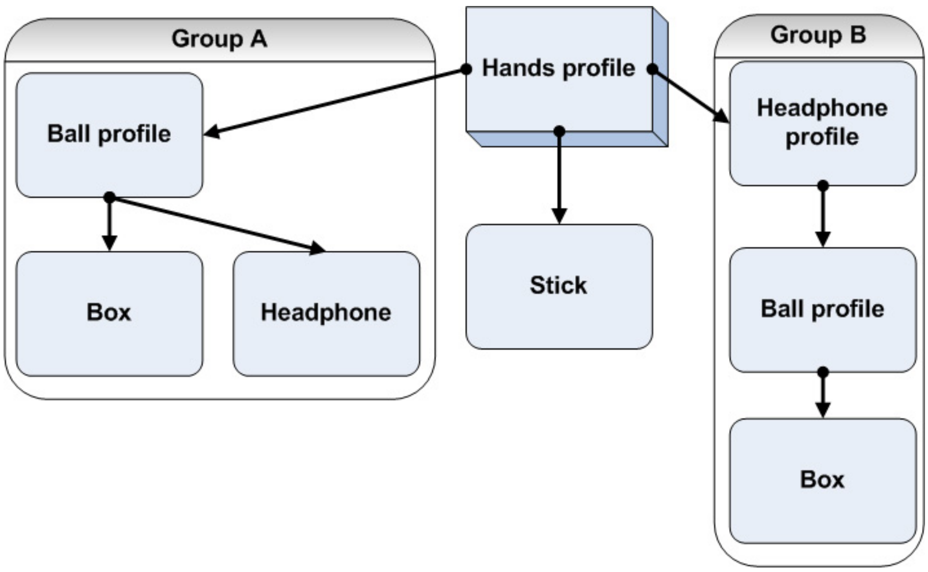


Figure 3.12: Group A and B sequence of doing experiments and profiles downloaded for each session.

3.7.4 Results

The more the subjects were not satisfied by gestures the more they change the objects gesture shapes. When the objects changed, people changed the gestures according to the difference between objects attributes like, size, handling position. Figure 3.13 shows the satisfaction percentage and changes of gestures percentage. We have noticed that subjects did not like to change gestures of the box and stick objects because they are held same way as the ball or hands respectively. We find a strong correlation value of -0.95 between the object satisfaction and percentage of changing gestures.

Some objects like stick have the same accuracy as that of hands. Others fails to perform similar accuracy like the ball or headphones because of size and handling position are different. Figure 3.14 shows the comparison for the predefined gesture profile of hand gestures when performed by different objects.

We have noticed that the gestures done by subjects using head cannot be the same done by stick as they have a different holding position, also gestures done by ball cannot be same gestures for stick because the size of stick and the ball is different in means of thickness. The results also show us that if subjects adjusted objects gestures, they can interact more accurately. Figure 3.15 shows the accuracy comparison of gestures before and after adjusting of gestures.

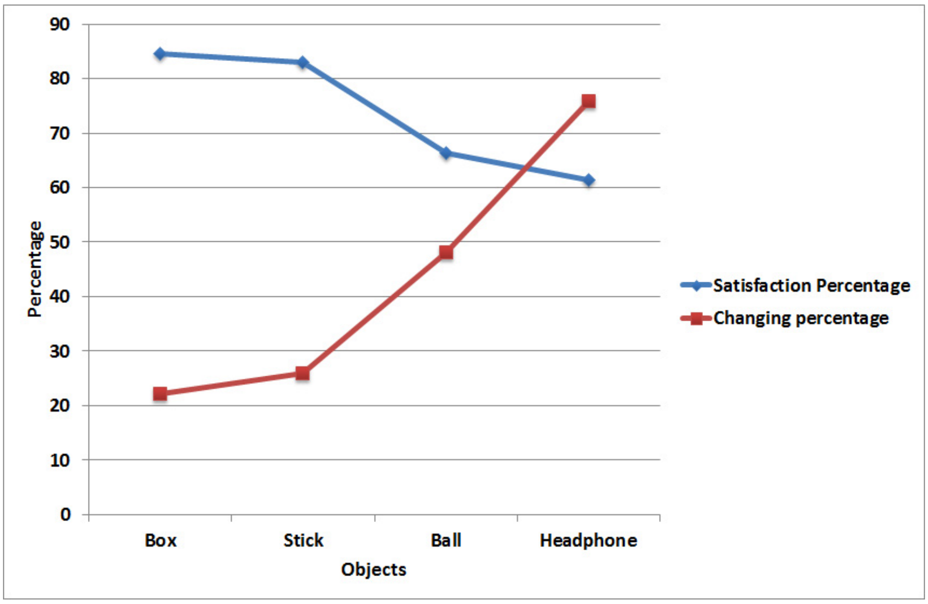


Figure 3.13: Objects satisfaction and gesture changes percentage.

The results show us that whenever users adjusted their gesture profile the accuracy almost remains above the 85% with high accuracy for the box and headphones. A one way ANOVA to test for the difference among the accuracy for original and adjusted profiles. There was a significant difference between original and adjusted profiles, $F(1, 8) = 8.2, P < 0.05, F_{crit} = 5.98$.

3.8 Discussions

We analyzed two of the parameters for selecting profiles objects attachment and users skills in more details. We found that objects that were fixed on the head like the headphone have more accurate results. We think that the head has a fixed initial position for all subjects. The movement of the head has a limited spatial for doing gestures, so users depend on tilting and tapping gestures to perform their gestures. The tapping gesture can be easily achieved by most of the subjects because of its fixed pattern. Figure 3.16 (a) shows that when users used the objects that fixed on the head can get less number of errors for both SGS profile or adjusted profiles. A one way ANOVA to test for the difference among the number of errors between objects fixed on hands and objects fixed on head.

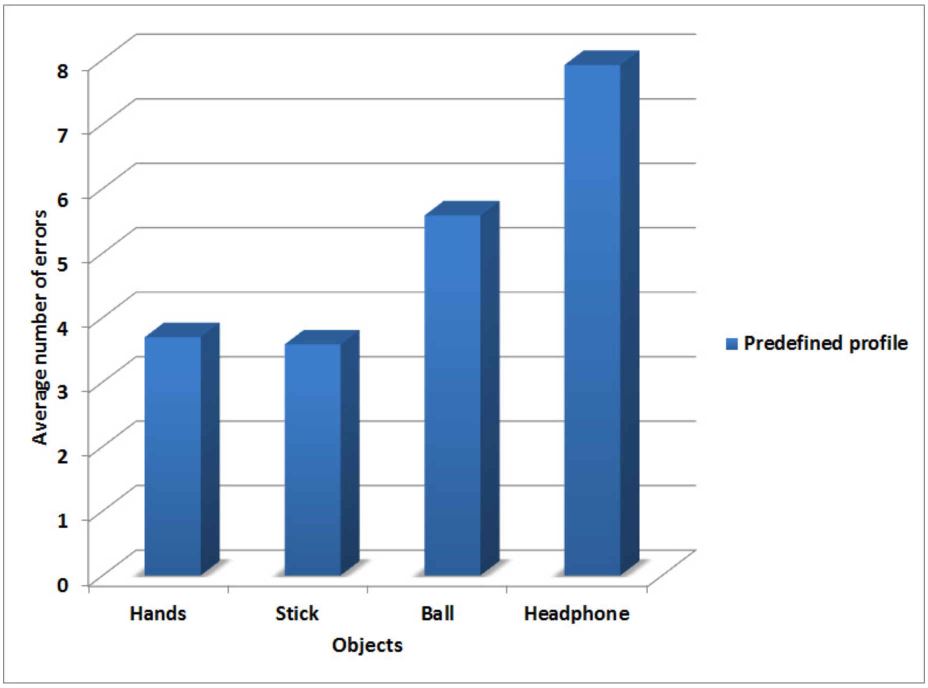


Figure 3.14: Comparing hand gestures predefined profile to stick, ball and headphone objects.

There was a significant difference between objects fixed on hands and objects fixed on head, $F(1,8) = 6.3, P < 0.05, F_{crit} = 5.31$.

We classify the subjects according to their previous experience in using gestures for interactions. It can be shown in Figure 3.16 (b) that beginner users have the highest error counts; however they show enhanced in their interaction after adjusting their gestures. A one way ANOVA to test for the difference among the number of errors between beginner users and moderate, expert users. There was a significant difference between beginner and moderate, expert users, $F(1,8) = 14.6, P < 0.05, F_{crit} = 5.31$. Expert users showed that they were not comfortable of the provided SGS profile and when they adjusted some gestures their accuracy enhanced. The reduction of the errors was due to the innovative gestures that performed by the expert subjects. However, moderate users show a constant error level for for SGS provided profile and their adjusted profiles. A one way ANOVA to test for the difference among the number of errors between moderate and expert users. There was no significant difference between moderate and expert users, $F(1,6) = 0.7, P > 0.05, F_{crit} = 5.98$. Thus, we expect that SGS can support normal users with appropriate

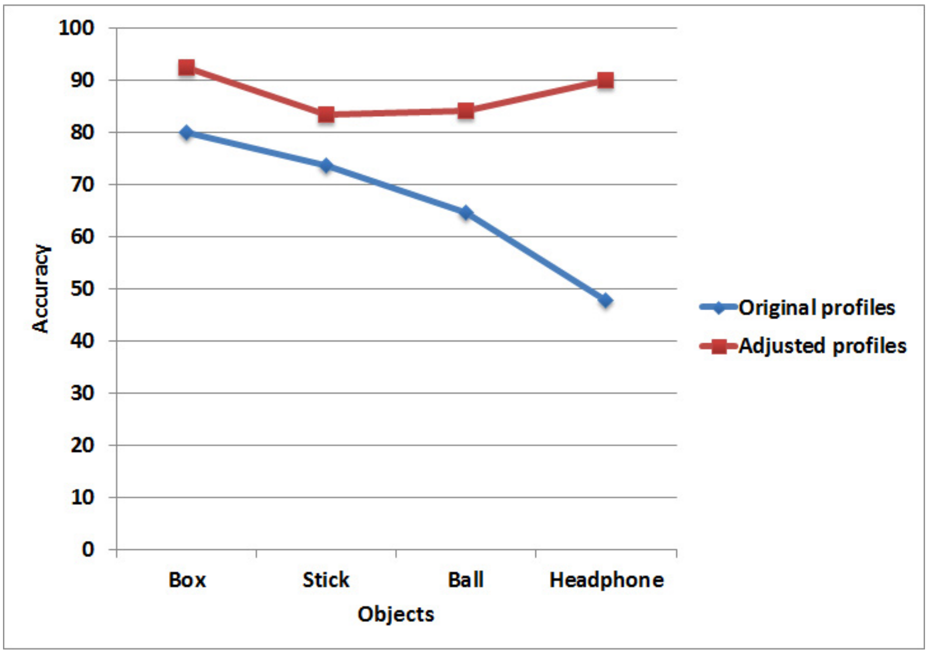


Figure 3.15: Comparing of accuracy of object gesture before and after adjusting profiles.

profiles. SGS supported beginner and expert users to adjust and use their own customized object gestures.

We get also some feedback from the subjects regarding their feedback for SGS and the fundamental parameters that can change the object gestures. Table 3.3 shows feedback we get from the users interviewing. 83% of the subjects think that using gestures for the ball object is similar to the gestures they will define for the box. 75% of subjects think that stick and cellphone could have similar objects. However, the subjects who have defined hitting the table using the stick say that similar gesture is extremely difficult to achieve using cellular phone. The results for ball and headphone were quite natural, as 59% of the subjects think that some gestures like the tilting gestures have similarity. 90% of the subjects agree that the helmet has similar gestures to that of headphone. However, helmet is heavier than headphones, so some subjects were curious about the weight of the helmet. The importance of the parameters that can change the object gesture shapes (size, lenght, position) was rated in importance as 82%, 77% and 85% respectively. 91% of subjects think that using objects around them can enrich their interaction in the environment. The majority of the subjects agree that they like to have their object gestures downloaded to them according to their context. The satisfaction towards the recommended gestures of

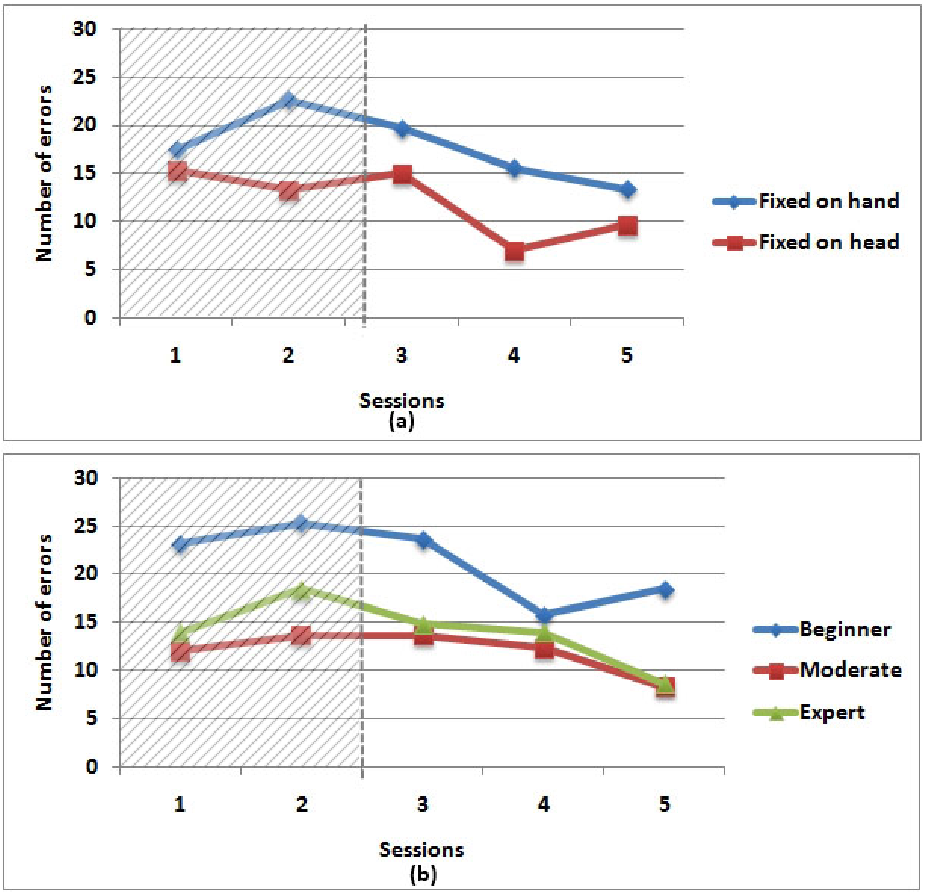


Figure 3.16: Total average error for object attachment parameter (a), User preference skill level (b), Shaded area for SGS object profile

SGS was 77%. We think this percentage could be enhanced more when people practice their gestures profiles more frequently.

3.9 Summary

We have presented SGS system, which is capable of understanding users object gestures in ubiquitous environment. We conducted a primary experiment to understand how users think about object gestures, how they prefer to adjust their gestures and appropriate objects for interaction. In general we found that predefined gestures are not always sufficient for users interactions. We think also that when people change objects they are interacting with, their gestures will also change. We presented a system called “Smart Gesture Sticker (SGS)” that supports users while interacting with objects in ubiquitous environments. SGS

Table 3.3: Users feedback

Parameter	Percentage
You like to use same gestures for the ball to box	83%
You like to use same gestures for stick to cellphone	75%
You like to use same gestures for the ball to headphone	59%
You like to use same gestures for headphone to helmet	90%
Importance of object Size	82%
Importance of object Length	77%
Importance of object Position	85%
Can objects enrich interaction in environment	91%
You like object gestures to be downloaded	98%
Do you like recommended gestures	77%

was presented and used to implement smart applications with daily life objects. SGS evaluated by applying different scenarios and situations for users. Results show that SGS can assist and support users to interact with their environment. SGS provide sufficient gesture profiles that can allow users to discover new usage for objects around them. It was shown that people can interact more accurately than hand gestures, if and only if they used some appropriate objects. We think that vast applications that run with objects can get use of SGS, Smart door key, smart meeting rooms and others.

Chapter 4

Tilting interface methods and applications

4.1 Introduction

Human gestures are typical examples of nonverbal communication and help people to communicate smoothly. The substantial developments made in smart devices and small sensors are directly affecting the arena of human-computer interaction and smart ubiquitous environments. As a result, many researchers have considered using sensors to capture users environment and the humans intentions in order to get more intuitive interactions. A typical use of sensors in daily life is to capture user activity presented by in the work done by Nishkam et al. [65] and Slyper et al. [93].

Sensors were also used for the interaction with objects in the work done by Feldman et al. [94] and Ayman et al. [63] [95]. Interaction in the ubiquitous environment could suffer from the absence of direct, conventional interaction methods like keyboards, mice, and direct control devices. Conventional methods designed mainly for stationary interaction situations. Using a camera to capture hand gestures depends on the lighting of an environment and requires one to face the camera, which limits mobility and involves privacy issues. Another method of interaction is using full hand gestures in ubiquitous environments. Some situations, like being on public transportation, however, are not conducive to full hand movements due to limited space. Hand gestures could also have unintended social implications based on the shape of the gesture [84] [39]. Finally, when users interact with applications using hand gestures for long periods of time, they may suffer from hand tremors and pain, to an increase in error rates.

The above issues relating to hand gestures for remote interactions have brought about the use of tilt gestures, as they depend merely on moving the wrist. Rath et al. [96] stated that people use human tilt hand gestures to express directions or to alter the position of some object in hand. Many advantages exist for tilt gestures over hand gestures. First, they require less space to be performed. Second, they can be easily memorized because they have a fixed number of positions. Third, the gestures can be performed at a faster speed in comparison with full hand gesture movements because they depend on moving only the wrist.

Tilt gestures, however, also possess some research challenges for ubiquitous environment interaction, as they are limited in the number of actions they can perform. Furthermore, the interaction accuracy must be high, and the recognition engine must distinguish among different tilt directions and levels performed by a user. Finally, the suitable applications that benefit from tilt gestures in ubiquitous environments should be further studied. Users in some situations might be involved in other activities or may not have a free hand. Consequently, Everyday objects like pens, bottles, books, headphones, balls, cellular phones, and computers can act as input devices by using tilt gestures.

4.2 Related work

Many techniques using hand gestures that captured and tracked by a camera discussed in the literature. Sawasaki et al. [97] and Azaz et al. [98] developed a camera-based system that captures the position of the hand and applies filtering techniques to convert it into mouse positions. Using a camera, however, could result in decreased accuracy and low-processing power, and may be difficult to set up in ubiquitous environments. Another method to capture full hand gestures is through sensors. Lee et al. [99] developed the I-throw system to integrate location awareness and hand gestures. Farella et al. [75] developed a glove using bending sensors for hand interaction. Wobbrock [100] designed a $1 recognition engine for recognizing and classifying hand gestures using cheap devices.

Tilting operations for small screen interfaces introduced by Rekimoto [101], who showed the advantages of having a tilt interface as an input device, especially in field work with the use of one-hand interaction. Furthermore, they pointed out the advantages of the tilt interface in the interaction with small screen devices that are not compatible with a pen device. However, the developed prototype system depends on combining tilt and normal device buttons for interaction. The lack of input devices present a challenge in the ubiquitous environment that requires further research. Levin et al. [102] reported a study on embedded accelerometers as a gesture interface for an extremely small computer using tilt and shake gestures for drawing. Lee et al. [103] proposed a tilting table as a movable screen to interact with several applications such as “Beadball” and “Cross-Being: Todd” tables, tilt maps, and bio browsers. They used the tilt gestures to move a display screen in four directions only. Tilting gestures used to generate a synchronous gesture for multi or single user entertainment in the work done by Hinckley et al. [104]. They proposed different gestures like bumping and angle of tilting to tile different screen devices and copying some object from one device to another.

Hence, we thought about the suitable number of tilt gestures to be executed per application. Cho et al. [105] explored the use of the tilting gesture for a multi-context photo browsing on mobile devices and addressing some issues of the tilt interface, such as overshooting or a partial image problem. Partidge et al. [106] proposed a tilting device using accelerometer-supported text entry for small devices. They proposed a wristwatch with buttons for selecting the letter pad and use of the tilt to choose among letters. Their device, however, requires using both hands to control the device. Jani et al. [107] proposed a scan-and-tilt interaction technique for more natural interaction in museum guide applications. Andrew et al. [108] studied the variability in wrist-tilt accelerometer-based gesture interaction. They showed an increase in variability of motions upwards from the center in comparison to downwards. They analyzed trajectories of accelerometer values and put in place design standards for tilt-based hand interactions. During their experiments, they controlled mouse cursor movements to hit targets using tilt directions.

Antii et al. [109] proposed a controller for an “on the move” mobile media player using gestures and audio metaphors. They discussed the importance of using the gestures to control the medial player while users involved in other activities, such as walking or running. They pointed out the difficulty of operating small buttons inside small devices and the need for visual focus to control the media player buttons. They showed a limited number of gestures to command sound-based basic menus. However, they did not propose a method to control a menu with more functions and commands. H.Kimura, et. al [110] defined three motions rotating, tilting, and knocking as a predefined gestures for interaction. They

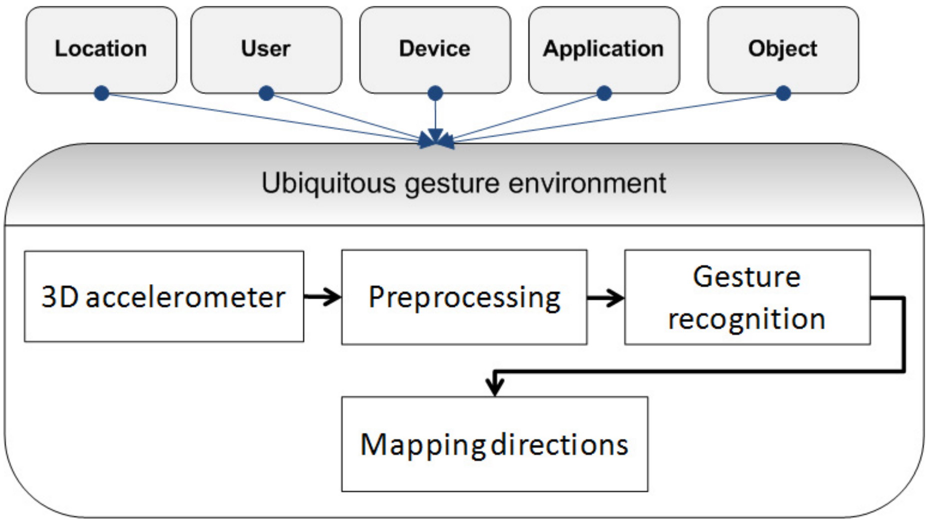


Figure 4.1: Tilt gestures system overview

used those gestures for map viewer application. Users observed less motion accuracy and time delay between gesture and system action. This research does not define personalized gestures. To the best of our knowledge, the number of targets and the applications suitable for tilt gestures in ubiquitous environment have not been studied. In this study, we explored applications and tilt gesture mapping techniques. We also studied the suitable range of commands that can be achieved by tilt gestures.

4.3 Tilting gestures in ubiquitous environment proposed system

4.3.1 System overview

We extended UbiGesture [84] for capturing user context in ubiquitous environments. The context of the user is composed of location, user preferences, device, application and object for interaction. Figure 4.1 illustrates the system overview. UbiGesture will record the captured context parameters in shared storage, so it can be accessed by the proposed tilting interface. Context parameters include location, user, device, application and object used. Once the interface receives the context environment parameters, it will load the appropriate tilt gestures profile for the user. If the user has not defined any profile for this context, then it will download a default predefined profile for tilt gestures.

The interface connects to a 3D accelerometer sensor then pass by a preprocessing phase to enhance the recognition of tilt gesture. The tilt gesture recognition engine is done through the usage of predefined threshold values. Finally, the tilt gesture will be mapped to some application function to be executed to control the application running on the device.

4.3.2 Sensors used

In this research, we used two types of sensors, first called Nokia cookie sensor [111] Nokia cookie is an experimental device developed to test ubiquitous context aware applications. The chassis contains 8 sensors and sends data using Bluetooth. The device is about the size of 5 stacked quarter coins. The sensor composed of 3 axis linear accelerometer, Compass 3 axis sensor. There are other extension sensors that can be attached to cookie base giving more flexibility to the sensor. Sensors that can be attached are ambient light sensor extension board (RGB color, UV), force vibration motor, galvanic skin response sensor, heart rate sensor and skin temperature sensor see Figure 4.2 (a). The sensor has two communication interfaces Bluetooth and UART wired connection.

The second sensor called WAA-006 sensor developed by [44], it can be attached easily to various objects. Figure 4.2 (b) shows the 3D accelerometer sensor. It has a built-in 3D accelerometer, angular rotation rate sensor, and temperature sensor. The sensor sends data as a pattern of signals that represent the acceleration measurements applied over the sensor's three axes. When a user tilts the 3D accelerometer sensor in any direction, the acceleration magnitude will be loaded on the axis along which motion was performed, plus the gravity value. The output of the accelerometer is a strip sequence of 3D points called G, such that a point over G is represented by $ht_{ax,ay,az}$. The time stamp generated automatically by the sensor is denoted by ht, and records from the accelerometer at time t are denoted by $t_{ax,ay,az}$. The angle of tilting among the accelerometer axes is calculated in equation 4.1.

$$Angle = \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{z}\right) \tag{4.1}$$

4.3.3 Pre-processing accelerometer data

There are two approaches for capturing gestures using accelerometers, discrete and continuous gesture recognition. Discrete gestures must be initiated by the user, who determines the start and end of each gesture by hitting buttons or clicking. On the other hand, contin-

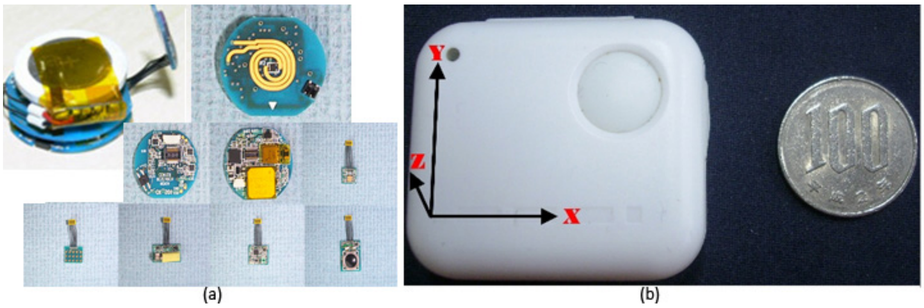


Figure 4.2: (a) Nokia cookie sensor. (b) WAA-006 Sensor

uous gesture recognition is captured directly from user hand movements. Using the discrete gesture recognition is not intuitive for users as they must use another device or remain focus on precisely when to hit and release the button. Thus, we used the continuous gesture recognition to enrich the interaction with devices. After making the tilt gesture, however, the user must hold the position for a period (less than 200 ms) for the movement to be recognized by the system. This value can be customized by the user depending on the interaction application. To ensure that the system captures the correct tilt gesture performed by the user, the system keeps a continuous record of six successive accelerometer points. The pre-processor calculates the standard deviation of the six points; if the standard deviation is small, this means that the points are near one another. Thus, it indicates that the user's tilt hand is fixed in this direction, and the position is not classified as noise. The system calculates the average of the six points as the users tilt gesture point in 3D space denoting it by $G_{x,y,z}$.

4.3.4 Capturing tilt gestures

Tilting the sensor in different directions will add gravity values over the x, y, and z axis respectively. We have to define three threshold values Tx, Ty, and Tz. We measure these threshold values for each of the basic 8 directions (North, NorthWest, North east, east and west, south, south east, and south west). Also, we measure the threshold values for the position of no tilting, we call this steady state. These positions must have no conflicts between each other. If the user tilts the sensor in some direction, the system will match the value of G with the defined threshold values. Table 4.1 shows each direction and the axis used for comparing threshold values. The system starts comparing with directions that use

Table 4.1: Directions and axis used.

Direction	X	Y	Z
North East	$\oplus$	$\oplus$	$\oplus$
North West	$\oplus$	$\oplus$	$\oplus$
South East	$\oplus$	$\oplus$	$\oplus$
South West	$\oplus$	$\oplus$	$\oplus$
Steady State	$\oplus$	$\oplus$	$\oplus$
North		$\oplus$	$\oplus$
South		$\oplus$	$\oplus$
East	$\oplus$		$\oplus$
West	$\oplus$		$\oplus$

3 axes: northeast, northwest, southeast, southwest then steady state. After that, system compares directions that use 2 axes north, south, east and west.

The gravity value denoted by 'G' can be shown on Figure 4.3 (a), the direction of gravity over the x axis while tilting the sensor in the east and west (left and right) directions. Furthermore, the extracted motion change of G values over the x axis can be shown in Figure 4.3 (b). The squares marked can show the threshold values for capturing the start and end of tilting gesture to the left and tilting to the right. The area before and after those rectangles are marked as steady state direction. The threshold values are defined by ranges of values after an empirical study to determine appropriate values for those threshold values for each object used.

The system stores information for each application, object, available directions and 3D point for each direction. This information is provided to the interface through the UbiGesture system. The tilt directions are represented as points in 3D space. When the user performs a new tilting gesture, the system first ascertain whether the users hand is intentionally fixed in this position, and then it will calculate the minimum distance with all the stored points, and returns the appropriate direction.

4.4 Applications of tilt gestures

We studied some of the applications that could be suited for tilt gesture mapping. Applications as Google Earth, web browsers, media players, PTZ Camera Protector, Japanese text tool called “Popie” [112], presentation viewers, image viewers and several gaming programs tested with tilt gestures. We have selected four samples of application types that

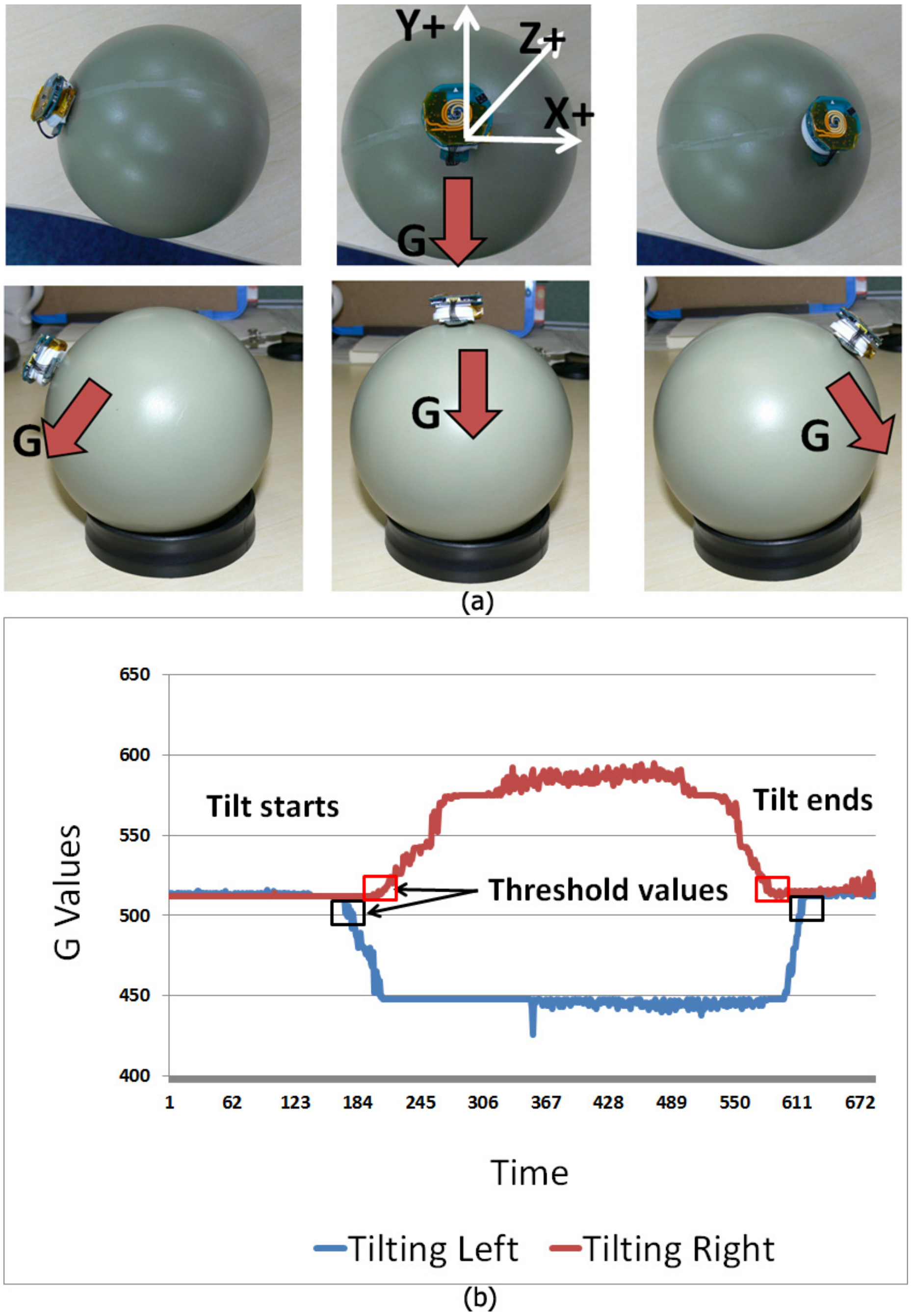


Figure 4.3: (a) Tilting to left and right and effect of gravity directions. (b) Threshold values for left / right tilting

range in function and are suitable for remote interactions to evaluate the usability of the system.

4.4.1 Presentation viewers

Computer-based presentation viewers are commonly used by many users. Currently, a user must bend over to look for an icon on the display to change the slides, or press a button on the keyboard to control the slides. Other alternatives for interaction are using a remote mouse or asking an assistant to flip slides, both of which interrupt the flow of a presentation. In this situation, hand gestures could be appropriate, as they are a method of nonverbal communication between the presenter and audience. The presenter, however, cannot execute full hand gestures to flip slides, as this could be disruptive. Hence, tilt gestures are useful because they can be performed subtly in a small space. In our experiment, we assume that the user has created his presentation slides with commercial presentation software such as Microsofts PowerPoint. The user attaches the sensor directly to his hand or another object and can browse through the presentation slides directly by performing tilt gestures.

We presented a method for users to interact with presentation slides through tilt gestures by executing two successive tilt motions in the desired direction, and then returning to the initial position. Figure 4.4 shows a user browsing presentation slides backwards and forwards.

We choose one-to-one direct mapping between operation and gestures, so users easily manipulate the slides with a minimum number of gestures. The reverse-operation nature of browsing PowerPoint slides as next/previous slides was mapped directly to right/left tilt gestures. Table 4.2 shows the slide-show operations and mapping gestures. The close operation was mapped to the down-right tilt direction. It was shown in the results of Andrew et al. [108] that the down-right tilt direction was the easiest diagonal direction for right handed users.

4.4.2 Photo viewers

Photo browsing applications are common and used in different situations. People like to browse through pictures with their families on large displays in living rooms. Students like to share pictures among each other using mobile devices. We implemented a photo browsing application that displays the stored pictures in rows and columns. Figure 4.5 shows a user selecting a picture from a grid and opening the picture in a new window. The combinations of tilt up then tilt right or left are triggers to execute commands.

Figure 4.4: User browsing presentation slides.

Table 4.2: Slide show operations and mapping gestures.

Operation	**Description**	**Tilt gesture**
Next	Advance to next slide	Right
Previous	Return to previous slide	Left
Home	Go to first slide	Up
End	Go to last slide	Down
Close	End slide show	Down-right

The system maps the operations to move right, left, up, and down as direct mapping for tilt gestures, right, left, up, and down respectively. There is also a function for adjusting the brightness of an image by a certain value. The user opens the image in a new window and selects the up tilt gesture to select the "adjust image brightness" function. Then, he/she either increase or decrease the brightness value by right and left tilting gestures. Table 4.3 shows the mapping between operations and gestures.

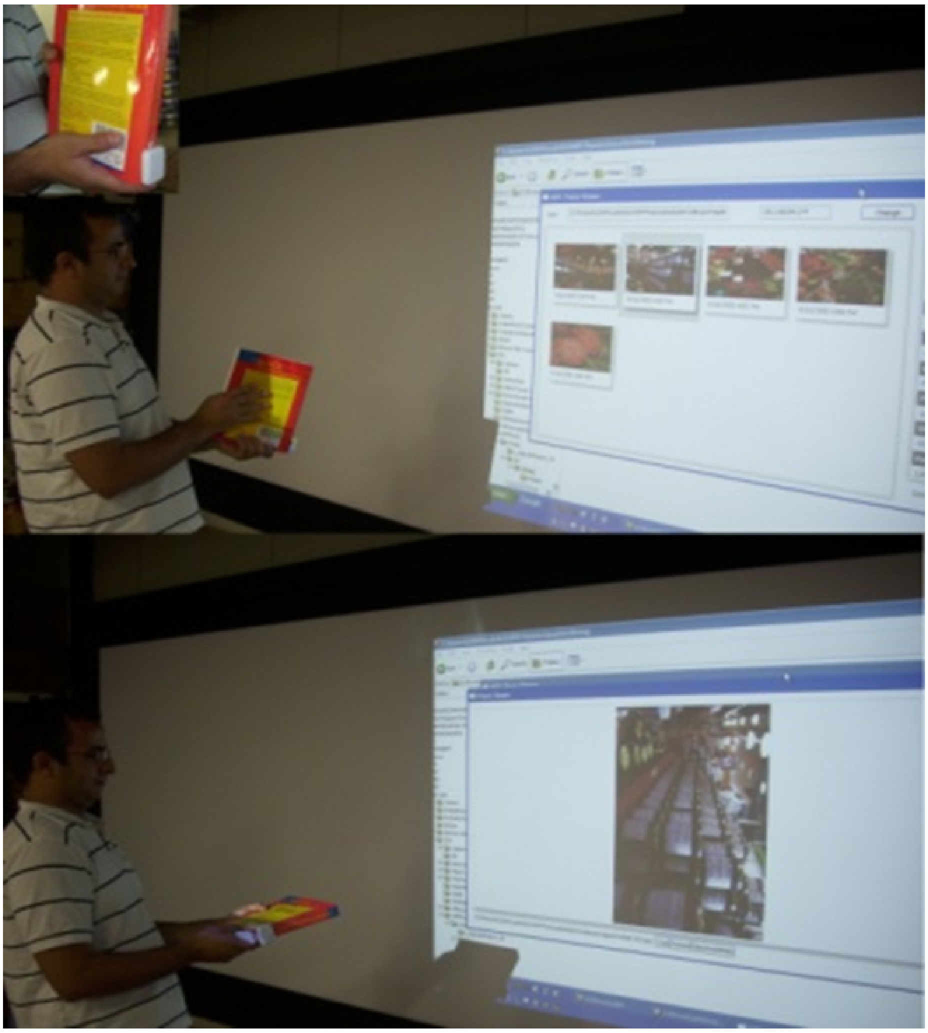


Figure 4.5: User selecting picture from grid, and opening the picture in a new window.

4.4.3 Gaming systems

Gaming systems are an attractive field for many researchers to study user feedback and usability of systems. We tested a sample of gaming interaction using a large-screen display in our laboratory and a flight simulator game. Figure 4.6 shows tilting sensor to control flight speed with levels and interacting with flight simulator game.

The game operated by four directional movements to control the mouse cursor on the screen. The user attaches the sensor to a flight object toy. The system extracts motion while interacting with the remote display screen. The displacement value calculated according to the captured acceleration, screen resolution factor, and angle of the three axes of the

Table 4.3: Photo browser application operations and tilt gestures.

Operation	Description	Tilt gesture
Browsing mode		
Move right	Advance cursor 1 column	Right
Move left	Return cursor 1 column	Left
Move up	Advance cursor 1 row	Up
Move down	return cursor 1 row	Down
View picture	Open picture in new window	Down-right
Editing mode		
Increase brightness	Increase brightness value	Up then right
Decrease brightness	Decrease brightness value	Up then left
Black and white filter	Grey scale the picture	Down
Close	Close current	window Left

Table 4.4: Flight simulator game operations and gestures.

Operation	Description	Tilt gesture
Move right	Advance mouse cursor over x-axis	Right
Move left	Reduce mouse cursor over x-axis	Left
Move up	Reduce mouse cursor over y-axis	Up
Move down	Advance mouse cursor over y-axis	Down

accelerometer. The interface then calculates remote screen mouse display positions. The more the user tilts the object in one direction, the faster the flight speed on the screen becomes. The center point of the remote display screen is the initial starting point for cursor movement. This displacement value used to set the new position on the remote display screen. If the user holds the sensor horizontally towards the ground, the cursor will not move. Table 4.4 shows the flight simulator commands and mapped tilt gestures.

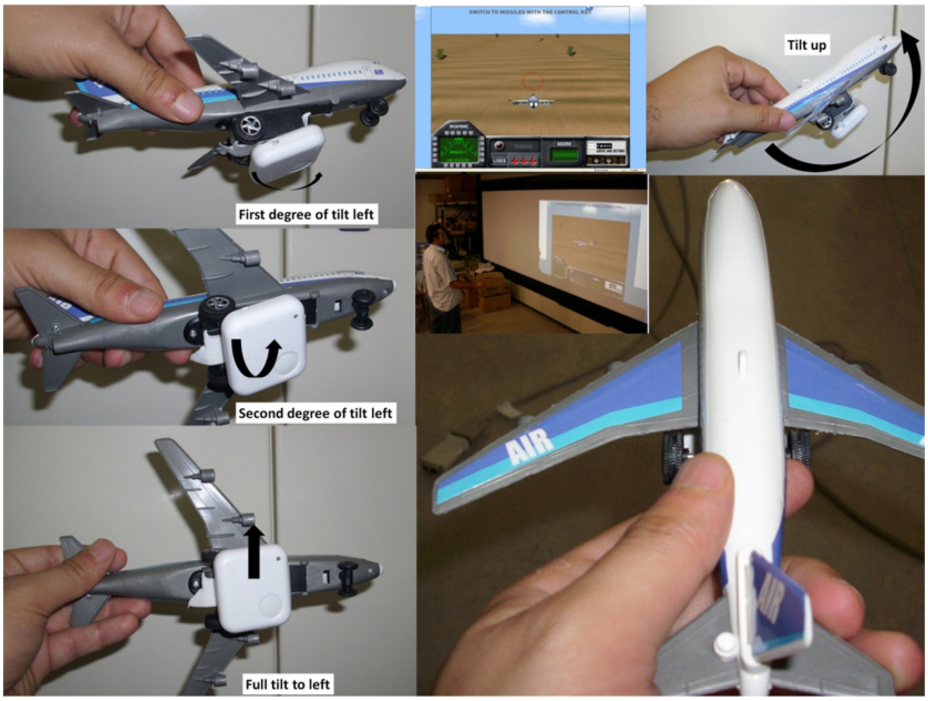


Figure 4.6: Tilting sensor to control flight speed and interacting with flight simulator game.

4.4.4 Popie

Popie is a tool that facilitates entering Japanese text based on flow menu. Tilt gestures interface is a promising interface because of its fast and natural of usage. Interaction with Popie using tilt hand gestures in ubiquitous environments has a lot of potentials to overcome the nonexistence of direct input methods. Tilt gesture has a lot of possibilities to operate Popie remotely in a faster manner compared to hand gestures. Conventional methods for text entry like keyboards have been used efficiently. Keyboards could sustain from the limitation of space to be hold. It should be hold in a stable state for efficient and fast input and the user should be hands free. Users would like to check their email, check train schedule and write on their PDA, write notes on move. In these situations, standard keyboard is not the best choice. Touch panels are one possible alternative for entering text using tap operations. However, it brings some problems when the touch panel display became bigger. Software keyboards have to be operated while users are standing beside the display, so they can make the tap operations, this cause a limitation for user freedom and put a heavy strain on people.

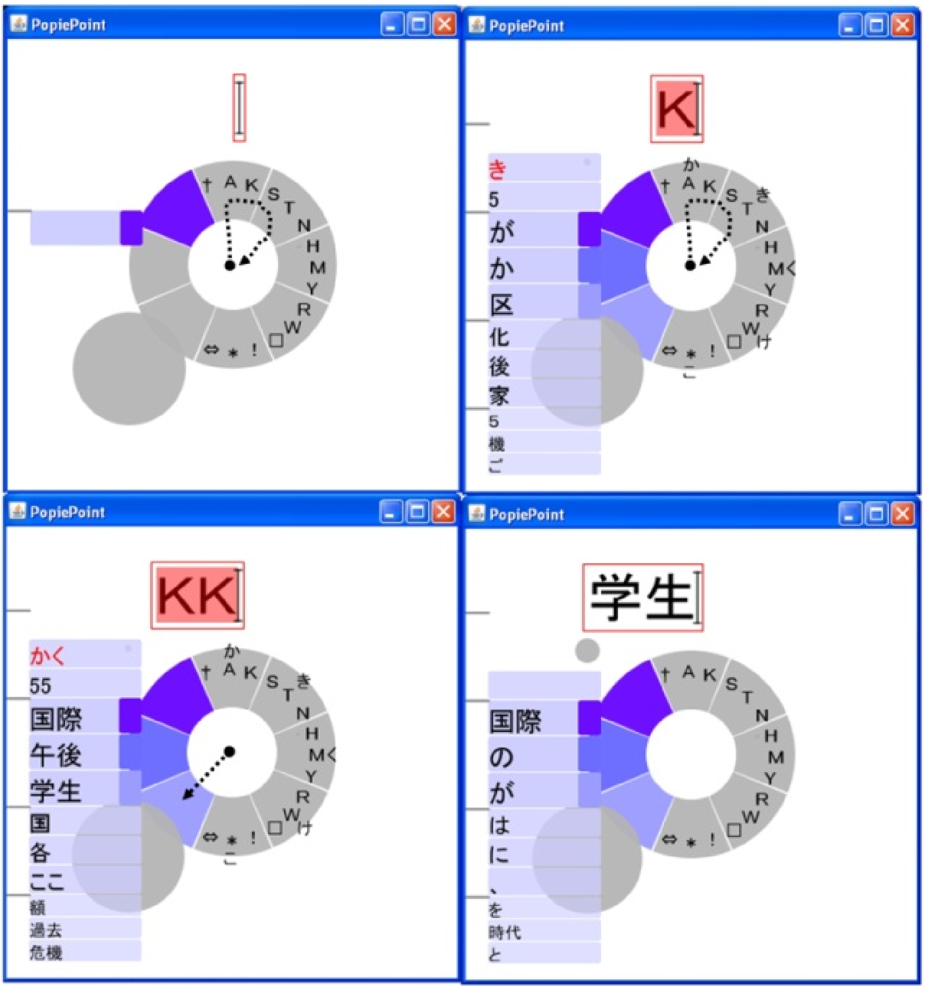


Figure 4.7: Original directions to enter the word "Gakusei" showed by arrows.

Popie originally operated by 3 steps: (a) User selects one of the basic 8 directions. (b) User chooses either to move right or left relatively or to go the next step. (c) User returns back to the rest area. If the user wants to enter the word "Gakusei" [1], he should select appropriate constants characters which are KKSA, then select the word from the candidate words. Figure 4.7 shows original directions to enter the word "Gakusei" showed by arrows. Table 4.5 shows Popie key configurations and mapped tilt gestures. If the user want to select "ka" constant, he/she must enter tilt gestures in this sequence "up- upright- rest area".

[1]Gakusei means student in Japanese language.

Table 4.5: Flight simulator game operations and gestures.

Operation	**Tilt gesture**	**Operation**	**Tilt gesture**
"A"	up	"K"	up then up-right
"T"	upright	"S"	upright then up
"N"	upright then right	"M"	right
"H"	right then up-right	"Y"	right then downright
"W"	downright	"R"	downright then right
Space	downright then down	Return	downright then downleft
Tab	downright then left	Symbol	down then downright
Delete	down	Undo	down then downleft
Scroll up	left then up	Scroll down	left then down
Candidate select	left , upleft , downleft		

4.5 Classifying tilt application fields

We have classified the mapping of tilt gestures according to application needs. Tilt gestures mapped to application commands in means of direct, sequence and mouse mapped.

4.5.1 Direct mapping

A normal user can make hand tilt gestures in two, four, six or eight directions. A typical movement for direct mapping gesture composed of two movements. First, the user must start from a steady state, and choose the direction of movement. Second, the user must return to the steady state. In the direct mapping, each tilt gesture mapped to one command. If the user wants to move in more over than eight directions, he/she could lose the sense of the direction of hands in space. The limit of the movement space is one of the main challenges of hand tilt. To overcome the limited space of hand-tilt gestures, we presented the concept of the tilt gesture level, so that each of the eight basic directions can have up to three levels of tilting. Figure 4.8 shows three levels for the up-right tilt gesture.

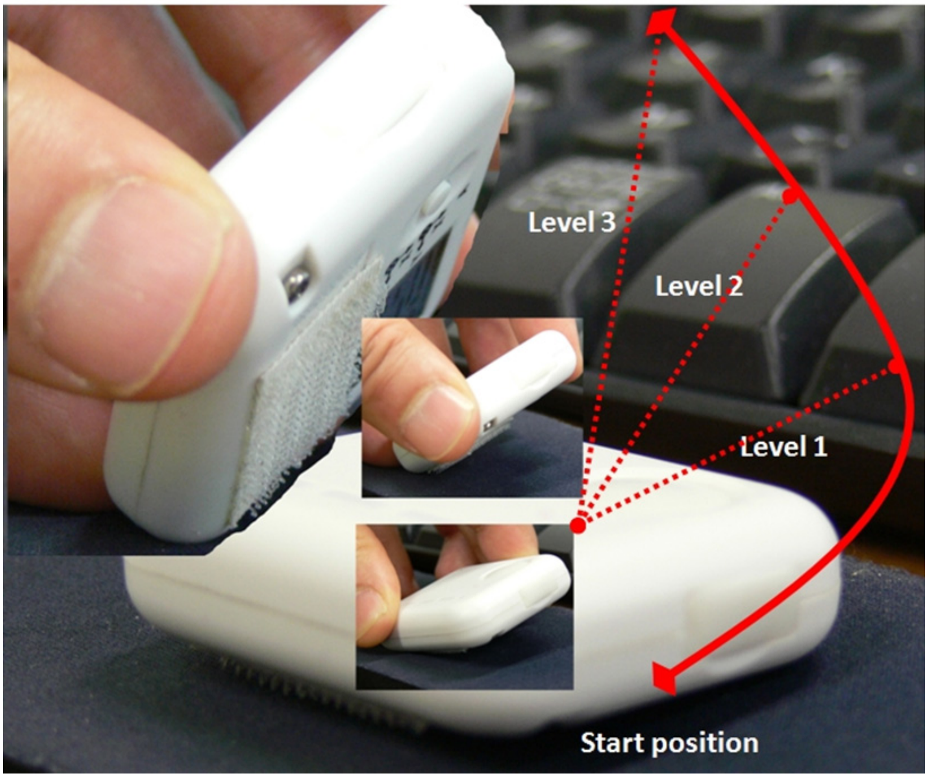


Figure 4.8: Tilt levels for the up right direction.

This combination could allow us to provide up to 24 gestures for an application (eight directions with three levels each). Several commercial and developed applications can utilize this method, such as presentation viewers, photo viewers, media players and map-browsing applications like Google Earth. Also, some applications like PTZ (pan, tilt, zoom) camera protector application require the use of many functions, such as controlling PTZ or redirecting the camera to predesignated targets.

4.5.2 Tilt sequence codes

In some situations, when too many commands used to control an application, a sequence of consecutive tilt gestures can be used. A tilt gesture is built from two, three, four, or even more sequential tilt gestures. First, the user starts from a steady state and selects the first direction. Then, the user performs one or more other directions. The interface keeps a record of this sequence until the user has finished the sequence by returning to the steady state position. Figure 4.9 shows the process of building a tilt sequence for an up then right

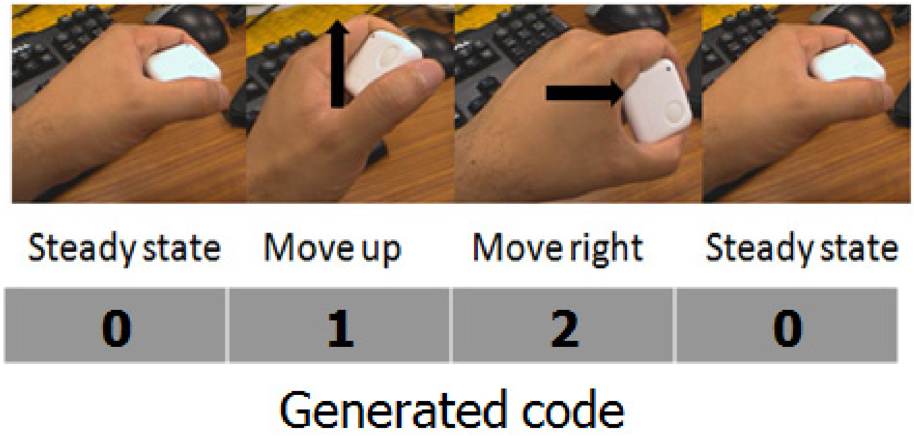


Figure 4.9: Tilt sequence, building a sequence of tilt movements up then right.

sequence. the user starts from steady state which will be mapped to number 0. Then he move to up direction which will be mapped to 1. He/she moves to the right direction and maps to number 3. Finally, he/she ends the gesture by returning back to steady state. The generated code for this sequence will be "0120" and the interface can use to map it to some command.

The interface will parse the sequence and trigger the functions. This mode is suitable for applications with many functions that depend on adjusting values or executing a sequence of commands. An example of applications that could use tilt sequence is editing photos and applying various filters. Furthermore, text editors that depend on a sequence or pattern of movements are adequate for this tilt method.

4.5.3 Mouse movements

Some applications require movement of the mouse to control a cursor or objects on the screen. In this case, direct mapping or tilt sequences are not sufficient. They are designed for triggering events to execute commands but cannot directly control mouse movements across the screen. Tilt gestures can have some feedback thought the control of the 2D (x, y) mouse coordinates across the display. There exist two methods to extract mouse motion from 3D accelerometer sensors. The first method is by calculating a displacement value relative to the angle of tilt among the 3D accelerometer axes. The second method depending on threshold values over the x and y axes, and then calculating a displacement value from static

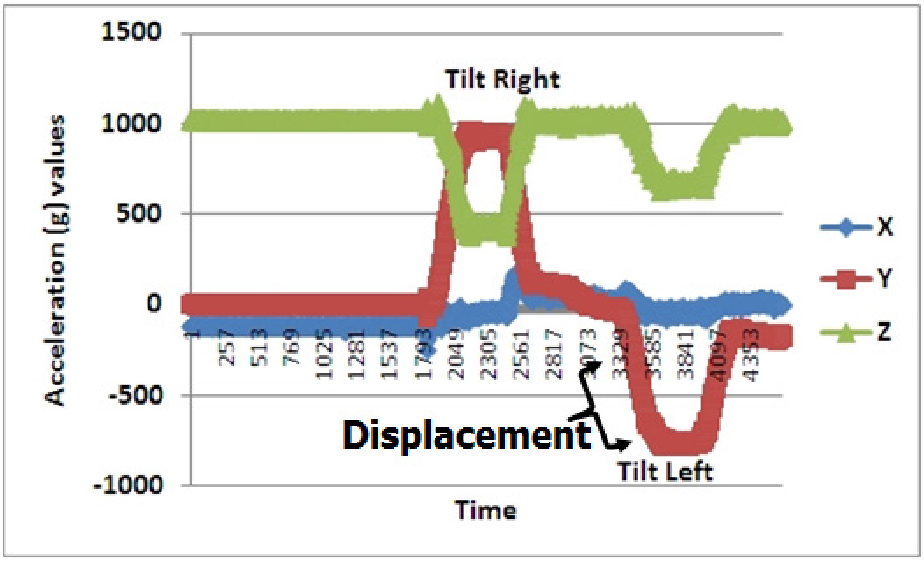


Figure 4.10: Displacement value relative to angle of tilt or threshold values.

acceleration. Figure 4.10 shows the displacement value that can be calculated relative to a threshold value or angle of tilt over the three axis.

In both methods, screen resolution is a factor for smooth movement of the mouse cursor over the display. To perform the mouse left clicks, right clicks using continuous gesture recognition, we use a "tap" or "shake" operation. This technique is best suited for applications that depend on mouse movements, like gaming programs such as flight simulators, car-racing games, and drawing applications.

4.6 Evaluation

The main purpose of the evaluation is to examine the usability of the interface while interacting with graphical user interface items like pie menu items. First experiment concerns measuring accuracy and time to select items. The second experiment compares different objects that can be attached to the sensor, so that we can find the effect of the attached object to the proposed methods. Subjects are 19 users between the age of 22 and 33 take part in the experiments. We evaluate usage of Cone, chopsticks, Sphere (ball) objects and projector mouse device.

Each subject has to finish three sessions per each object. Each session composed of 12 target selections. The first two targets are for training only. Targets are appearing on the

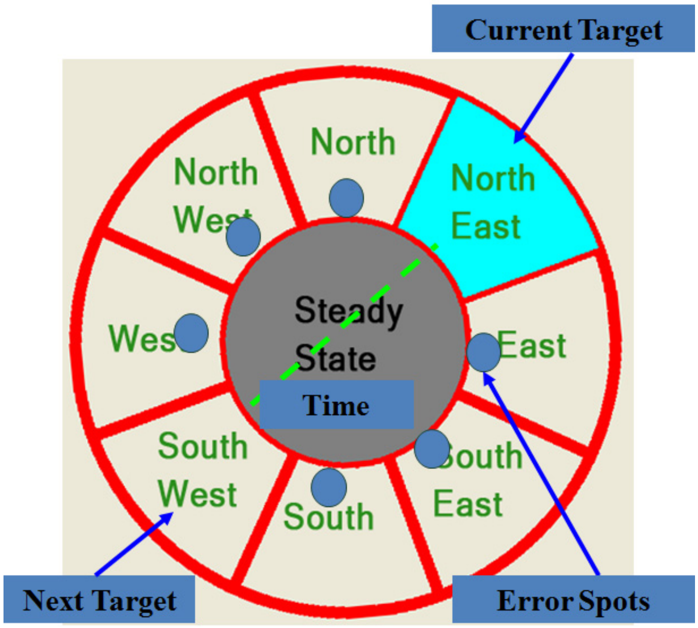


Figure 4.11: Menu interface showing basic 8 directions and targets.

pie menu randomly with marked color, and the user must tilt the sensor until hitting the target. Figure 4.11 shows the steady state, eight directions, current and next targets. The motion composed of two gestures: selecting direction and, moving towards then returning back to steady state in order to execute another command. If the user hits unmarked pie items, then this case will be counted an error.

The hand gestures were designed to be symmetric in shape across the main four directions labeled on figure 4.12 as {1,5,9,13}. We assumed that if all hand gestures were mirrored over x, y axis, then it will make subjects easy to memorize directions without the need to look for print out of gestures shapes. At the beginning of each session, we ask the subjects to make a practice session for hitting the targets using the proposed hand gestures.

4.6.1 Accuracy, speed and easy to practice

The results in Figure 4.13 show that most of the users can hit targets using a wireless mouse within 1 second. Errors found in using mouse was because using the thumb to control mouse makes user gesture extremely fast without giving eye focus on next targets. This makes the cursor frequently pass by error pie menu items. We think that this result was

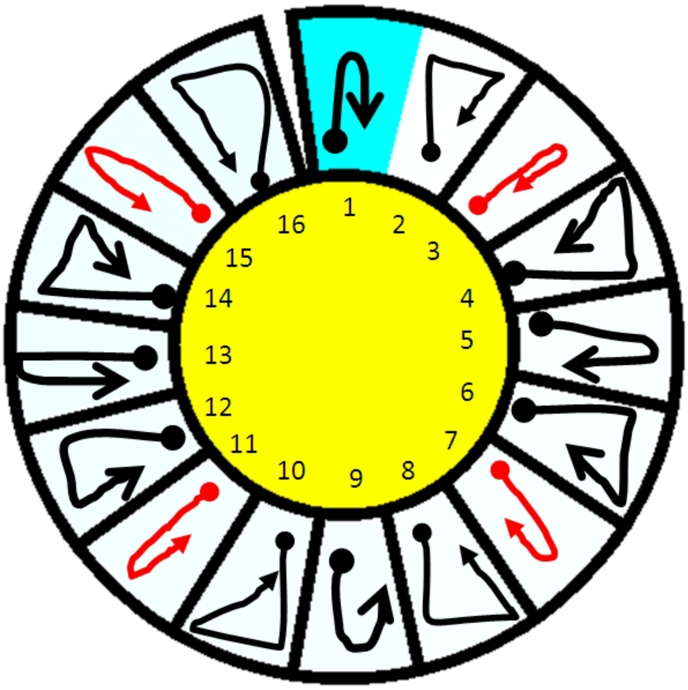
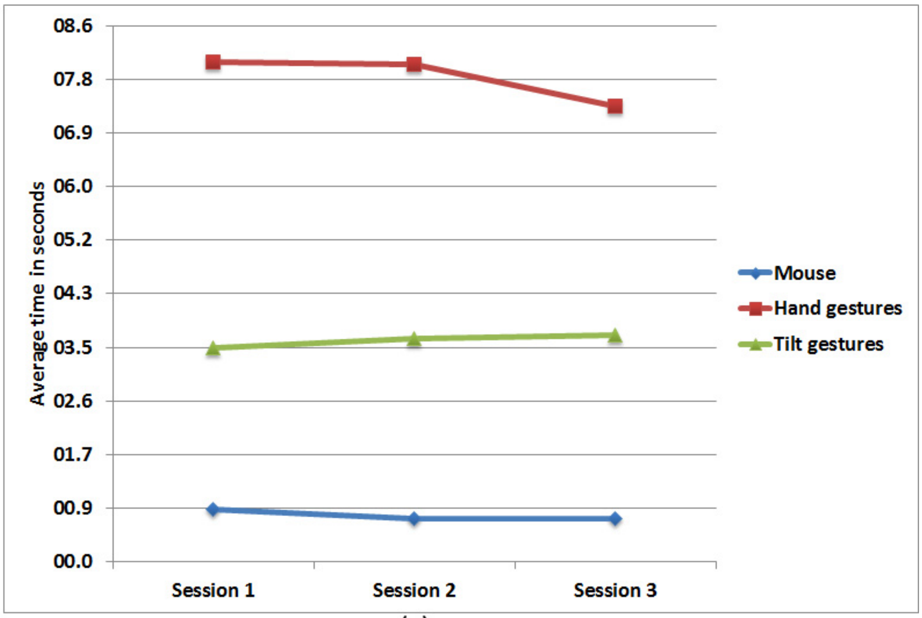


Figure 4.12: Hand gesture shapes for the 16 targets.

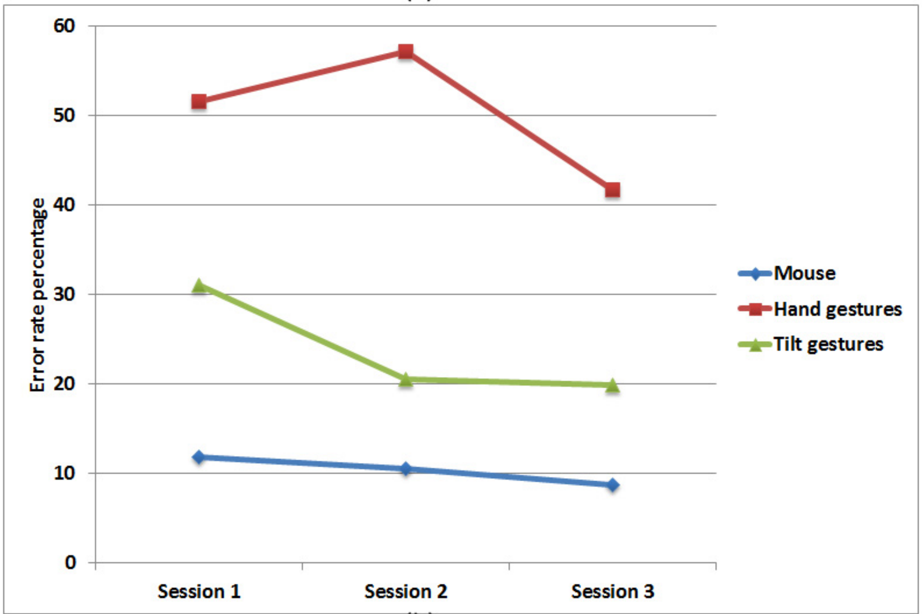
natural and coincide with the study of Berard et al. [113] as they found that still the 2D mouse is the most reliable device to control 2D and 3D applications.

The results indicate that the use of conventional interaction methods like the mouse is the fastest methods for interacting with pie menu items. However, the tilting interface has a high potential to be a convenient interface for user interaction in some situations. The average time over all three sessions to hit targets using the tilting interface was less than the one of predefined hand gestures. Furthermore, the tilting interface requires less space to be done as it depends only on wrist movement.

The predefined hand gestures consumes a lot of time to hit the targets because first the users takes time to understand the shape of each gesture. Second the users have to wait at least 1 to 2 seconds between each successive gesture, so they can return their hand to the starting position. Third the users complain from feeling tired and having hand tremor while using hand gestures. This leads them to make wrong hand gesture patterns. It was noticed that tilt gestures error rates are well close to mouse error rates. Tilt interface can have some promising results to improve the speed and accuracy of interaction. When users got trained to use the tilt interface in session 3, the numbers of errors reduced compared to



(a)



(b)

Figure 4.13: (a) Average time to hit targets per session, (b) Total number of errors

the first session. The learning curve for users could be affected when they trained for tilt gestures.

4.6.2 Easy to practice

The detailed comparison for each direction performance for the mouse, hand gestures, tilt gesture in means of speed to hit targets in seconds shown in Figure 4.14. We observed that users can use tilt hand gestures to enter the eight directions much faster than hand gestures. Moreover, the four basic directions (Down, Left, Up, Right) using tilting gestures are much faster than the “diagonal” tilt directions (Downleft, Upleft, Upright). However, the Downright direction speed was much faster, right-hand users can perform the tilt right and down gesture easily. Users say that the diagonal directions are sometimes confusing to do using predefined tilt gesture, and sometimes they confused about which direction exactly they should move their wrist. Some users try to make the move left full hand gesture as tilt to left. Users indicated that using the tilting interface was easy and natural to use and less painful compared to full hand gestures.

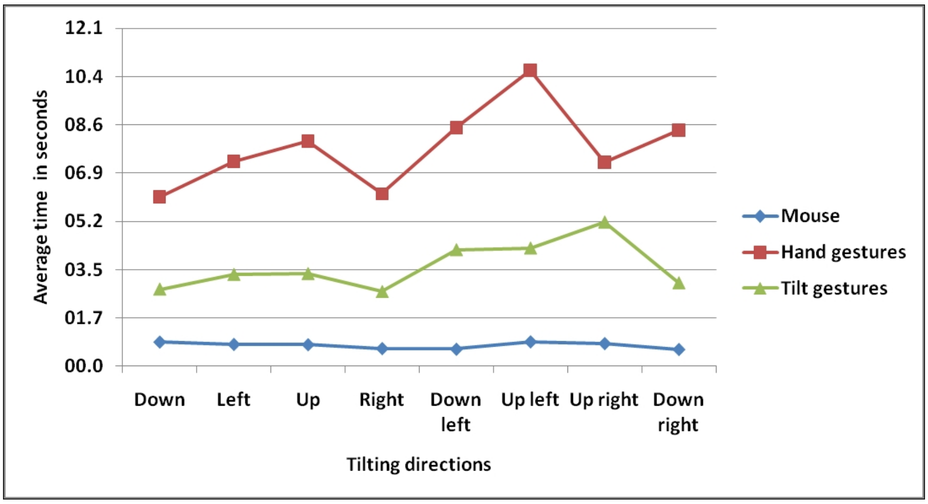


Figure 4.14: Comparing directions easy to practice.

The upleft, downleft directions using hand gestures was not easy for right-handed users to perform. This appears in Figure 4.14 as the peaks on hand gestures line. However, for the tilt gestures downright direction was better than other diagonal directions. We think that subjects being right/left handed has a direct impact on easiness to achieve gestures.

Figure 4.15: Objects used in the experiments (a) ball, (b) Cone and (c) chopsticks

4.6.3 Objects effect

We evaluate the use of 3 different shaped objects: ball, Cone, Chopsticks objects. Figurefig:TiltObjectsUsed shows objects used in the experiment. We selected those objects because of their individual geometrical shape and hand holding control. For each object, we compared predefined directions to interact with the developed pie menu with 8 directions. This combination between objects allow us to get feedback from users about what are most suitable objects. Figure 4.16 shows the average time to hit targets with corresponding to predefined tilt gestures.

The results demonstrate that ball and cone objects are suitable for predefined directions. Spherical and cone objects held by all fingers, so they are easy to expect their directions by users because of their symmetric shape (width and height almost identical). When we used the chopsticks as a controller, we observed more time to hit targets. It was observed that thin objects handled using the thumb and index finger only, allows the user to move the object in all the spatial space around their wrist. This leads to misinterpretation of tilting directions. Object tilt gestures pretended to work fast if the appropriate object was selected for interaction. We think that object shapes carried or held by all fingers are easy to twist by wrest, giving freedom for subjects to do well tilt gestures.

We studied the effect of objects on the tilting interface for subjects while they interact with the experimental menu. We ask the subjects to fill in a satisfaction survey for each method and objects used and give a rate on Likert scale from 1 to 5. We analyzed the satisfaction between objects, and we found that subjects were less satisfied for stick object. The ball and hand got the same satisfaction percentage for the 12 targets around 65%. Subjects say that they feel more comfortable while holding the ball and they feel easy to

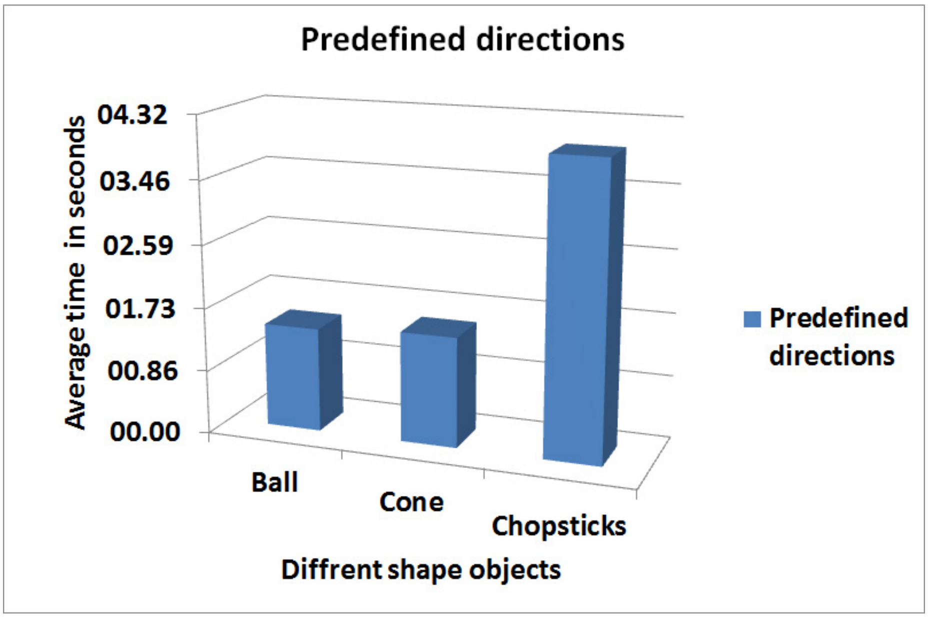


Figure 4.16: Objects average time to hit targets using predefined tilt gestures

control. The stick object satisfaction was far away from what we expected at 16 targets, as it was difficult to perform levels using thin objects with large spatial space around. Figure 4.17 shows satisfaction percentage for each object and the number of targets.

Users have some comments while using the proposed tilt gesture interface, the users have some problems in understanding how the predefined positions is performed. Sometimes users forget the steady state position. This situation leads to false gesture tilting detections. Moreover, users commented that using the chopsticks with predefined directions method and moving to Up and Down directions is confusing.

4.6.4 Number of tilts

To evaluate the usability of the tilting interface while interacting with graphical user interface items (GUI), we built a circle menu in which each piece virtually represents application functions or GUI items, such as buttons, icons and thumbnails. The circular menu can display up to 16 circular sequences of targets. Figure 4.18 displays the experimental circular menu in different sizes. The target item is shown on the menu by the highlighted light-blue color. Every time the subject hits a target successfully a new target appears in a different place. If the subject hits unmarked target items, this is counted as an error. Time

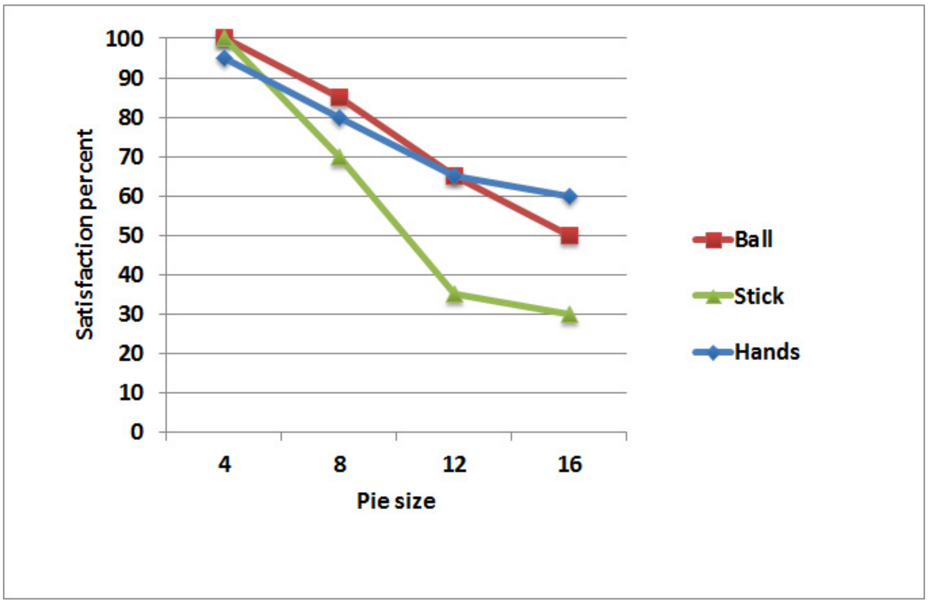


Figure 4.17: Subjects satisfaction percent per number of targets and objects.

is measured as the time in seconds needed to move from the steady state position to hit the current target. Subjects were divided into two groups to test the learning effect. Each group was assigned to do the task in a given sequence.

First group was asked to interact with a number of targets in an ascending order 4, 8, 12 and 16 targets. Second group was asked to interact in descending order 16, 12, 8, and 4. At the end of the experiments, we make a short interview with the subjects get their feedback, and we asked them to fill a satisfaction survey. The experiment concerns the measurement of accuracy, speed, learning of directions to practice and the appropriate number of tilt directions to be recognized. Each subject has to perform two sessions per each interaction method. Each session composed of 4,8,12,16 targets selections, a total of 40 targets per session. At the beginning of each session, we ask the subjects to make a practice session for hitting the targets using the three methods.

The results indicate that the use of conventional interaction methods such as the mouse is the fastest methods for interacting with the experimental application. However, the results founded that tilt gestures have a stable speed regardless the number of targets. The hand gestures were the lowest in speed and showed a linear increase of speed with respect to the number of targets. Figure 4.19 (a) shows the speed comparison between mouse, hand gesture and tilt interface. The accuracy to hit targets while using the tilt interface was

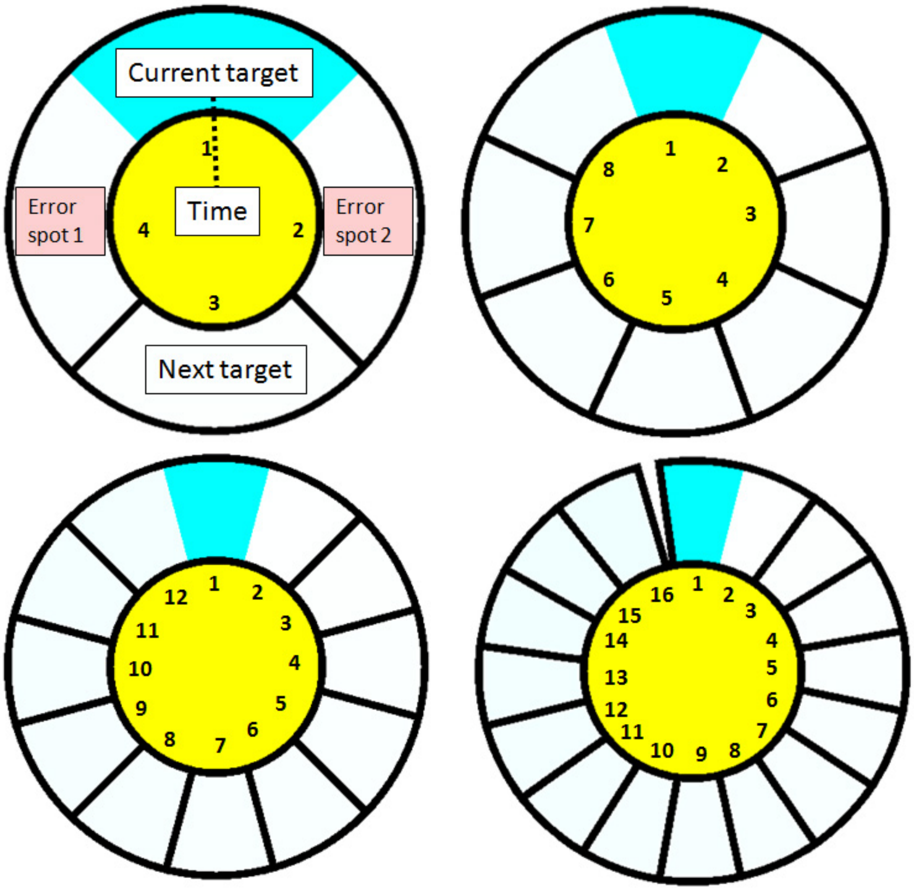


Figure 4.18: Developed experimental application with different target numbers and sizes.

close to mouse for the 4, 8 targets, but the results show a linear decrease of accuracy when the number of targets increased to 12, 16. Figure 4.19 (b) shows the accuracy comparison between mouse, hand gestures and tilt interface.

The accuracy for hand gestures was less than tilt and mouse, but the results showed that for the number of targets of 8, 12, 16 the accuracy was stable on average of 75%. However, tilt gesture accuracy for 16 targets was still more accurate than hand gestures. The tilt interface took more time than that of the mouse. Subjects sometimes either forget to return to steady state after hitting the target, or they missed the position of steady state. The declination of accuracy has two reasons, first was the difficulty to imagine the direction levels for each direction without having a direct visual feedback from the interface. Second, subjects find it hard to make 12 gestures or more using wrist movement only and direct mapping. The hand gestures consumed a lot of time to hit the targets and accuracy was the minimum. The reason behind this is that first, the subjects took time to understand the shape of each gesture especially gestures done by subjects towards the down directions. Second, the subjects felt tired and complained from hand tremor while using hand gestures, leading them to make wrong hand gesture patterns.

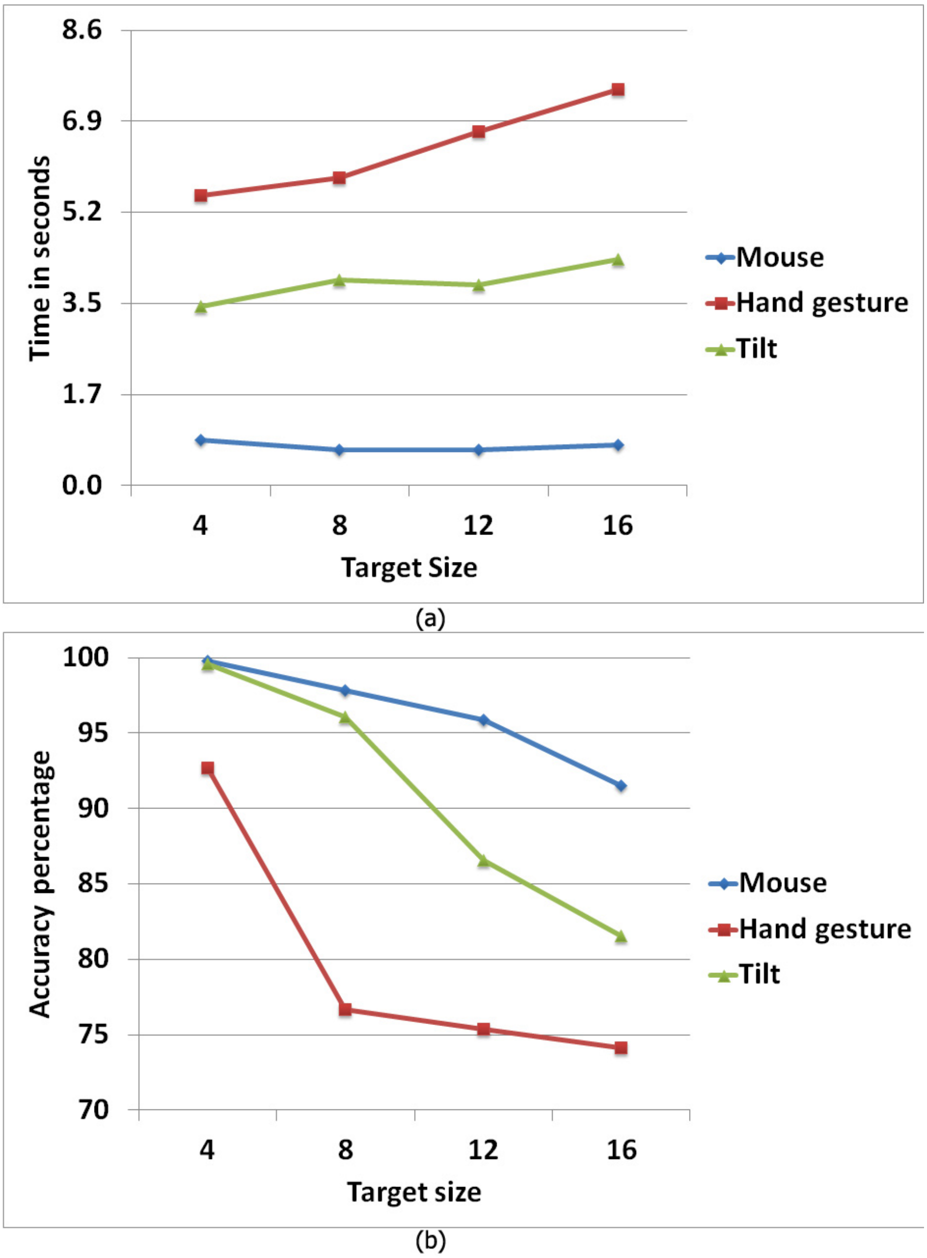


Figure 4.19: Comparison between mouse, hand gestures and tilt interface (a) speed (b) accuracy.

We have done further analysis for the speed comparison between the devices with respect to learning effect for Group1 4, 8, 12, 16 and Group2 16, 12, 8, 4. The results founded that there is a difference of 1 second between groups for the hand gestures. However, the learning effect for tilt gestures remains constant for both Group1 and Group2. Stability in learning effect is an advantage of tilt gestures over hand gestures as subjects can easily memorize direction of tilts. When subjects used the hand gestures they consume less learning time when they start from the large amount of targets (16) towards (4) targets as they learn most of the hand gestures in the session of 16 targets.

4.7 Discussions

The subjects have given feedback during an interview about the use of tilting interface after finishing the experiments. Subjects make the move left full hand gesture as tilt to left. Subjects indicate that using the tilting interface was easy and natural to use and less painful in comparison to full hand gestures. Subjects cannot imagine how to tilt the sensor in 3D space and the mapping towards the experimental application in 2D directions. All subjects agree that the mouse was the most comfortable tool for interaction, but it lacks remote interaction and should have a stable hand position which is unsuitable for on the move interaction. Some subjects confirm that using objects around as balls, pens, and headphones could enrich their interaction with devices around in the ubiquitous computing environment.

4.8 Summary

Tilting gestures were studied as an interaction method in ubiquitous environments where it is difficult to have ordinary input devices. Tilt gestures were combined with objects to control applications and devices. In this research, we proposed a system that captures people context and provides them tilt gestures for objects they interact with. We discuss the system settings, gesture recognition and methods for mapping directions using a 3D accelerometer to capture the tilting gestures. We explored application fields and areas were tilting gestures could be useful by applying it to map and web browsers, media players, PTZ camera protector and special Japanese text tool. The usefulness and feedbacks of the tilt gesture interface were evaluated by studding it with three different types of remote interaction applications (computer presentation viewer, photo browsers, and entertainment games). We present the mapping techniques for each application commands and tilt gestures based on objects they use for interaction and some users feedback. We addressed the classification of suitable applications for tilting interfaces as direct mapping, tilt sequence and mouse moves.

An experiment to evaluate the interaction with a menu representing GUI items was conducted. The results show that tilt interface was close to mouse in accuracy and speed for up to 8 targets. Even when the number of targets increased up to 16 targets tilt gestures still perform much better than hand gestures. Also, we found that tilt gestures were easy to memorize by most of the users, whether they started by 4 or 16 targets there was stability in results. Furthermore, a group of users feedback showed the usage of tilt gestures and objects around have a lot of possibilities to enrich their interaction in ubiquitous environments.

Chapter 5

Conclusion

One of the main challenges of interaction in a ubiquitous environment is the use of hand gestures for interacting with day to day applications. This interaction may be negatively affected due to the change in the user's position, interaction device, or the level of social acceptance of a specific hand gesture. We presented a system called "UbiGesture" to provide users with enough appropriate customized hand gesture profiles for any given situation. UbiGesture packs user context and his gestures, save it to the network, then whenever the user changes his context it will download appropriate profile. In this research, we used Wii remote and WAA006 sensors to capture hand gestures done by users. We studied whether users are going to change their gesture shapes behavior according to their context environment. Furthermore, we studied the effect of imposing position and social parameters on users interacting with predefined hand gestures.

Sticker can be added to some object to give new functions and attributes, we think that if gesture profiles attached to objects they can give them new functions and meanings, so we call it "Smart Gesture Sticker (SGS)". We allow user's to use any object around in different ubiquitous contexts in an intuitive way. A proposed a system that makes object gestures portable and customizable with users was presented. We extend UbiGesture architecture so that it can assist users to interact and customize gesture profiles for objects existing in their ubiquitous environment. The detailed architecture of SGS was presented and a system implemented SGS in ubiquitous environments with different applications is presented. Gestures of users have been captured with a small coin-size wireless attachable sensor. Object gestures were recognized continuously through the capturing of the starts and ends of each gesture with a pause interval. SGS tracks the user's context in the ubiquitous environment. When the user enters some location, the system will provide him with his/her appropriate gesture profiles, so he can interact or adjust his gestures. A typical gesture profile composed of profile ID, location name, device, application, object and gestures attached to this object. Later on, when people change their location, SGS will search it's repository to get them the most appropriate profile for their context.

We conducted experiments concerning object gestures then we studied if the SGS suggested profiles were efficient compared to adjusted profiles. The first observation we found was that some objects like the cellular phone and book have positively affected the interactions compared to other objects like leg and pen. In the second part of the experiments, the user evaluates the profile created by SGS without any adjustments for gestures twice. After that, he/she is asked to adjust some of the gestures that one did not like then evaluate the profile again three times. The results show that the more the application has functions as the case of the photo viewer, adjusting profiles was necessary to enhance the speed of interaction. In other scenarios, we observed that the selected SGS profile speed remains constant even after the user adjusted some of the gesture shapes. The movement of the head has a limited spatial for doing gestures, so users depend on tilting and tapping gestures to perform their gestures. We found that object gestures are faster and more accurate than hand gestures, if and only if the appropriate objects were selected.

Our main motivation to enhance the interaction between humans and their environment through the usage of more faster and robust gestures such as the "Tilting gestures". We developed a user interface that use 3D accelerometer as an input device. We describe the interface setting, method of extracting motion and direction from 3D accelerometer, using the tilting gesture. We proposed a method that allows users to define their own tilting positions and assign it to directions. Then we describe a menu selection interface based on pie menu for interaction with remote displays. The system receives wireless sensor data, pre-process, recognize and translate it into commands. The sensor is small enough to be attached to everyday objects like pen, sphere and cellular phone. When object has no tilting directions or in the initial state it is called in a steady state. When the users tilt the objects over the x, y, z axis, the gravity value will be applied on each axis respectively. The system captures tilting reading from the accelerometer axis and compares it to threshold values to recognize the directions. A typical tilt movement is composed of starting from steady state, moving to direction and return to steady state.

We explored application fields and areas were tilting gestures could be useful by applying it to map and web browsers, media players, PTZ camera protector and special Japanese text tool. The usefulness and feedbacks of the tilt gesture interface were evaluated by studding it with three different types of remote interaction applications (computer presentation viewer, photo browsers, and entertainment games). Furthermore, we addressed the classification of suitable applications for tilting interfaces as direct mapping, mouse movement and tilt sequence. First, direct mapping was based on one tilt direction mapped to one command, it is suitable for up to 8 directions and commands. Applications in this category can depend on reverse operations like presentation and photo viewers, media players and remote controlling devices. Second, was mouse movements which depend on controlling the mouse cursor on a display based on the angle of tilt. This category is suitable for gaming programs and drawing applications. Third, was tilting sequence or codes, if we need to control more than 8 commands in the application. Users should enter two or more successive tilt gestures and the interface will parse the sequence and convert it into a command. This category is suitable for editing photos and applying filters and special text entry tools.

We conducted an experiment to compare three different interaction methods for interaction with a circular menu: mouse device, tilting interface and predefined hand gestures. The results show that the usage of conventional interaction methods like the mouse is the fastest methods for interacting with pie menu items. However, the tilting interface has a high potential to be a convenient interface for user interaction in some situations. The average time over for the three sessions to hit targets using the tilting interface was less than the one of the predefined hand gestures. The reason for this is that the tilting interface requires less space to be done and depends only on wrist movement.

Another experiment was conducted to evaluate the interaction with a menu representing GUI items in different size. The results show that when the number of targets increased to 16 targets, tilt gestures still performed in stable speed regardless the number of targets. The results for accuracy even was far better than hand gestures. Furthermore, we found that most of the users found the tilt gestures easy to memorize. The results were comparable between the two groups that started with four or with 16 targets preserving stability in learning tilt gestures. Whether they start by 4 or 16 targets, there was stability in results. Furthermore, some user feedback indicated a consensus on how using everyday objects like balls, pens or headphones enriched their interaction with applications. We also found that accuracy for tilt gestures was similar to that of mouse gestures for up to 8 directions.

5.1 Summary

We believe that interaction in ubiquitous environments could sustain from users changing their context, leading us to present gestures profiles in ubiquitous environments as one of the ways that can enhance interaction between people and their environments. We studied the effect of context in ubiquitous environments and gesture shapes. We found that position and social parameters affect directly hand gesture interaction. Augmenting objects with sensors could lead to new smart applications and appliances. Selecting an appropriate object can enhance time and accuracy of interacting with the environment. Tilt gestures bring a lot of interesting enhancement to gesture interactions in means of small, fast and easy to memorize directions. The accuracy of tilt gesture interface while interacting with GUI can be increased by using appropriate objects. We studied the effect of object shape while interacting with different interface methods. We explored and classified tilt gestures application fields. We think that Hand, object and tilt gestures have a lot possibilities to enrich future interaction in ubiquitous environments.

5.2 Future directions

We would like to consider more parameters of interaction with hand gestures in ubiquitous environments like users activity, fitness and cultural background in more details. We want to investigate more objects effect for creating smart appliances in the user environment like dishes in restaurants, keys, frying pots in the kitchen and doors. Such object most people interacting with can be used to build a practical everyday appliances. We need to conduct further study about different parameters that can affect the selection of appropriate SGS object gesture profile like the wight of objects. Furthermore, we need to study other types of tilt gestures mapping techniques based on detailed long term usage of the system. We want to evaluate the usage of tilt levels as a method to increase the number of commands to be executed by tilt gestures.

Bibliography

[1] M. Weiser and J. S. Brown, *The coming age of calm technolgy.* New York, NY, USA: Copernicus, 1997, pp. 75–85.

[2] M. Weiser, R. Gold, and J. S. Brown, "The origins of ubiquitous computing research at parc in the late 1980s," *IBM Syst. J.*, vol. 38, pp. 693–696, December 1999.

[3] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, ser. HUC '99. London, UK: Springer-Verlag, 1999, pp. 304–307.

[4] C.-W. Kuo, J. K. Chiang, and Q.-P. Lin, "The research on applying rfid information system architecture for museum service," *Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, ser. ICCIT '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 218–222.

[5] C.-Y. Tsai, S.-Y. Chou, and S.-W. Lin, "Location-aware tour guide systems in museum," *Collaborative Product and Service Life Cycle Management for a Sustainable World*, ser. Advanced Concurrent Engineering, R. Curran, S.-Y. Chou, and A. Trappey, Eds. Springer London, 2008, pp. 349–356.

[6] W.-S. Yang, H.-C. Cheng, and J.-B. Dia, "A location-aware recommender system for mobile shopping environments," *Expert Syst. Appl.*, vol. 34, pp. 437–445, January 2008.

[7] M. Keidl and A. Kemper, "Towards context-aware adaptable web services," *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters.* New York, NY, USA: ACM, 2004, pp. 55–65.

[8] C.-F. Lai, Y.-M. Huang, and H.-C. Chao, “A context-aware multi-model remote controller for electronic home devices,” *J. Supercomput.*, vol. 54, pp. 43–60, October 2010.

[9] D. Jea, I. Yap, and M. B. Srivastava, “Context-aware access to public shared devices,” *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, ser. HealthNet ’07. New York, NY, USA: ACM, 2007, pp. 13–18.

[10] A. Artail and M. Raydan, “Device-aware desktop web page transformation for rendering on handhelds,” *Personal Ubiquitous Comput.*, vol. 9, pp. 368–380, November 2005.

[11] B. Schilit, N. Adams, and R. Want, “Context-aware computing applications,” *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications.* Washington, DC, USA: IEEE Computer Society, 1994, pp. 85–90.

[12] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context aware systems,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, pp. 263–277, June 2007.

[13] C. K. Hess and R. H. Campbell, “An application of a context-aware file system,” *Personal Ubiquitous Comput.*, vol. 7, pp. 339–352, December 2003.

[14] O. Conlan, R. Power, S. Higel, D. OSullivan, and K. Barrett, “Next generation context aware adaptive services,” *ISICT ’03: Proceedings of the 1st international symposium on Information and communication technologies.* Trinity College Dublin, 2003, pp. 205–212.

[15] A. K. Dey and G. D. Abowd, “Cybreminder: A context-aware system for supporting reminders,” *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, ser. HUC ’00. London, UK: Springer-Verlag, 2000, pp. 172–186.

[16] J. Budzik and K. J. Hammond, “User interactions with everyday applications as context for just-in-time information access,” *Proceedings of the 5th international conference on Intelligent user interfaces*, ser. IUI ’00. New York, NY, USA: ACM, 2000, pp. 44–51.

[17] J. McGrenere, R. M. Baecker, and K. S. Booth, "A field evaluation of an adaptable two-interface design for feature-rich software," *ACM Trans. Comput.-Hum. Interact.*, vol. 14, no. 1, p. 3, 2007.

[18] S. Baraldi, L. Benini, O. Cafini, A. Del Bimbo, E. Farella, G. Gelmini, L. Landucci, A. Pieracci, and N. Torpei, "Evolving tuis with smart objects for multi-context interaction," *CHI '08 extended abstracts on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 2955–2960.

[19] J. E. Bardram, "Applications of context-aware computing in hospital work: examples and design principles," *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2004, pp. 1574–1579.

[20] A. Zimmermann, M. Specht, and A. Lorenz, "Personalization and context management," *User Modeling and User-Adapted Interaction*, vol. 15, pp. 275–302, August 2005.

[21] P. M. Juan-Carlos Cano and C.-K. TohDOI, "Ubiqmuseum: A bluetooth and java based context-aware system for ubiquitous computing," *Wireless Personal Communications*, vol. 38, no. 2, pp. 187–202, 2006.

[22] Z. Idrus, S. Z. Z. Abidin, R. Hashim, and N. Omar, "Social awareness: the power of digital elements in collaborative environment," *W. Trans. on Comp.*, vol. 9, pp. 644–653, June 2010.

[23] A. Oulasvirta and J. Blom, "Motivations in personalisation behaviour," *Interact. Comput.*, vol. 20, pp. 1–16, January 2008.

[24] G. Kappel, B. Proll, W. Retschitzegger, and W. Schwinger, "Customisation for ubiquitous web applications a comparison of approaches," *Int. J. Web Eng. Technol.*, vol. 1, pp. 79–111, August 2003.

[25] S. Kim, S. Takahashi, and J. Tanaka, "New interface using palm and fingertip without marker for ubiquitous environment," *Computer and Information Science, ACIS International Conference on computer and information science*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 819–824.

[26] T. Nakamura, S. Takahashi, and J. Tanaka, "One-finger interaction for ubiquitous environment," *Computer and Information Science, ACIS International Conference on computer and information science.* Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 267–272.

[27] S. T. Ho Wong and J. Tanaka, "Tangible linker: organizing and accessing digital contents using untagged tangible objects," *Proceedings of 9th Asia-Pacific Conference on Computer-Human Interaction (APCHI2010)*, 2010, pp. 693–698.

[28] J. Higuchi, S. Takahashi, and J. Tanaka, "Comecam: A communication support system between both ends of the live camera connection," *Proceedings of the Third International Conference on Collaboration Technologies (CollabTech2007)*, 2007, pp. 33–38.

[29] S. Nagao, S. Takahashi, and J. Tanaka, "Mirror appliance: Recommendation of clothes coordination in daily life," *Human Factors in Telecommunication (HFT2008)*, 2008, pp. 367–374.

[30] M. Naito, B. Shizuki, J. Tanaka, and H. Hosobe, "Interaction techniques using a spherical cursor for 3d targets acquisition and indicating in volumetric displays," *Proceedings of the 2009 13th International Conference Information Visualisation.* Washington, DC, USA: IEEE Computer Society, 2009, pp. 607–612.

[31] "American water ski association," http://www.usawaterski.org/.

[32] D. Surie, F. Lagriffoul, T. Pederson, and D. Sjolie, "Activity recognition based on intra and extra manipulation of everyday objects," *Proceedings of the 4th international conference on Ubiquitous computing systems*, ser. UCS'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 196–210.

[33] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recogn. Lett.*, vol. 29, pp. 2213–2220, December 2008.

[34] T. Baudel and M. Beaudouin-Lafon, "Charade: remote control of objects using freehand gestures," *Commun. ACM*, vol. 36, no. 7, pp. 28–35, 1993.

[35] S. Keates and P. Robinson, “The use of gestures in multimodal input,” *Assets ’98: Proceedings of the third international ACM conference on Assistive technologies.* New York, NY, USA: ACM, 1998, pp. 35–42.

[36] M. Kurze, “Personalization in multimodal interfaces,” *TMR ’07: Proceedings of the 2007 workshop on Tagging, mining and retrieval of human related activity information.* New York, NY, USA: ACM, 2007, pp. 23–26.

[37] F. Kawsar and T. Nakajima, “Persona: a portable tool for augmenting proactive applications with multimodal personalization support,” *MUM ’07: Proceedings of the 6th international conference on Mobile and ubiquitous multimedia.* New York, NY, USA: ACM, 2007, pp. 160–168.

[38] T. Nakajima, “How to reuse existing interactive applications in ubiquitous computing environments?” *SAC ’06: Proceedings of the 2006 ACM symposium on Applied computing.* New York, NY, USA: ACM, 2006, pp. 1127–1133.

[39] S. Ronkainen, J. J. Hakkila, S. Kaleva, A. Colley, and J. Linjama, “Tap input as an embedded interaction method for mobile devices,” *TEI ’07: Proceedings of the 1st international conference on Tangible and embedded interaction.* New York, NY, USA: ACM, 2007, pp. 263–270.

[40] J. Rico and S. Brewster, “Gestures all around us: user differences in social acceptability perceptions of gesture based interfaces,” *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’09. New York, NY, USA: ACM, 2009, pp. 64:1–64:2.

[41] “Sunspot,” http://www.sunspotworld.com/.

[42] “Ubisense,” http://www.ubisense.net/.

[43] “Wii remote,” http://wii.nintendo.com.

[44] “Art promotions waa-006,” http://www.atr-p.com/sensor06.html.

[45] J. Kela, P. Korpipaa, J. Mantyjarvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, “Accelerometer-based gesture control for a design environment,” *Personal Ubiquitous Comput.*, vol. 10, no. 5, pp. 285–299, 2006.

[46] Z. Prekopcsák, P. Halácsy, and C. Gáspár-Papanek, "Design and development of an everyday hand gesture interface," *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, ser. MobileHCI '08. New York, NY, USA: ACM, 2008, pp. 479–480.

[47] R. Dachselt and R. Buchholz, "Natural throw and tilt interaction between mobile phones and distant displays," *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 3253–3258.

[48] J.-M. Sanchez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Bluemall: a bluetooth-based advertisement system for commercial areas," *PM2HW2N '08: Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks.* New York, NY, USA: ACM, 2008, pp. 17–22.

[49] S. Sreedharan, E. S. Zurita, and B. Plimmer, "3d input for 3d worlds," *OZCHI '07: Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments.* New York, NY, USA: ACM, 2007, pp. 227–230.

[50] L. Gallo, G. D. Pietro, and I. Marra, "3d interaction with volumetric medical data: experiencing the wiimote," *Ambi-Sys '08: Proceedings of the 1st international conference on Ambient media and systems.* ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–6.

[51] T. Schlomer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction.* New York, NY, USA: ACM, 2008, pp. 11–14.

[52] A. Shirai, E. Geslin, and S. Richir, "Wiimedia: motion analysis methods and applications using a consumer video game controller," *Sandbox '07: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games.* New York, NY, USA: ACM, 2007, pp. 133–140.

[53] B. Bruegge, C. Teschner, P. Lachenmaier, E. Fenzl, D. Schmidt, and S. Bierbaum, "Pinocchio: conducting a virtual symphony orchestra," *ACE '07: Proceedings of the*

international conference on Advances in computer entertainment technology. New York, NY, USA: ACM, 2007, pp. 294–295.

[54] H.-J. Lee, H. Kim, G. Gupta, and A. Mazalek, "Wiiarts: creating collaborative art experience with wiiremote interaction," *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction.* New York, NY, USA: ACM, 2008, pp. 33–36.

[55] A. J. H. Peddemors, M. M. Lankhorst, and J. de Heer, "Presence, location, and instant messaging in a context-aware application framework," *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management.* London, UK: Springer-Verlag, 2003, pp. 325–330.

[56] J. Mantyjarvi, J. Kela, P. Korpipaa, and S. Kallio, "Enabling fast and effortless customisation in accelerometer based gesture interaction," *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, ser. MUM '04. New York, NY, USA: ACM, 2004, pp. 25–31.

[57] P. W. Jang, I.J., "Signal processing of the accelerometer for gesture awareness on hand held devices," *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop.* Dept. of Information and Commun., Kyungpook Nat. Univ., South Korea;: IEEE, 2003, pp. 139– 144.

[58] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[59] A. Coyette, S. Schimke, J. Vanderdonckt, and C. Vielhauer, "Trainable sketch recognizer for graphical user interface design," *Human-Computer Interaction INTERACT 2007*, ser. Lecture Notes in Computer Science, C. Baranauskas, P. Palanque, J. Abascal, and S. Barbosa, Eds. Springer Berlin / Heidelberg, 2007, vol. 4662, pp. 124–135.

[60] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168–173, 1974.

[61] X. Cao and R. Balakrishnan, "Evaluation of an on-line adaptive gesture interface with command prediction," *GI '05: Proceedings of Graphics Interface 2005.* School

of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2005, pp. 187–194.

[62] S. T. Ayman Atia and J. Tanaka, "Smart gesture sticker: Smart hand gestures profiles for daily objects interaction," *Computer and Information Science, ACIS International Conference on*, vol. 0, pp. 482–487, 2010.

[63] A. Atia, S. Takahashi, and J. Tanaka, "Coin size wireless sensor interface for interaction with remote displays," *The 12th International Conference on Human-Computer Interaction (HCI International 2007*, 2007, pp. 733–742.

[64] E. Farella, O. Cafini, L. Benini, and B. Ricco, "A smart wireless glove for gesture interaction," *SIGGRAPH '08: ACM SIGGRAPH 2008 posters.* New York, NY, USA: ACM, 2008, pp. 1–1.

[65] *Activity recognition from accelerometer data.* AAAI Press, 2005.

[66] I.-J. Kim, S. C. Ahn, H. Ko, and H.-G. Kim, "Automatic lifelog media annotation based on heterogeneous sensor fusion," *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008.*, 2008, pp. 703 –708.

[67] M. Blum, A. S. Pentland, and G. Troster, "Insense: Interest-based life logging," *IEEE Multimedia*, vol. 13, pp. 40–48, 2006.

[68] Y. Cho, Y. Nam, Y.-J. Choi, and W.-D. Cho, "Smartbuckle: human activity recognition using a 3-axis accelerometer and a wearable camera," *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, ser. HealthNet '08. New York, NY, USA: ACM, 2008, pp. 7:1–7:3.

[69] Y. Shi and T. Tsui, "An fpga-based smart camera for gesture recognition in hci applications," *Proceedings of the 8th Asian conference on Computer vision - Volume Part I*, ser. ACCV'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 718–727.

[70] M. Pouke, S. Hickey, T. Kuroda, and H. Noma, "Activity recognition of the elderly," *Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems*, ser. CASEMANS '10. New York, NY, USA: ACM, 2010, pp. 7:46–7:52.

[71] I. Mavrommati and A. Kameas, "The evolution of objects into hyper-objects: will it be mostly harmless?" *Personal Ubiquitous Comput.*, vol. 7, no. 3-4, pp. 176–181, 2003.

[72] A. Kameas, S. Bellis, I. Mavrommati, K. Delaney, M. Colley, and A. Pounds-Cornish, "An architecture that treats everyday objects as communicating tangible components," *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications.* Washington, DC, USA: IEEE Computer Society, 2003, pp. 115–x2.

[73] F. Kawsar, K. Fujinami, and T. Nakajima, "Augmenting everyday life with sentient artefacts," *sOc-EUSAI '05: Proceedings of the 2005 joint conference on smart objects and ambient intelligence.* New York, NY, USA: ACM, 2005, pp. 141–146.

[74] P.-Y. Chi, J.-H. Chen, S.-Y. Liu, and H.-H. Chu, "Designing smart living objects: enhancing vs. distracting traditional human-object interaction," *Proceedings of the 12th international conference on Human-computer interaction: interaction platforms and techniques*, ser. HCI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 788–797.

[75] E. Farella, O. Cafini, L. Benini, and B. Ricco, "A smart wireless glove for gesture interaction," *SIGGRAPH '08: ACM SIGGRAPH 2008 posters.* New York, NY, USA: ACM, 2008, pp. 1–1.

[76] X. Cao and R. Balakrishnan, "Visionwand: interaction techniques for large displays using a passive wand tracked in 3d," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 729–729, 2004.

[77] A. Schmidt, M. Beigl, and H. w. Gellersen, "There is more to context than location," *Computers and Graphics*, vol. 23, pp. 893–901, 1998.

[78] T. Strang and C. Linnhoff-Popien, "A context modeling survey," *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[79] M. Sutterer, O. Droegehorn, and K. David, "Making a case for situation-dependent user profiles in context-aware environments," *MNCNA '07: Proceedings of the 2007*

Workshop on Middleware for next-generation converged networks and applications. New York, NY, USA: ACM, 2007, pp. 1–6.

[80] S. Michael, D. Olaf, and D. Klaus, "User profile selection by means of ontology reasoning," *AICT '08: Proceedings of the 2008 Fourth Advanced International Conference on Telecommunications.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 299–304.

[81] K.-Y. Cheng, R.-H. Liang, B.-Y. Chen, R.-H. Laing, and S.-Y. Kuo, "icon: utilizing everyday objects as additional, auxiliary and instant tabletop controllers," *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1155–1164.

[82] H. Witt and H. Kenn, "Towards implicit interaction by using wearable interaction device sensors for more than one task," *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems.* New York, NY, USA: ACM, 2006, p. 20.

[83] D. Merrill and P. Maes, "Augmenting looking, pointing and reaching gestures to enhance the searching and browsing of physical objects," *Proceedings of the 5th international conference on Pervasive computing*, ser. PERVASIVE'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–18.

[84] A. Atia, S. Takahashi, K. Misue, and J. Tanaka, "Ubigesture: Customizing and profiling hand gestures in ubiquitous environment," *Human-Computer Interaction. Novel Interaction Methods and Techniques,13th International Conference, HCI International 2009*, 2009, pp. 141–151.

[85] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, pp. 1–9, 2009.

[86] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, *Gesture Recognition with a 3-D Accelerometer*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5585, pp. 25–38.

[87] T. Inomata, F. Naya, N. Kuwahara, F. Hattori, and K. Kogure, "Activity recognition from interactions with objects using dynamic bayesian network," *Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, ser. Casemans '09. New York, NY, USA: ACM, 2009, pp. 39–42.

[88] P. Ljungstrand, J. Redstrom, and L. E. Holmquist, "Webstickers: using physical tokens to access, manage and share bookmarks to the web," *Proceedings of DARE 2000 on Designing augmented reality environments*, ser. DARE '00. New York, NY, USA: ACM, 2000, pp. 23–31.

[89] A. Bernheim Brush, T. Combs Turner, M. A. Smith, and N. Gupta, "Scanning objects in the wild: Assessing an object triggered information system," *UbiComp 2005: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, M. Beigl, S. Intille, J. Rekimoto, and H. Tokuda, Eds. Springer Berlin / Heidelberg, 2005, vol. 3660, pp. 305–322.

[90] M. Rehm, N. Bee, and E. Andre, "Wave like an egyptian: accelerometer based gesture recognition for culture specific interactions," *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, ser. BCS-HCI '08. Swinton, UK, UK: British Computer Society, 2008, pp. 13–22.

[91] D. Kim, D. Kim, and D. Kim, "An intelligent smart home control using body gestures," *ICHIT '06: Proceedings of the 2006 International Conference on Hybrid Information Technology.* Washington, DC, USA: IEEE Computer Society, 2006, pp. 439–446.

[92] E. ONeill, M. Kaenampornpan, V. Kostakos, A. Warr, and D. Woodgate, "Can we do without guis? gesture and speech interaction with a patient information system," *Personal Ubiquitous Comput.*, vol. 10, no. 5, pp. 269–283, 2006.

[93] R. Slyper and J. K. Hodgins, "Action capture with accelerometers," *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 193–199.

[94] *ReachMedia: On-the-move interaction with everyday objects.* Washington, DC, USA: IEEE Computer Society, 2005.

[95] A. Atia and J. Tanaka, "Interaction with tilting gestures in ubiquitous environments," *International Journal of UbiComp*, vol. 1, pp. 1–13, 2010.

[96] *Explorations in sound for tilting-based interfaces.* New York, NY, USA: ACM, 2006.

[97] N. Sawasaki, t. Morita, and t. Uchiyama, "Design and implementation of high-speed visual tracking system for real time motion analysis," *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276 - Volume 7276*, ser. ICPR '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 478–483.

[98] Y. Azoz, L. Devi, M. Yeasin, and R. Sharma, "Tracking the human arm using constraint fusion and multiple-cue localization," *Machine Vision and Applications*, vol. 13, pp. 286–302, 2003.

[99] J. Lee, S.-H. Lim, J.-W. Yoo, K.-W. Park, H.-J. Choi, and K. H. Park, "A ubiquitous fashionable computer with an i-throw device on a location-based service environment," *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 59–65.

[100] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 2007, pp. 159–168.

[101] J. Rekimoto, "Tilting operations for small screen interfaces," *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 1996, pp. 167–168.

[102] G. Levin and P. Yarin, "Bringing sketching tools to keychain computers with an acceleration-based interface," *CHI '99: CHI '99 extended abstracts on Human factors in computing systems.* New York, NY, USA: ACM, 1999, pp. 268–269.

[103] H.-J. Lee, M. Khandelwal, and A. Mazalek, "Tilting table: a movable screen," *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction.* New York, NY, USA: ACM, 2007, pp. 93–96.

[104] K. Hinckley, "Synchronous gestures for multiple persons and computers," *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ser. UIST '03. New York, NY, USA: ACM, 2003, pp. 149–158.

[105] S.-J. Cho, C. Choi, Y. Sung, K. Lee, Y.-B. Kim, and R. Murray-Smith, "Dynamics of tilt-based browsing on mobile devices," *CHI '07: CHI '07 extended abstracts on Human factors in computing systems.* New York, NY, USA: ACM, 2007, pp. 1947–1952.

[106] K. Partridge, S. Chatterjee, V. Sazawal, G. Borriello, and R. Want, "Tilttype: accelerometer-supported text entry for very small devices," *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 2002, pp. 201–204.

[107] J. Mantyjarvi, F. Paternò, Z. Salvador, and C. Santoro, "Scan and tilt: towards natural interaction for mobile museum guides," *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services.* New York, NY, USA: ACM, 2006, pp. 191–194.

[108] A. Crossan and R. Murray-smith, "Variability in wrist-tilt accelerometer based gesture interfaces," *Lecture Notes In Computer Science, MobileHCI 2004: 6th International Symposium*, 2004, pp. 144–155.

[109] A. Pirhonen, S. Brewster, and C. Holguin, "Gestural and audio metaphors as a means of control for mobile devices," *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 2002, pp. 291–298.

[110] H. Kimura, Y. Okuda, and T. Nakajima, "Cookieflavors: Rapid composition framework for tangible media," *Proceedings of the The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 100–109.

[111] "Nokia research center," http://research.nokia.com/.

[112] D. Sato, B. Shizuki, M. Miura, and J. Tanaka, "Menu-selection-based japanese input method with consonants for pen-based computers," *Proceedings of APCHI 2004*, 2004, pp. 399–408.

[113] F. Berard, J. Ip, M. Benovoy, D. El-Shimy, J. R. Blum, and J. R. Cooperstock, "Did "minority report" get it wrong? superiority of the mouse over 3d input devices in a 3d placement task," *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, ser. INTERACT '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 400–414.

List of Publications

Journals

Atia Ayman and Jiro Tanaka. "Interaction with tilting gestures in ubiquitous environments", *International Journal of UbiComp*, Vol.1, No.3, 2010, pp.1-13.

Conference proceedings

Atia ayman, Shin Takahashi and Jiro Tanaka. "Coin Size Wireless Sensor Interface for Interaction with Remote Displays," *Proceedings of the 12th International Conference on Human-Computer Interaction (HCI International 2007), PartII, LNCS 4551* , pp.733-742, Beijing, China, July 22-27, 2007.

Atia ayman, Shin Takahashi, Kazuo Misue and Jiro Tanaka. "UbiGesture: Customizing and Profiling Hand Gestures in Ubiquitous Environment". *Proceedings of 13th International Conference on Human-Computer Interaction (HCI International 2009), Human-Computer Interaction, Part II, LNCS 5611*, pp.141-150, San Diego, California, USA, July 19-24, 2009.

Atia ayman, Shin Takahashi and Jiro Tanaka. "Smart Gesture Sticker: Smart Hand Gestures Profiles for Daily Objects Interaction". *Proceedings of 9th IEEE/ACIS international conference on computer and information science* , pp.482-487, Yamagata, Japan, August 18-20, 2010.