INTERACTIVE GESTURE CHAIR

Name:Shamsul Arefin

St.ID:114414

Name:Muhaiminul Islam Foysol

St.ID:114416

Thesis Supervisor:

Md. Kamrul Hasan ,PHD

Associate Professor

Islamic University of Technology

OCTOBER 30, 2015

Table of contents

Abstract:

A computer operator or an office worker now-a-days spend most his time sedentary. Technology is improving day by day and we are also becoming mechanic. Office employees have to stay in front of the PC almost all the day long and this sedentary behavior is not good for health. As a consequence people of all ages can suffer through health problems. Doctor suggested to make some movements during office time so that the probability of attacking by chronic diseases decreases. Therefore many proposals had been proposed to keep people moving during work time. However, for most office workers it is difficult to achieve a considerable reduction of the time spent seated within the office environment. To promote physical activity even in such sedentary situations, this work explores the possibilities of using an interactive office furniture to smoothly integrate physical activity into the daily working routine Chair is the most frequently used furniture by the office workers. We have made a system to interact with the PC using chair. By equipping motion sensing sensors with a chair the movement of the user can be detected which can be used as input device for PC. This way, the “Interactive Gesture Chair” becomes an input device that is ubiquitously embedded into the working environment, and provides an office worker with the possibility to use the movements of his body for rotating, tilting, or bouncing a chair to intuitively control the operations in Desktop Computer. In our thesis work we have used thresholding to define the chair gestures/movements. By analyzing the result we saw that threshold based gestures vary with the variety of weight of the people i.e. the people with different weights have different threshold value for same gesture. Then we tried machine learning algorithms to define gestures so that defined gestures should work for the people of different weights. First of all we tried Euclidian distance method to define gestures. Then we tried Dynamic Time Warping algorithm to define gestures and then we tried decision tree to find a universal threshold for gestures. These defined gestures can be used to control many application of PC. We defined these gestures to control Windows Multimedia Player.

1 Introduction:

During everyday office work used to control our computers with keyboard and mouse sitting in front of computers. When we work in our office in front of computer we spend most of our work time sedentary throughout our daily routine. We remain seated when we are on the way to office by car, during meetings, during lunch etc. This sedentary behavior is considered as an important ergonomic factor which may lead to a variety of chronic diseases for both aged and young persons. Due to long time staying in chair, people may suffer from back pain, neck pain, etc. Therefore many proposals had been proposed to keep people moving during work time. However, for most office workers it is difficult to achieve an Ease of Use considerable reduction of the time spent seated within the office environment. To promote physical activity even in such sedentary situations, this work explores the possibilities of using an interactive office chair to smoothly integrate physical activity into the daily working routine. By equipping a flexible chair with a motion sensor, the movements of a person sitting on the chair can be tracked and transformed into input events that trigger various actions on a computer. Besides, interacting with computers for a long period of time is tiresome, so there is a need of an alternate way to do the tasks other than regular mouse, keyboard operations. This way, the “Interactive Gesture Chair” becomes an input device that is ubiquitously embedded into the working environment, and provides an office worker with the possibility to use the movements of his body for rotating, tilting, or bouncing a chair to intuitively control operations in Desktop Computer. Ubiquitous Information Systems (UIS) provide means for supporting single actors and groups in real-world situations by services over ubiquitous computing technologies anywhere and anytime. UIS come with more complex requirements than the more strongly constrained Information Systems for office settings. Contents shall be seamlessly provided by any kind of mobile or embedded device based on loosely coupled service infrastructures while users are moving in physical environments. Furthermore, situated communication and collaboration of user groups in highly dynamic and context-dependent physical environments are far more complex than well-structured online environments. While working on a desktop computer [2], we are used to traditional mouse and keyboard input. We have learned how to handle these devices and utilize them as powerful tools for our daily computer-based activities. Their operation, however, involves making the same small repetitive movements with our hands over and over again, while the rest of our body remains largely unchallenged [15,29]. On the other hand, with computing interfaces becoming more and more ubiquitous, we see increasing numbers of input technologies making their way into the user interface that support free-form manipulation of digital objects through touch, speech, gaze, or body gestures. Taking this evolution further, we believe that interaction through furniture has high potential to open a new input domain beyond the current mouse-and-keyboard paradigm. Thus, in a common desktop workplace for example, where we are surrounded by functional furniture of various kinds, we believe that these readily available elements (e.g., chair, desk) could actually become part of the computer interface [3]. The ubiquity of seated activities [2] in our daily lives, for example, provides an excellent context for the development of alternative human-computer interfaces. Thus, we have only recently begun to explore the potential of chair-based interaction as novel input dimension for desktop interfaces [23]. Extending this concept in the present work, we explore the

Potential of gestural chair interaction during regular desktop activities. In office environment when user needs to listen music or do some secondary task besides the primary task then it also interrupts the primary work, therefore decreasing performance.

So regardless of the specific use context, such scenarios always involve a focused primary task and a peripheral secondary task requiring temporary attention, only to slide back into the periphery again. Still, such short interruptions can disrupt our concentration, make us lose focus and decrease our performance [3]. This is especially problematic in the office context, where we want our attention focused on the actual work. Thus, it is desirable for transitions between primary and secondary tasks to work rather effortlessly [3], with minimal physical and mental demand. We think that this type of interaction with secondary tasks should aim at keeping a task in the periphery of our attention, while still providing the opportunity to control it when needed. In our work, we focus on improving users' interaction with peripheral tasks in the office context by providing the opportunity for gestural interaction with flexible office chair (e.g., navigating to the next item in a playlist by briefly swinging the lower body to the right while sitting on the chair). Thereby, in comparison to traditional input devices, our goal is to reduce physical constraints [2] (i.e., supporting hands-free, eyes-free interaction) and to support input that can take place nearly in parallel with a user's primary task. We believe that this flexible chair is very well-suited for such peripheral interaction styles due to its ubiquity and currently untapped potential as input medium. In this work, we explain the implementation of our sensor-based chair interface, we explore three different gesture styles for chair interaction (i.e., rotate left, rotate right, tilt backward, bounce), and describe possible application scenarios (such as music player control) of using recognized input gestures for the control of a desktop computer.

The results of explorative gesture studies provide implications on the benefits and limitations of the proposed concept. User feedback and performance show high potential of chair gestures as additional input modality for ubiquitous, hands-free interaction with a desktop computer. Considering health issues, in our proposed architecture, user has to move his body which is at least better than staying a long period of time in a same position. Recently MIT media interaction LAB have started working on chair gesture. Also lots of research work are ongoing to explore the additional capabilities of this kind of gestural interaction between human and computer. Utilizing these chair gesture into a frequently used application is challenging. Actually success of this 'interactive gesture chair' depends on proper integration of the gestures with frequently used desktop application and accuracy of detecting movements.

2 Related Works

Following the trend of designing more natural ways of interaction with computer systems, several attempts have been made to develop technologies that extend interaction beyond long-established mouse-and-keyboard input. Along with that, we see more and more devices with embedded sensing and communication capabilities [30].While early work in the field of smart office environments has demonstrated the ubiquitous integration of interactive technology [27], research interest has more recently also considered the extension of interaction into the periphery. The Unadorned Desk is an example for this kind of interaction, which used the physical space around a desktop computer as input canvas [12]. Similarly, our work adds to the research conducted within this field by proposing the utilization of a chair as ubiquitous input device.

While [19] Wexelblat [8] and Quek [7] both claim that semaphoric gestures are not natural for computer interactions and represent only a small part of human communication, we suggest that the "small part" is potentially well matched to secondary interactions. Secondary or background tasks are those which can take place concurrently with the primary task [2]. When the demands of the secondary task cause it to become the user's primary focus, referred to as "forced divided attention" [9], negative performance effects on the primary task can occur [3]. Gestures have been used as an approach for reducing these effects, facilitating "eyes-free" interaction as demonstrated with marking menus, for example [5].Secondary task interactions are often considered a primary concern within notification systems research where a secondary event such as an instant messenger window may interrupt a primary task such as editing a text document [6]. Research in this area investigates the effects of distraction and recovery caused to a primary task by an interruption, looking at methods for reducing distraction through positioning of a notification and prioritizing interruptions [7,9].

Previous works [19] also done to utilize gestural device use of projectors [ref to 20, 31], a tabletop computer as desk replacement [6], or by adding tablet computers next to the display as an interactive region [8]. These tend to be complex (or expensive) to set up. In contrast, the new generation of depth-sensing technologies mean that detecting touches and hovering is low-cost, such as via Leap Motion or Microsoft's Kinect camera. The problem is that these technologies do not provide visual feedback.

Chairs have been used in computer-based scenarios, where body movements are sensed and interpreted as implicit input to a system [2]. The Internet Chair for example, used orientation tracking to support social situation awareness through spatialized audio [7]. The Chair Mouse translated natural chair rotation into large-scale cursor movement for navigation on large displays [8]. Rather than exploiting existing user behavior, however, our work focuses on the encouragement of new interaction metaphors that are currently not possible with traditional input devices. Based on a similar concept, ChairIO introduced a gaming interface that allows users to navigate virtual environments by controlling a flexible chair with their body [1]. In contrast to this, we are interested in the application of input metaphors for more general scenarios beyond the gaming context, which have the potential to enrich and facilitate user interaction during everyday desktop work.

In particular, our work explores chair input in the context of gestural interaction, which has been proposed as alternative human-computer interface in the context of hand, finger, or full-body gestures [17,24]. Integrating our physical body into interactions with digital interfaces [7], it has been shown that such gestural interfaces can potentially provide wide-reaching benefits over traditional input devices, including naturalness and expressiveness of interaction, learnability, directness of interaction, available degrees of freedom, and exploitation of existing dexterous skills [2,5].

So like these many efforts have been made to design calm technologies that aim to reduce information overload by letting users select what information is at the center of their attention [11]. Moreover, special interest has been on the design of inattentive interaction techniques that can be easily performed in the periphery of attention. Whack[38] Gestures is an example of an inattentive, inexact interaction technique, allowing users to interact without the use of fine motor skills or detailed visual attention [3]. It has been shown that such semaphoric gestures can provide substantial benefits for secondary task interactions [4], and allow users to vary their level of engagement with a task [7].

Chairs have been used for unobtrusive [3] measurement of body movements [7, 10] and the development of posture recognition systems for the classification of a person's sitting behavior [20] or emotions [4,21]. Chair-based tracking data has furthermore been explored for supporting people in learning [10,14] and ambient assisted living [7] environments. Finally, the usage of interactive chairs has also been explored as input devices for the navigation of virtual 2D and 3D environments in computer games [4], or for direct control of the mouse cursor in a desktop environment [6].The usage of hand, finger and full-body gestures has been explored extensively in many works [18], as well as their usage for primary and secondary tasks like controlling a music application while typing [ref to 12] or controlling an interface while driving [2].

The ChairIO[6] was also a famous a chair-based computer interface developed in the interactive media / virtual environments (im/ve) group at the University of Hamburg. The motivation for the development of the ChairIO came from one of the group's interests, the search for alternative interaction devices and techniques. Of particular interest is the search for intuitive methods to control movement in immersive virtual environments. Experience, using the standard hand-coupled devices, had shown some of the disadvantages and deficiencies of those methods. A chair, especially one which allows movement in several directions, seemed like a potential candidate to support 3D motion, both directional and rotational. After some research into this idea, a new device and method was introduced in [2] the chair-based interface to provide intuitive and hands-free navigational control of a virtual environment.

Based on these related works, we have only recently come up with the idea of applying different gesture styles to flexible chairs [23], since we believe that many of the benefits seen in in other body-based gestural interfaces can transfer to the desktop environment. Building upon this work, we are further exploring the specific subset of semaphoric [3] chair gesture to find some effective gestures.

Now-a-days people are trying to design some natural ways to interact with computers instead of mouse and keyboard. To follow that, people are equipping sensors with frequently used things to establish communication with PC. Chair is one of the frequently used furniture to equip sensor with to use as input device of computer. In the previous works of Media Interaction Lab [14, 21] they have used Gyroscope and Accelerometer to detect movement of the chair and controlled multimedia player by defined gestures. Another chair based gesture detection [15] uses Lumia smartphone for getting sensors data which performs both music player control and web browsing. To detect the movement of the chair we need to equip accelerometer and gyroscope with the chair. For that we need an MPU-6050 sensor, which have to be well equipped with the chair and will detect chair movements.

3 Background study:

3.1 Ubiquitous Computing:

Ubiquitous computing is a paradigm in which the processing of information is linked with each activity or object as encountered. It involves connecting electronic devices, including embedding microprocessors to communicate information. Devices that use ubiquitous computing have constant availability and are completely connected. Ubiquitous computing focuses on learning by removing the complexity of computing and increases efficiency while using computing for different daily activities. Ubiquitous computing is also known as pervasive computing, every ware and ambient intelligence. The main focus of ubiquitous computing is the creation of smart products that are connected, making communication and the exchange of data easier and less obtrusive. Here computing is made to appear anytime and everywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. A user interacts with the computer, which can exist in many different forms, including laptop computers, tablets and terminals in everyday objects such as a fridge or a pair of glasses. The underlying technologies to support ubiquitous computing include Internet, advanced middleware, operating system, mobile code, sensors, microprocessors, new I/O and user interfaces, networks, mobile protocols, location and positioning and new materials.

Ubiquitous computing is viewed less as a discrete field of technology, but rather as an emerging application of information and communications technology that is integrated into the everyday world more than ever before. The goal is to meet the claim of “everything, always, everywhere “for data processing and transmission through the ubiquity of ICT systems. The following characteristics define this application paradigm: (i)miniaturization: ICT components are becoming smaller and more mobile,(ii) embedding: as ICT components are integrated into everyday objects, they transform them into smart objects, (iii)networking: ICT components are linked to each other and communicate generally via radio; they are therefore not part of a fixed environment or application, but are instead designed to form networks spontaneously, (iv)ubiquity: while embedded ICT components are increasingly ubiquitous, they are at the same time increasingly less noticeable—or even invisible—to most people, (v)context-awareness: ICT components use sensors and communication to collect information about their users and environment and adjust their behavior accordingly (Wagner et al., 2006). Communications technologies and microelectronics, in particular, are key requirements for almost all ubiquitous computing applications. Although energy autarky is certainly not an important characteristic of all ubiquitous computing applications, supplying energy is clearly a central task. Maturation and availability of ubiquitous computing-relevant technologies is expected soon, within the next one to four years; nearly all of the technological requirements needed for ubiquitous computing should be met in the foreseeable future. Ubiquitous computing will permeate everyday life—both private and working—and is therefore expected to have far-reaching consequences that will be reflected in a variety of socio-economic contexts. Both positive and negative effects are likely in equal measure at several levels. Safety and privacy, for example, make up two ends of one key pole. The following discussion presents the impact of ubiquitous computing in terms of privacy, economics, society and the digital divide.

Key features of ubiquitous computing include consideration of the human factor and placing of the paradigm in a human, rather than computing, environment, use of inexpensive processors, thereby reducing memory and storage requirements which is now a days a critical issue , capturing of real-time attributes, Totally connected and constantly available computing devices, focus on many-to-many relationships, instead of one-to-one, many-to-one or one-to-many in the environment, along with the idea of technology, which is constantly present.

3.2 Gestures:

A gesture is a form of non-verbal communication or non-vocal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with, speech. Gestures include movement of the hands, face, or other parts of the body. Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention. Gestures allow individuals to communicate a variety of feelings and thoughts, from contempt and hostility to approval and affection, often together with body language in addition to words when they speak. Gesture processing takes place in areas of the brain such as Broca's and Wernicke's areas, which are used by speech and sign language. In fact, language is thought by some scholars to have evolved in Homo sapiens from an earlier system consisting of manual gestures. The theory that language evolved from manual gestures, termed Gestural Theory, dates back to the work of 18th-century philosopher and priestAbbé de Condillac, and has been revived by contemporary anthropologist Gordon W. Hewes, in 1973, as part of a discussion on the origin of language.

3.2.1 Study on gestures:

Gestures have been studied throughout the centuries from different viewpoints. During the Roman Empire, Quintilian studied in his Institution Oratoria how gesture may be used in rhetorical discourse. Another broad study of gesture was published by John Bulwer in 1644. Bulwer analyzed dozens of gestures and provided a guide on how to use gestures to increase eloquence and clarity for public speaking. Andrea De Jorio published an extensive account of gestural expression in 1832.

3.2.2 Categories of gestures:

Pointing at another person with an extended finger is considered rude in many cultures.

List of gestures:

Although the scientific study of gesture is still in its infancy, some broad categories of gestures have been identified by researchers. The most familiar are the so-called emblems or quotable gestures. These are conventional, culture-specific gestures that can be used as replacement for words, such as the hand wave used in the US for "hello" and "goodbye". A single emblematic gesture can have a very different significance in different cultural contexts, ranging from complimentary to highly offensive[8] The page List of gestures discusses emblematic gestures made with one hand, two hands, hand and other body parts, and body and facial gestures.

Another broad category of gestures comprises those gestures used spontaneously when we speak. These gestures are closely coordinated with speech. The so-called beat gestures are used in conjunction with speech and keep time with the rhythm of speech to emphasize certain words or phrases. These types of gestures are integrally connected to speech and thought processes.

Other spontaneous gestures used during speech production known as iconic gestures are more full of content, and may echo, or elaborate, the meaning of the co-occurring speech. They depict aspects of spatial images, actions, people, or objects. For example, a gesture that depicts the act of throwing may be synchronous with the utterance, "He threw the ball right into the window. “Such gestures that are used along with speech tend to be universal. For example, one describing that he/she is feeling cold due to a lack of proper clothing and/or a cold weather can accompany his/her verbal description with a visual one. This can be achieved through various gestures such as by demonstrating a shiver and/or by rubbing the hands together. In such cases, the language or verbal description of the person does not necessarily need to be understood as someone could at least take a hint at what's being communicated through the observation and interpretation of body language which serves as a gesture equivalent in meaning to what's being said through communicative speech.

Studies affirm a strong link between gesture typology and language development. Young children under the age of two seem to rely on pointing gestures to refer to objects that they do not know the names of. Once the words are learned, they eschewed those referential (pointing) gestures. One would think that the use of gesture would decrease as the child develops spoken language, but results reveal that gesture frequency increased as speaking frequency increased with age. There is however a change in gesture typology at different ages, suggesting a connection between gestures and language development. Children most often use pointing and adults rely more on iconic and beat gestures. As children begin producing sentence-like utterances, they also begin producing new kinds of gestures that adults use when speaking (iconics and beats). Evidence of this systematic organization of gesture is indicative of its association to language development.

Gestural languages such as American Sign Language and its regional siblings operate as complete natural languages that are gestural in modality. They should not be confused with finger spelling, in which a set of emblematic gestures are used to represent a written alphabet. American Sign Language is different from gesturing in that concepts are modeled by certain hand motions or expressions and has a specific established structure while gesturing is more malleable and has no specific structure rather it supplements speech. We should note, that before an established sign language was created in Nicaragua after the 1970s, deaf communities would use "home signs" in order to communicate with each other. These home signs were not part of a unified language but were still used as familiar motions and expressions used within their family—still closely related to language rather than gestures with no specific structure. This is similar to what has been observed in the gestural actions of chimpanzees. Gestures are used by these animals in place of verbal language, which is restricted in animals due to their lacking certain physiological and articulatory abilities that humans have for speech. Corballis (2009) asserts that "our hominin ancestors were better pre-adapted to acquire language-like competence using manual gestures than using vocal sounds." This leads to a debate about whether humans, too, looked to gestures first as their modality of language in the early existence of the species. The function of gestures may have been a significant player in the evolution of language.

3.2.3 Social significance:

Gestures, commonly referred to as “body language,” play an important role in industry. Proper body language etiquette in business dealings can be crucial for success. However, gestures can have different meanings according to the country in which they are expressed. In an age of global business, diplomatic cultural sensitivity has become a necessity. Gestures that we take as innocent may be seen by someone else as deeply insulting. The following gestures are examples of proper etiquette with respect to different countries’ customs on salutations: In the United States, “a firm handshake, accompanied by direct eye contact, is the standard greeting. Direct eye contact in both social and business situations is very important.” In the People’s Republic of China, “the Western custom of shaking a person's hand upon introduction has become widespread throughout the country. However, oftentimes a nod of the head or a slight bow will suffice.” In Japan, “the act of presenting business cards is very important. When presenting, one holds the business card with both hands, grasping it between the thumbs and forefingers. The presentation is to be accompanied by a slight bow. The print on the card should point towards the person to which one is giving the card.” In Germany, “it is impolite to shake someone's hand with your other hand in your pocket. This is seen as a sign of disrespect” In France, “a light, quick handshake is common. To offer a strong, pumping handshake would be considered uncultured. When one enters a room, be sure to greet each person present. A woman in France will offer her hand first.

Gestures are also a means to initiate a mating ritual. This may include elaborate dances and other movements. Gestures play a major role in many aspects of human life. Gesturing is probably universal; there has been no report of a community that does not gesture. Gestures are a crucial part of everyday conversation such as chatting, describing a route, negotiating prices on a market; they are ubiquitous. Additionally, when people use gestures, there is a certain shared background knowledge. We use similar gestures when talking about a specific action such as how we gesture the idea of drinking out of a cup. When an individual makes a gesture, another person can understand because of recognition of the actions/shapes. Gestures have been documented in the arts such as in Greek vase paintings, Indian Miniatures or European paintings.

Gestures play a central role in religious or spiritual rituals such as the Christian sign of the cross. In Hinduism and Buddhism, a mudra (Sanskrit, literally "seal") is a symbolic gesture made with the hand or fingers. Each mudra has a specific meaning, playing a central role in Hindu and Buddhist iconography. An example is the Vitarka mudra, the gesture of discussion and transmission of Buddhist teaching. It is done by joining the tips of the thumb and the index together, while keeping the other fingers straight.

Gestures according to Neurology:

Gestures are processed in the same areas of the brain as speech and sign language such as the left inferior frontal gyrus (Broca's area) and the posterior middle temporal gyrus, posterior superior temporal sulcus and superior temporal gyrus (Wernicke's area). It has been suggested that these parts of the brain originally supported the pairing of gesture and meaning and then were adapted in human evolution "for the comparable pairing of sound and meaning as voluntary control over the vocal apparatus was established and spoken language evolved".As a result, it underlies both symbolic gesture and spoken language in the present human brain. Their common neurological basis also supports the idea that symbolic gesture and spoken language are two parts of a single fundamental semiotic system that underlies human discourse. The linkage of hand and body gestures in conjunction with speech is further revealed by the nature of gesture use in blind individuals during conversation. This phenomenon uncovers a function of gesture that goes beyond portraying communicative content of language and extends David McNeill's view of the gesture-speech system. This suggests that gesture and speech work tightly together, and a disruption of one (speech or gesture) will cause a problem in the other. Studies have found strong evidence that speech and gesture are innately linked in the brain and work in an efficiently wired and choreographed system. McNeill's view of this linkage in the brain is just one of three currently up for debate; the others declaring gesture to be a "support system" of spoken language or a physical mechanism for lexical retrieval.

Because of this connection of co-speech gestures—a form of manual action—in language in the brain, Roel Willems and Peter Hagoort conclude that both gestures and language contribute to the understanding and decoding of a speaker's encoded message. Willems and Hagoort's research suggest that "processing evoked by gestures is qualitatively similar to that of words at the level of semantic processing." This conclusion is supported through findings from experiments by Skipper where the use of gestures led to "a division of labor between areas related to language or action (Broca's area and premotor/primary motor cortex respectively)." The use of gestures in combination with speech allowed the brain to decrease the need for "semantic control." Because gestures aided in understanding the relayed message, there was not as great a need for semantic selection or control that would otherwise be required of the listener through Broca's area. Gestures are a way to represent the thoughts of an individual, which are prompted in working memory. The results of an experiment revealed that adults have increased accuracy when they used pointing gestures as opposed to simply counting in their heads (without the use of pointing gestures) Furthermore, the results of a study conducted by Marstaller and Burianová suggest that the use of gestures affect working memory. The researchers found that those with low capacity of working memory who were able to use gestures actually recalled more terms than those with low capacity who were not able to use gestures.

Although there is an obvious connection in the aid of gestures in understanding a message, "the understanding of gestures is not the same as understanding spoken language." These two functions work together and gestures help facilitate understanding, but they only "partly drive the neural language system."

3.3 MPU-6050

3.3.1 MPU-6050 Overview:

According to the InvenSense MPU-6050 datasheet, this chip contains a 3-axis gyroscope and a 3-axis accelerometer. This makes it a “6 degrees of freedom inertial measurement unit” or 6DOF IMU, for short. Other features include a built in 16-bit analog to digital conversion on each channel and a proprietary Digital Motion Processor™ (DMP) unit.

The DMP combines the raw sensor data and performs some complex calculations onboard to minimize the errors in each sensor. Accelerometers and gyros have different inherent limitations, when used on their own. By combining the data from the two types of sensors and using some math wizardry (a process referred to as sensor fusion), you apparently can get a much more accurate and robust estimate of the heading. The DMP on the MPU6050 does exactly that and returns the result in “quaternions”. These can then can be converted to yaw-pitch-roll, or to Euler angles for us humans to read and understand. The DMP also has a built in auto-calibration function that definitely comes in handy, as we will see later.

The biggest advantage of the DMP is that it eliminates the need to perform complex and resource intensive calculations on the Arduino side. The main downside is that it seems that the manufacturer did not provide much information on the proprietary inner workings of the DMP. Nevertheless, smart and creative folks figured out how to use its main features, and were nice enough to share the results with the rest of us. You can still pull the raw, accelerometer and gyro data as well, disabling the DMP, if this works better for your application, or you want to apply your own filtering and sensor fusion algorithms.

The MPU-6050 communicates with a microcontroller through an I2C interface. It even has a built in an additional I2C controller that allows it to act as a master on a second I2C bus. The intention is for the IMU to read data from say, an external magnetometer (hooked up via those XDA / XCL pins you see on the breakout board) and send it to the DMP for processing. I have not found much detail on how to make the DMP use external magnetometer data yet, but fortunately that is not needed for my self-balancing robot at this point.

Lastly, the MPU-6050 has a FIFO buffer, together with a built-in interrupt signal. It can be instructed to place the sensor data in the buffer and the interrupt pin will tell the Arduino, when data is ready to be read.

3.3.2 MPU-6050 / GY-521 break-out board schematics:

Below is the schematic of the GY-521 break-out board for the MPU6050 chip.

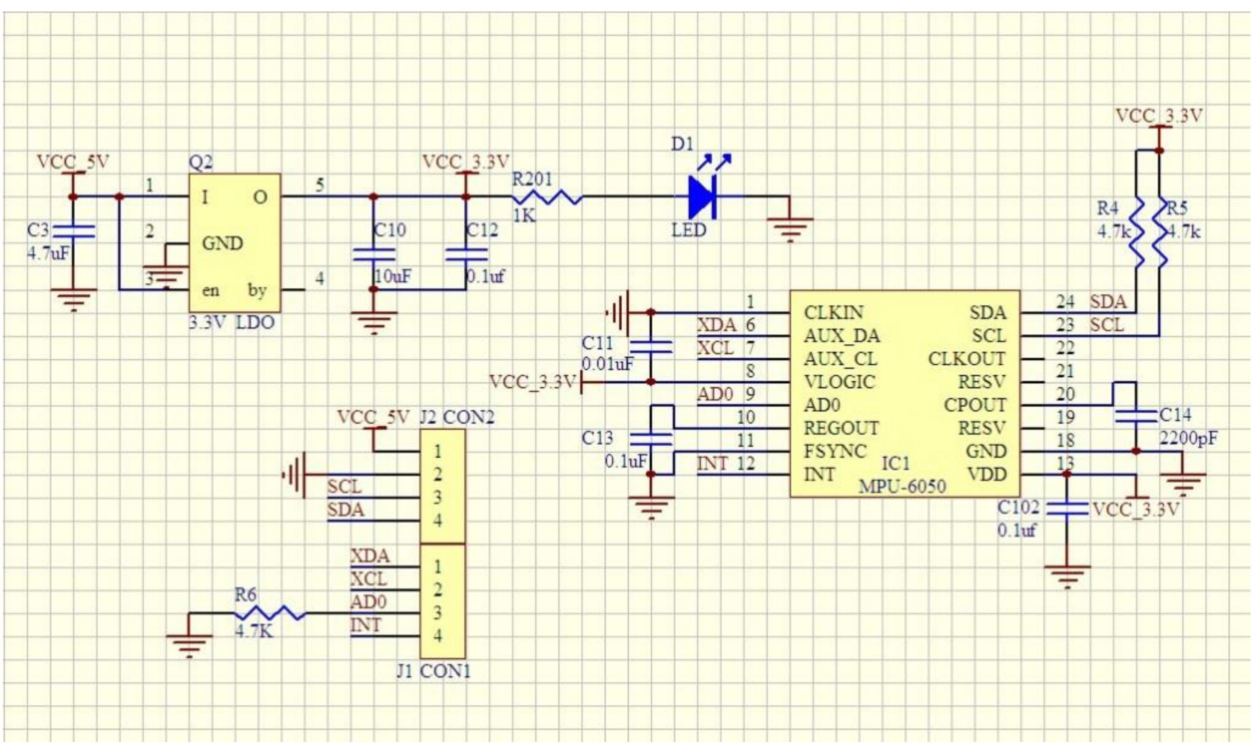

Fig: MPU6050-schematic Circuit View

3.3.3 Hooking the MPU-6050 / GY-521 to an Arduino Uno:

The InvenSense MPU6050 chip is a 3.3V IC, with a working voltage range of 2.375V-3.46V, according to its datasheet. As you can see from the schematics above, the GY-521 breakout board has a built in low drop-out voltage regulator, so it is safe to power the chip through the Arduino 5V rail. This is recommended, as due to the voltage drop-out of the regulator on the VCC line, using the Arduino 3.3V rail may not provide enough voltage. I tested powering the chip both with 3.3V and 5V from the Arduino successfully, but in my final set-up opted for the 5V input.

Based on what we have read online and what we saw on my tests, the 3.3V SDA / SCL lines of the IMU work fine connected directly to the corresponding Arduino 5V I2C pins. If you want to be absolutely safe, you could use a level shifter, voltage divider, or an inline 10k resistor to protect the MPU6050 I2C lines. I opted for simplicity over safety, in my set-up and (so far) all is well.

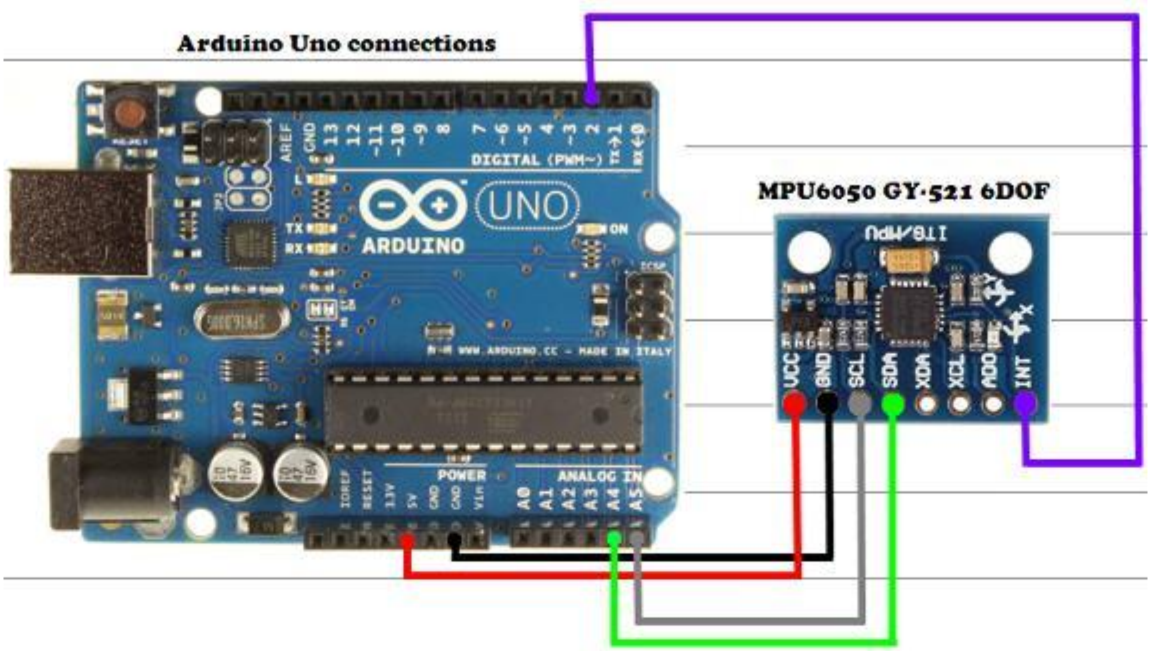

Fig: Arduino Uno connections

Pin configurations:

MPU6050 / GY-521	Arduino UNO Pin
VCC	5V (the GY-521 has a voltage regulator)
GND	GND
SDA	A4 (I2C SDA)
SCL	A5 (I2C SLC)
INT	D2 (interrupt #0)

3.3.4 How an accelerometer works:

An accelerometer works on the principle of piezo electric effect. Here, imagine a cuboidal box, having a small ball inside it, like in the picture above. The walls of this box are made with piezo electric crystals. Whenever you tilt the box, the ball is forced to move in the direction of the inclination, due to gravity. The wall with which the ball collides, creates tiny piezo electric currents. There are totally, three pairs of opposite walls in a cuboid. Each pair corresponds to an axis in 3D space: X, Y and Z axes. Depending on the current produced from the piezo electric walls, we can determine the direction of inclination and its magnitude. We can have a look at this figure below.

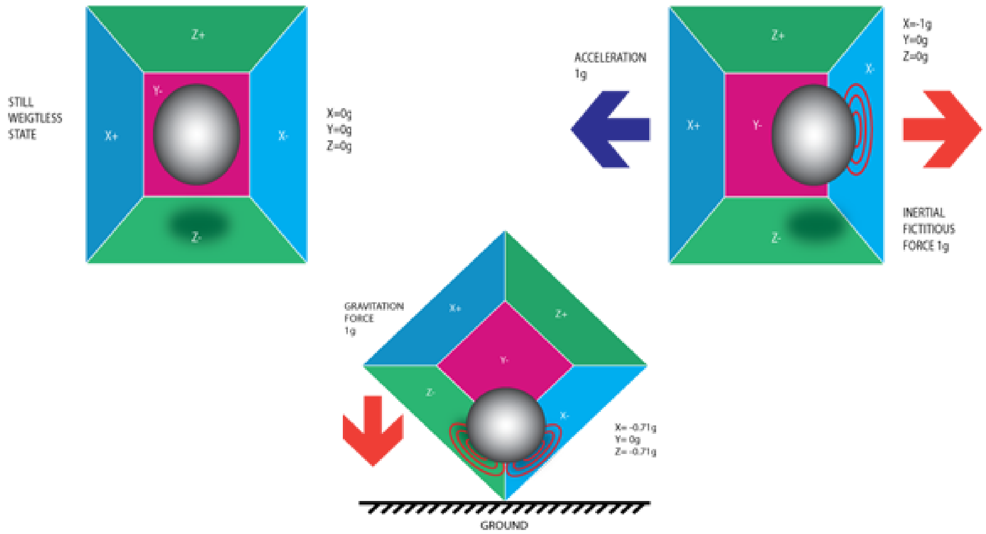

Fig:Accelerometer working mechanism

3.3.5 How gyroscope works:

Gyroscopes work on the principle of Coriolis acceleration. Imagine that there is a fork like structure that is in constant back and forth motion. It is held in place using piezo electric crystals. Whenever, you try to tilt this arrangement, the crystals experience a force in the direction of inclination. This is caused as a result of the inertia of the moving fork. The crystals thus produce a current in consensus with the piezo electric effect, and this current is amplified. The values are then refined by the host microcontroller. Now check this short video that explains, how a MEMS gyroscope works.

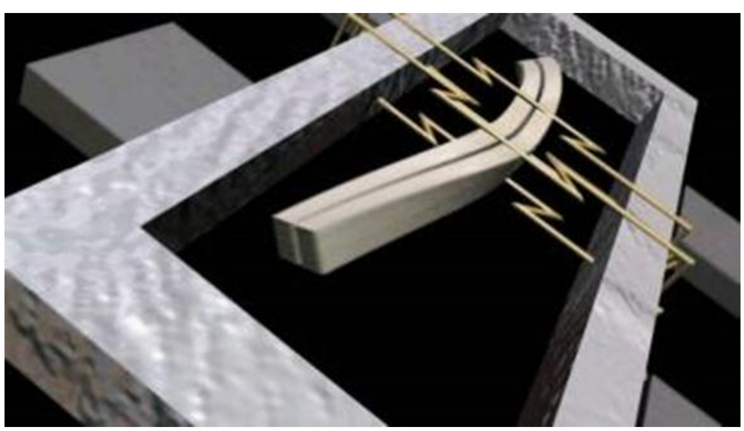

Fig: Gyroscope working principle

4 Problem statement:

Recent works show that, chair based gesture was approached through basic brute force mechanism, which involves thresholding in axes values. We also implemented that in the previous semester, but after implementing that we saw major drawbacks of the traditional thresholding approach, major drawback was threshold seems to be different for different person, different age, different height etc. We have seen that if we determine threshold for one people or group of people that works less efficiently for any other people of different age or different weight. Threshold have to be changed for different kind of people. The threshold which works for young people doesn't work good for mid aged people, the threshold have to be changed manually for the mid-range people. If we choose mid-range people for standard it also shows less accuracy for aged person, if we set threshold for a stand weighted person, then we saw threshold varies for greater weighted person, In this way the thresholding becomes a cumbersome way to resolve the detection of gestures for different people. So wee need to think some alternatives to this traditional thresholding approach, like machine learning or different improved thresholding approach which is based on decision tree.

Decision tree calculates threshold in a more different and efficient way which takes data's and class label, then learns from the data and creates good threshold. It is very important because if we are designing something that works for a very few people and if more people wants to use that system that system needs to be changed or configured manually every time, needs to calibrate for different group of people, but it shouldn't be the case, what we are implanting should work for numerous number of people, then the work will be great. Also sitting style is different for different kind of people, teenagers may sit in such way which may differs from the mid-range people or aged people. It's because of the center of gravity is different for different people, so while thresholding we have seen potential effect of this issue. For the changing of people weight reacts in different point that means center of gravity is changed, this changes the threshold for them, and so it also becomes an issue.

Recent works show that people are determining threshold in manual ways like x, y, z axis thresholding for both accelerometer and gyroscope. We worked on that type of thresholding in the previous semester we find out the drawbacks of it which was mainly reconfiguring and calibrating the system every time. We thought of some different approaches like thresholding with decision tree or implementing machine learning algorithm, in this context machine learning approaches like dtw or hmm can be used. Also about accuracy, accuracy means percentage of correctly determining the movements. Machine learning can give better results but for machine learning we have to train with lots of data with the classifier label also, but for decision tree based thresholding approach we don't need that much amount of data but decision tree thresholding may give less accuracy, so after finding effective gestures it is also necessary to analyze data and to find out which mechanism machine learning or decision tree thresholding is good.

Finding a suitable position to accouter sensor that will give a good precision of data is also important in this context because after setting up the sensor in the chair we saw that for setting the sensor in different place of the chair, the sensors give different kind of data, somewhere all values are negative always and somewhere mixture of negative and positive values, at some position sensors give meaningful data which can be easily detected and can be used for thresholding and other works, so it's another kind of issue here to effectively set up or accouter the sensor in the chair or in the system.

Distinguishing unconscious movements from gesture movements is also important because using the ubiquitous system, which takes input continuously from the movements can also give wrong data, it is called wrong because the user is not performing the gestures rather the system is considering the movements as gesture which is totally wrong, so what we can do is somehow isolate the state of taking input and not taking input such that user makes usual movement, incorrectly gesture is not detected.

Raw data collected from sensor have lots of noise especially the data which have no variations in them to detect change can be filtered and these data can be noise, also noise can unnecessarily create values in raw data which in return makes gestures what was not actually performed, so after getting raw data some filter such as high pass filter or low pass filter is needed to filter the data to remove the unnecessary signals.

5 Proposed methodology:

5.1 Prototype:

Before going to implementation, first we created a prototype of our system as shown in the preceding figures. It is portable system because we can move the chair anywhere while the sensors are accoutered at the back side of the chair. The chair should be a flexible office chair. We are considering to equip sensors on the back of the flexible chair, where data will be passed to our computer through data cable. As discussed earlier MPU-6050 sensor is connected with the arduino at mega 2560. We are now using MPU 6050 as a combined unit of accelerometer and gyroscope sensor to detect the movement of the chair based on the movement of the user. We have to collect the data of Accelerometer and Gyroscope for the people with different weights and ages for rotating, tilting and bouncing. We will be first using data cables or burner cables to pass the data and later on based on other measurements and accuracy we will collect data by using Bluetooth sensor. We have used thresholding to detect 3 gestures (tilt backward, rotate right, rotate left).We have got threshold by analyzing the data of the users. Then we have also analyzed thresholding and machine learning to find which suits best in the system.

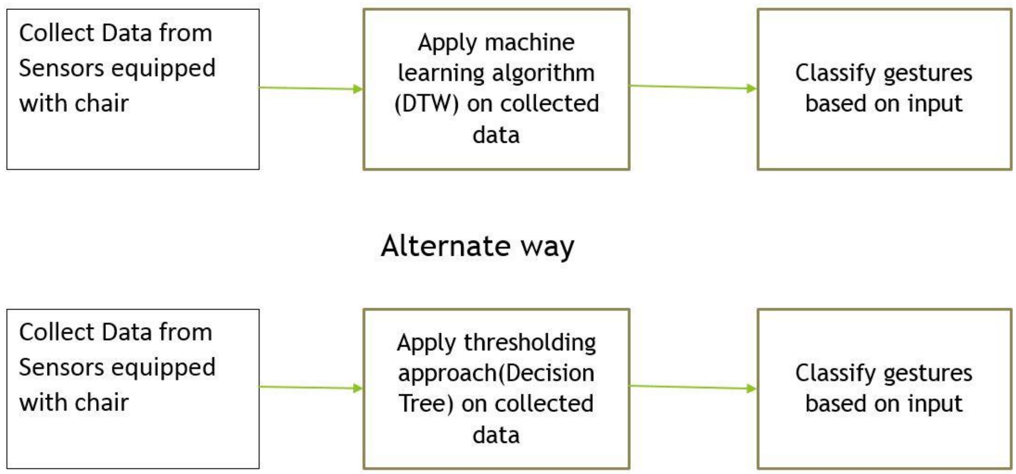

Fig: Work Flow of how machine learning algorithm will work

Figure: Proposed Interactive Gesture Chair

5.2 System Designed:

Most of the time we used to operate PC sitting on chair .So chair is so much related to computer activities and chair can be found everywhere. We can use the movements of a chair to communicate with PC. As much as movements we can define, we will be able to control more application. So the higher degree of freedom the chair have more gesture can be defined, more application can be controlled. Also gestures by chair movement is much easy because we operate PC by sitting on chair, as we ourselves is sitting on chair , and we are close to PC, so if we control PC by chair gesture it will be easy to implement and easy to set up in a new system. Based on our proposed architecture of the whole system we thought of buying a flexible office chair and arduino board and MPU 6050. We bought a flexible high quality office chair which has 3 degree of freedom, rotate right, and rotate left, tilt backward. Additionally we can use bounce movements as gesture but it is more difficult to perform for the user. An MPU-6050 sensor is accoutered with the chair to determine the movement of the chair according to the body movement of the user. The MPU 6050 communicates with the Arduino through the I2C protocol. The MPU 6050 is connected to Arduino as shown in the following diagram. A burner cable is connected to arduino to pass the data of Accelerometer and Gyroscope to PC.

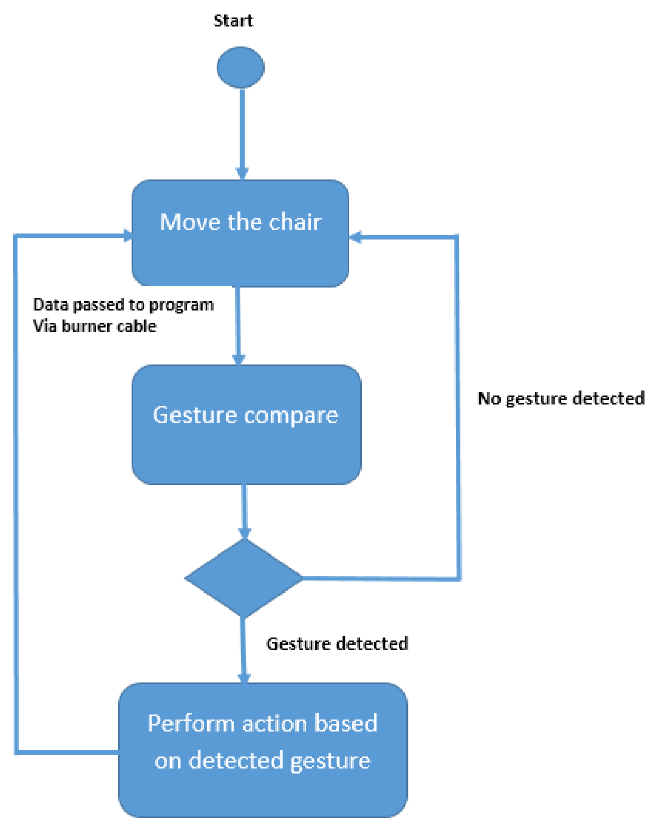

Fig: work flow of whole system

Fig: Chair and sensor implementations

Fig: Rotate right gesture

Fig: Rotate left Gesture

Fig: Back Gesture

We have collected the data of Accelerometer and Gyroscope for the people with different weights and ages for rotating, tilting and bouncing.

We have gone through four approaches to define gestures which are described below:

5.2.1 Euclidean distance:

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" (i.e. straight-line) distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric. A generalized term for the Euclidean norm is the L2 norm or L2 distance.

The Euclidean distance between points p and q is the length of the line segment connecting them (pq).In Cartesian coordinates, if p = (p1, p2,..., pn) and q = (q1, q2,..., qn) are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by the Pythagorean formula:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

The position of a point in a Euclidean n-space is a Euclidean vector. So, p and q are Euclidean vectors, starting from the origin of the space, and their tips indicate two points. The Euclidean norm, or Euclidean length, or magnitude of a vector measures the length of the vector:

$$\|\mathbf{p}\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{\mathbf{p}\cdot\mathbf{p}},$$

Where the last equation involves the dot product.

A vector can be described as a directed line segment from the origin of the Euclidean space (vector tail), to a point in that space (vector tip). If we consider that its length is actually the distance from its tail to its tip, it becomes clear that the Euclidean norm of a vector is just a special case of Euclidean distance: the Euclidean distance between its tail and its tip.

The distance between points p and q may have a direction (e.g. from p to q), so it may be represented by another vector, given by

$$\mathbf{q} - \mathbf{p} = (q_1-p_1, q_2-p_2, \cdots, q_n-p_n)$$

In a three-dimensional space (n=3), this is an arrow from p to q, which can be also regarded as the position of q relative to p. It may be also called a displacement vector if p and q represent two positions of the same point at two successive instants of time.

The Euclidean distance between p and q is just the Euclidean length of this distance (or displacement) vector:

$$\|\mathbf{q}-\mathbf{p}\| = \sqrt{(\mathbf{q}-\mathbf{p})\cdot(\mathbf{q}-\mathbf{p})}.$$

Which is equivalent to equation 1, and also to:

$$\|\mathbf{q}-\mathbf{p}\| = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p}\cdot\mathbf{q}}.$$

5.2.1.1 One dimension:

One In one dimension, the distance between two points on the real line is the absolute value of their numerical difference. Thus if x and y are two points on the real line, then the distance between them is given by:

$$\sqrt{(x-y)^2} = |x-y|.$$

In one dimension, there is a single homogeneous, translation-invariant metric (in other words, a distance that is induced by a norm), up to a scale factor of length, which is the Euclidean distance. In higher dimensions there are other possible norms.

5.2.1.2 Two dimensions:

In the Euclidean plane, if p = (p1, p2) and q = (q1, q2) then the distance is given by

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2}.$$

This is equivalent to the Pythagorean Theorem.

Alternatively, it follows from (2) that if the polar coordinates of the point p are (r1, θ1) and those of q are (r2, θ2), then the distance between the points is

$$\sqrt{r_1^2 + r_2^2 - 2r_1r_2\cos(\theta_1-\theta_2)}.$$

But we have used six dimensional Euclidean distance on the axis's of accelerometer(ax,ay,az) and gyroscope(gx,gy,gz) .So the distance between template(t) and sample(s) is

$$D(s,t)=\sqrt{(axs-axt)^2+(ays-ayt)^2+(azs-azt)^2+(gxs-gxt)^2+(gys-gyt)^2+(gzs-gzt)^2}$$

5.2.1.3 Limitations:

Where there is a high noise-to-signal ratio and negative spikes, any correlation is difficult to establish. The Euclidean distance method also suffers in such cases.

Euclidean distance measures the correlation between quantitative, continuous variables. It is not suitable for ordinal data, where preferences are listed according to rank instead of according to actual values.

It can't determine the correlation between user profiles who have similar trends in tastes, but different ratings for some of the same items. A method like the Pearson correlation would give an indication of how similar a set of preferences are, regardless of fluctuations in individual ratings.

5.2.2 DTW algorithm:

In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data which can be turned into a linear sequence can be analyzed with DTW. A well-known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

5.2.2.1 Implementation:

This example illustrates the implementation of the dynamic time warping algorithm when the two sequences s and t are strings of discrete symbols. For two symbols x and y, d(x, y) is a distance between the symbols, e.g. d(x, y) = | x - y |

```
int DTWDistance(s: array [1..n], t: array [1..m]) {

    DTW := array [0..n, 0..m]

    for i := 1 to n

        DTW[i, 0] := infinity

    for i := 1 to m

        DTW[0, i] := infinity

    DTW[0, 0] := 0

    for i := 1 to n

        for j := 1 to m

            cost:= d(s[i], t[j])

            DTW[i, j] := cost + minimum(DTW[i-1, j   ],    // insertion

                                        DTW[i   , j-1],    // deletion

                                        DTW[i-1, j-1])     // match

    return DTW[n, m]

}
```

We sometimes want to add a locality constraint. That is, we require that if s[i] is matched with t[j], then | i - j | is no larger than w, a window parameter.

We can easily modify the above algorithm to add a locality constraint (differences marked in bold italic). However, the above given modification works only if | n - m | is no larger than w, i.e. the end point is within the window length from diagonal. In order to make the algorithm work, the window parameter w must be adapted so that | n - m | ≤ w (see the line marked with (*) in the code).

```
int DTWDistance(s: array [1..n], t: array [1..m], w: int) {
    DTW := array [0..n, 0..m]
    w := max(w, abs(n-m)) // adapt window size (*)
    for i := 0 to n
        for j:= 0 to m
            DTW[i, j] := infinity
    DTW[0, 0] := 0
    for i := 1 to n
        for j := max(1, i-w) to min(m, i+w)
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i, j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
```

5.2.3 Decision tree:

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules. In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.

A decision tree consists of 3 types of nodes:

1. Decision nodes - commonly represented by squares
2. Chance nodes - represented by circles
3. End nodes - represented by triangles

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. If in practice decisions have to be taken online with no recall under incomplete knowledge, a decision tree should be paralleled by a probability model as a best choice model or online selection model algorithm. Another use of decision trees is as a descriptive means for calculating conditional probabilities.

Decision trees, influence diagrams, utility functions, and other decision analysis tools and methods are taught to undergraduate students in schools of business, health economics, and public health, and are examples of operations research or management science methods.

5.2.3.1 Advantages and disadvantages:

Among decision support tools, decision trees (and influence diagrams) have several advantages. Decision trees:

· Are simple to understand and interpret. People are able to understand decision tree models after a brief explanation.

· Have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes.

· Allow the addition of new possible scenarios

· Help determine worst, best and expected values for different scenarios

· Use a white box model. If a given result is provided by a model.

· Can be combined with other decision techniques.

5.2.3.2 Limitations of decision trees:

· For data including categorical variables with different number of levels, information gain in decision trees are biased in favor of those attributes with more levels.

· Calculations can get very complex particularly if many values are uncertain and/or if many outcomes are linked.

5.2.4 Thresholding based Approach:

```
Accelerometer X :208  Y:-16132  Z:-3952  Gyroscope X: -1809  Y:452  Z:  -310
Accelerometer X :192  Y:-16040  Z:-3948  Gyroscope X: -1814  Y:459  Z:  -313
Accelerometer X :68  Y:-16116  Z:-3940  Gyroscope X: -1836  Y:477  Z:  -297
Accelerometer X :176  Y:-16020  Z:-3952  Gyroscope X: -1825  Y:478  Z:  -328
Accelerometer X :220  Y:-16064  Z:-3932  Gyroscope X: -1830  Y:481  Z:  -319
Accelerometer X :200  Y:-16132  Z:-4068  Gyroscope X: -1804  Y:474  Z:  -309
Accelerometer X :196  Y:-16000  Z:-3952  Gyroscope X: -1827  Y:467  Z:  -331
Accelerometer X :192  Y:-16132  Z:-4060  Gyroscope X: -1797  Y:452  Z:  -285
Accelerometer X :296  Y:-16152  Z:-3996  Gyroscope X: -1796  Y:471  Z:  -315
Accelerometer X :216  Y:-16216  Z:-3920  Gyroscope X: -1806  Y:475  Z:  -263
Accelerometer X :248  Y:-16204  Z:-3752  Gyroscope X: -1828  Y:456  Z:  -285
Accelerometer X :260  Y:-16272  Z:-3912  Gyroscope X: -1842  Y:455  Z:  -283
Accelerometer X :224  Y:-16168  Z:-3868  Gyroscope X: -1883  Y:455  Z:  -279
```

Here we can see the raw data collected from the accelerometer and gyroscope and we saved the data in text file like this way, here we clearly isolated the data axes and the data type whether accelerometer or gyroscope data by putting spaces.In threshold based approach we took value of the accelerometer and gyroscope .Observed the values and analyze them, we saw values differs in which axis mainly then identified them for the respective gestures. After that we gradually put the threshold on the data one by one axes, because applying all the thresholds at once may create troubles in determining and evaluating the decision statements in respective axes, we mainly saw differences in gyroscope axis values, so we first set threshold on gyroscope axes values and later gradually put threshold on the accelerometer axes values. The arduino code where we set preliminary thresholds are shown below:

```
if(az>-500 && gx>0)
{
   Serial.println();
  Serial.println("back");
  delay(1000);
}
else if(ax>1500 && gy>7000 && gz>1000)
{
    Serial.println();
    Serial.print("right");
    delay(2000);
}
else if(ax>1000 && gy<0 && gz<-1000)
{
    Serial.println();
    Serial.print("left");
    delay(2000);
}
```

Fig: Thresholding of data on the axes values

6 Algorithm Analysis:

6.1 Euclidian distance based Approach:

```
{

  dtw(n1,tright_[i]);
 // dtw(n1,temp);
     tright_distances[i]=warpingDistance;
}

double tright_min=findmin(tright_distances);
Serial.print("   minimum:");
Serial.print(tright_min);
for(int i=0;i<6;i++)
{

  dtw(n1,tleft_[i]);
 // dtw(n1,temp);
     tleft_distances[i]=warpingDistance;
}

double tleft_min=findmin(tleft_distances);
Serial.print("   minimum:");
Serial.print(tleft_min);
```

In Euclidean based approach we take the values of six axes then manually select the best matching template and then manually compare the new values to the templates axes by axes, the matching procedure is based on six axes data. Euclidean distance gives us a good precision of data, so we primarily thought that Euclidean distance is best based on data analysis before we have seen the dtw algorithm. Manually calculated template matrix are shown here

```
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
double back_[6][6]={
{-764, -16356, -3268, -1698, 5169, 608},
{-714, -16144, -3818, -1581, 5515, 633},
{-714, -16124, -3818, -1581, 5515, 633},
{-800, -16900, -3333, -1699, 4900, 720},
{-714, -16154, -3818, -1581, 5515, 633},
{-714, -16114, -3818, -1581, 5515, 633},
};
```

```
double tright_[6][6]={
{-764, -16356, -3268, -1698, 5169, 608},
{-714, -16144, -3818, -1581, 5515, 633},
{-714, -16124, -3818, -1581, 5515, 633},
{-800, -16900, -3333, -1699, 5000, 720},
{-714, -16154, -3818, -1581, 5515, 633},
{-714, -16114, -3848, -1581, 5515, 633},
};
```

```
double tleft_[6][6]={
{-764, -16356, -3268, -1698, 5169, 608},
{-714, -16144, -3818, -1581, 5515, 633},
{-714, -16124, -3818, -1581, 5515, 633},
{-800, -16900, -3333, -1699, 4700, 720},
{-714, -16154, -3818, -1581, 5515, 633},
{-714, -16114, -3818, -1581, 5515, 633},
};
```

```
double steady_[6][6]={
{-764, -16356, -3268, -1698, 5169, 608},
{-714, -16144, -3818, -1581, 5515, 633},
{-714, -16124, -3818, -1581, 5515, 633},
{-800, -16900, -3333, -1699, 4900, 720},
{-714, -16154, -3818, -1581, 5515, 633},
{-714, -16114, -3818, -1581, 5505, 633},
};
```

Fig: calculated matrix of the templates

6.2 DTW Based Approach:

DTW is manly a dynamic time warping algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data which can be turned into a linear sequence can be analyzed with DTW. The limitations of Euclidean and advantages of the DFTW can be seen from the below figure.

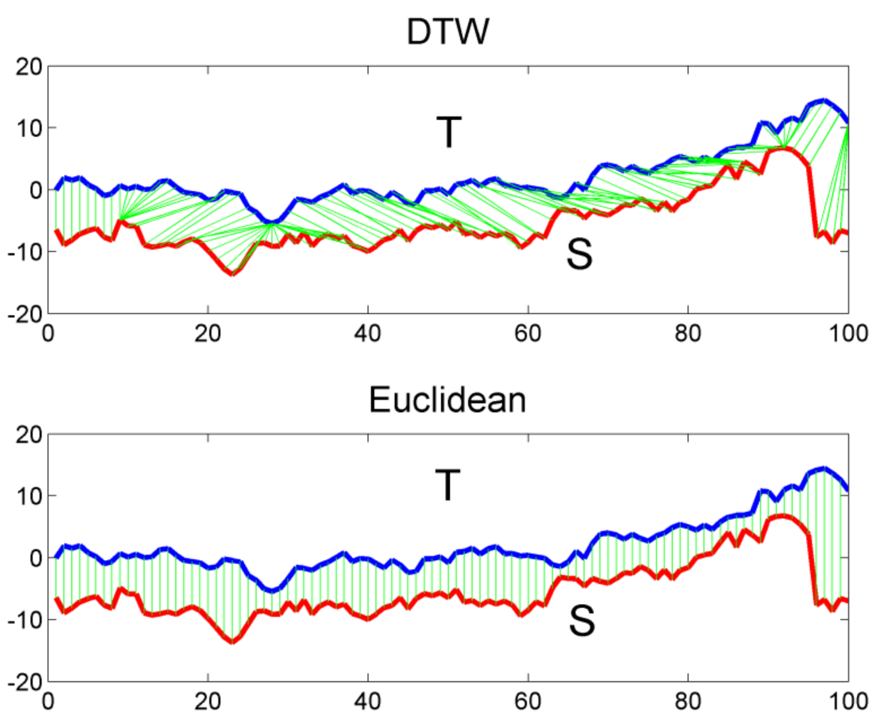

Fig: DTW algorithm and Euclidean comparison

```
warpingPath[K - 1][0] = i;
warpingPath[K - 1][1] = j;

while ((i + j) != 0) {
        if (i == 0) {
                j -= 1;
        } else if (j == 0) {
                i -= 1;
        } else {        // i != 0 && j != 0
                double array[3] = { D[i - 1][j], D[i][j - 1], D[i - 1][j - 1] };
                minIndex = getIndexOfMinimum(array);

                if (minIndex == 0) {
                        i -= 1;
                } else if (minIndex == 1) {
                        j -= 1;
                } else if (minIndex == 2) {
                        i -= 1;
                        j -= 1;
                }
        } // end else
        K++;
        warpingPath[K - 1][0] = i;
```

Fig: Code snapshot of DTW algorithm

After finding several flaws of Euclidean distance like not considering the points of different peaks and not considering the shape of the signal, we took decision of stepping out to dtw algorithm, then afterwards we followed window based dtw algorithm Implemented in arduino which identifies the vector class in real time while we are processing data, the code snapshot is provided, the dtw gave us the class level for each of the data and then we saved the data on the CSV file which is a standard comma separated value file for keeping this kind of data. The CSV file we get the 6 dimensional data and The decision criteria or the decision value of steady or right or left value. The sample output of CSV data is shown below.

Ax	Ay	Az	Gx	Gy	Gz	Decision
-212	-16664	-632	317	451	-355	steady
140	-16816	-344	206	594	-324	back
4	-15796	-3912	-5225	224	-415	steady
-12	-15732	-4000	-5133	229	-408	steady
36	-14756	-4372	-5054	48	-466	steady
192	-15772	24	-44	429	-310	steady
328	-15840	196	70	426	-333	back
280	-16164	2988	-2473	451	-264	steady
224	-16180	2940	-2483	437	-279	steady
216	-16388	2936	-2485	440	-302	steady
224	-16368	2788	-2555	489	-264	steady
180	-16220	-660	507	221	-303	steady
240	-16236	-652	510	255	-299	steady
204	-16288	-480	561	256	-316	back
268	-16204	2856	-2062	492	-282	steady
192	-16172	2796	-2083	484	-290	steady
240	-16288	-556	601	377	-300	steady
304	-16196	-452	596	366	-332	back

6.3 Decision Tree Based Approach:

After seeing the data analysis based on the Euclidean and dtw algorithm we decide to make a decision tree based on the class labels identified on the dtw algorithm and thresholding based approach. We put the output CSV files in the weka software and choose the ID3 decision tree classifier (named as J48) in the software and then got a graph, but from the graph , decisions were not clear how it was chosen , so we selected the visualize tree option , which shows us the below tree which clearly identifies the appropriate threshold , here is the uniqueness of our thesis work, the previous works don’t actually used the decision tree algorithm to find effective threshold , decision tree gives an well approximated threshold that works best with lots of people it also helps us to give a universal threshold which works for a variety of people more flawlessly. Though the success depends on the amount of data provides but we put some effective data which have some variations in them to get better results and then after training the decision tree algorithm with less amount of data , it shows us better performance in choosing threshold which we couldn’t achieve via manual thresholding and it also better that machine learning approach and manual thresholding of the data axes , in the tree we can see that data mainly depends on gyroscope values , especially on the Gy value then the depends on Gx values and later on chooses accelerometer values of making thresholds ,. After analyzing by decision tree we identified that decision tree gives us the actual thresholds which can be used to find changes on the data based on the respective gesture.

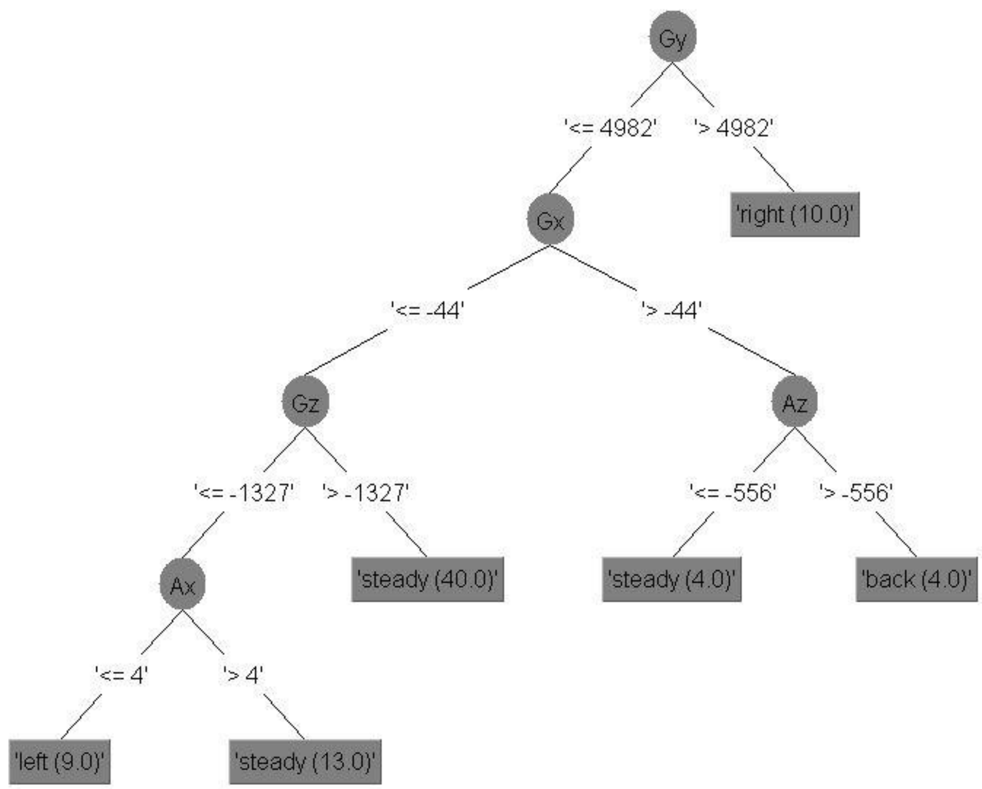

Fig: Decision tree

7 Result Analysis:

7.1 Threshold Based:

Task name	Used gesture	Error(out of 100)	Time(sec)
Pause/play	Tilt backward	15	178
Play next	Rotate right	12	261
Play previous	Rotate left	13	287

We took data of different users, we took data of approximately 10 persons then, inspected the performance of the data by calculating the accuracy, for each person we took rotate right, rotate left, steady and bounce operations and observed how much accurately the system can detect their movements, then we did some calculations which considers the time of inputs, number of detected gestures and number of persons.

Accuracy = ((300-(15+12+13))/300)*100%=86.67% cc

Average time to perform a gesture = (178+261+287)/300=2.42 seconds.

7.2 Euclidian Distance Based:

Task name	Used gesture	Error(out of 100)	Time(sec)
Pause/play	Tilt backward	11	171
Play next	Rotate right	6	263
Play previous	Rotate left	4	279

After thresholding we chose Euclidean based approach we take the values of six axes then manually select the best matching template and then manually compare the new values to the templates axes by axes, the matching procedure is based on six axes data. Euclidean distance gives us a good precision of data, so we primarily thought that Euclidean distance is best based on data analysis before we have seen the dtw algorithm. In the same way we collected data for 10 persons calculated the gesture perform time and number of persons and number of gestures correctly detected and number of gestures wrongly detected. So we considered all this and calculated the total accuracy of the Euclidean algorithm.

Accuracy = ((300-(11+6+4))/300)*100%=93%

Average time to perform a gesture = (171+263+279)/300=2.38 seconds

7.3 DTW Based:

Task name	Used gesture	Error(out of 100)	Time(sec)
Pause/play	Tilt backward	8	242
Play next	Rotate right	7	257
Play previous	Rotate left	4	223

Again we saw some limitations of Euclidean distance like not considering the points of different peaks and not considering the shape of the signal, we took decision of stepping out to dtw algorithm, then afterwards we followed window based dtw algorithm Implemented in arduino which identifies the vector class in real time while we are processing data, the code snapshot is provided, the dtw gave us the class level for each of the data and then we saved the data on the CSV file. In the same way we collected data for 10 persons calculated the gesture perform time and number of persons and number of gestures correctly detected and number of gestures wrongly detected. So we considered all this and calculated the total accuracy of the DTW algorithm.

Accuracy = ((300-(8+7+4))/300)*100%=93.67%

Average time to perform a gesture = (242+257+223)/300=2.41 seconds

7.4 Decision Tree Based

Task name	Used gesture	Error(out of 100)	Time(sec)
Pause/play	Tilt backward	13	235
Play next	Rotate right	9	252
Play previous	Rotate left	11	219

Here is the uniqueness of our thesis work, the previous works don't actually used the decision tree algorithm to find effective threshold , decision tree gives an well approximated threshold that works best with lots of people it also helps us to give a universal threshold which works for a variety of people more flawlessly. Though the success depends on the amount of data provides but we put some effective data which have some variations in them to get better results and then after training the decision tree algorithm with less amount of data, it shows us better performance in choosing threshold which we couldn't achieve via manual thresholding and it also better that machine learning approach and manual thresholding of the data axes. Like previous way we collected data for 10 persons calculated the gesture perform time and number of persons and number of gestures correctly detected and number of gestures wrongly detected. So we considered all this and calculated the total accuracy of the DTW algorithm

Accuracy = ((300-(13+9+11))/300)*100%=89%

Average time to perform a gesture = (235+252+219)/300=2.35 seconds

8 Conclusion & Future Work:

In this paper, we presented the iterative design process of a novel input technique based on gestural interaction with a sensor-equipped flexible office chair. A basic set of semaphoric chair gestures was defined, which can be mapped to specific functions for controlling applications on a desktop computer. We explored the application of these chair gestures as additional input modality for focused interaction with a web browser, and peripheral interaction with a music player. Corresponding experimental results indicate overall positive user feedback, and comparable task performance as with more familiar keyboard or touch input.

The embodied aspects of chair-based input [7] seemed to facilitate interaction, which indicates high potential to enrich the computing experience and capture the interest of a larger audience. However, as with other gestural interfaces, there are some challenges when designing for chair-based gestural interaction [21,24,31], such as distinguishing between natural behavior and gestural input, revealing the gestural input scope, creating understandable metaphors for actions invoked by the gestural input, or providing reliable real-time system feedback [23]. Further, ergonomic aspects regarding the effects of physical body movements to perform gestures with a flexible chair will have to be taken into account for future design of gestures and applications.

Summarizing, as for any other input modality, there are always trade-offs: While input time and physical effort may be higher for chair gestures than traditional hand-mediated input devices, this must be balanced against the benefits of always-available, eyes-free, hands-free operation that an interactive chair can provide in a computing environment. Based on these unique features, chair gestures seem highly promising for opportunistic interaction in a desktop environment. Particularly suitable in the context of peripheral interaction, interactive chairs could therefore serve as platform for a whole range of applications that enable users to quickly issue commands to an application and rapidly return to their other ongoing activities. In addition, the introduction of technologies integrating the body into our interactions is a strategy with great potential to avoid physical inactivity [7]. Chair gestures are certainly not a panacea for every potential input problem faced by end users. Still, the gestures can serve a valuable purpose as alternative input modality in situations that afford eyes- or hands-free interaction, or when we simply want to break up the monotony of the mouse-and-keyboard desktop paradigm.

Future work will focus on solving practical challenges that arise with gestural chair interaction, including improvement of recognition robustness, further exploration of the gestural input scope, and definition of suitable applications that integrate chair gestures into the computing experience. Therefore, we will investigate the application of gestural chair input for other usage scenarios besides the proposed web browser and music player control (e.g., application launching, task management, notification handling). Finally, as for a novel input technique it is always the question of longterm acceptance and use, we plan to conduct a long-term insitu deployment to further explore the integration of semaphoric chair gestures into the computing experience.

9 References:

1. K. Probst, D. Lindlbauer, P. Greindl, M. Trapp, M. Haller, B. Schwartz, and A. Schrempf, 2013."Rotating, Tilting, Bouncing: Using an Interactive Chair to Promote Activity in Office Environments," in Proceedings of the 31st international conference extended abstracts on Human factors in computing systems, Paris, France, 2013, pp. 79-84.
2. K. Probst, D. Lindlbauer, M. Haller, B. Schwartz, and A. Schrempf, 2014."A Chair as Ubiquitous Input Device: Exploring Semaphoric Chair Gestures for Focused and Peripheral Interaction" in CHI14: Proceedings of the 32nd International Conference on Human Factors in Computing Systems, Toronto, ON, Canada, 2014.
3. K. Probst, D. Lindlbauer, M. Haller, B. Schwartz, and A. Schrempf, 2014."Exploring the Potential of Peripheral Interaction through Smart Furniture" in Peripheral

Interaction: Shaping the Research and Design Space, Workshop at CHI 2014, Toronto, ON, Canada, 2014.

4. Anttonen, J. and Surakka, V. Music, Heart Rate, and Emotions in the Context of Stimulating Technologies. ACII 2007, 290–301.
5. Baudel, T. and Beaudouin-Lafon, M. Charade: Remote Control of Objects Using Free-Hand Gestures. Communications of the ACM 36, 7 (1993), 28–35.
6. Beckhaus, S., Blom, K., and Haringer, M. ChairIO – The Chair-Based Interface. In Concepts and Technologies for Pervasive Games. 2007, 231–264
7. Van Beurden, M., Ijsselsteijn, W., and De Kort, Y. User Experience of Gesture Based Interfaces: A Comparison with Traditional Interaction Methods on Pragmatic and Hedonic Qualities. GW 2011, 36–47
8. Card, S., Moran, T., and Newel, A. The KeystrokeLevel Model for User Performance Time with Interactive Systems. Communications of the ACM 23, 7 (1980), 396–410
9. Cohen, M. The Internet Chair. International Journal of Human Computer Interaction 15, 2 (2003), 56–57.
10. Dourish, P. Where The Action Is. MIT Press, 2001
11. Endert, A., Fiaux, P., Chung, H., Stewart, M., Andrews, C., and North, C. ChairMouse: Leveraging Natural Chair Rotation for Cursor Navigation on Large, HighResolution Displays. Ext. Abstracts CHI 2011, 571–580.
12. Forlizzi, J., DiSalvo, C., Zimmerman, J., Mutlu, B., and Hurst, A. The SenseChair: The Lounge Chair as an Intelligent Assistive Device for Elders. DUX 2005, 31
13. Hart, S. and Staveland, L. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Human Mental Workload. 1988, 139–183.
14. Hausen, D., Bakker, S., Van den Hoven, E., Butz, A., and Eggen, B. Peripheral Interaction: Embedding HCI in Everyday Life. Workshop at INTERACT 2013.
15. Hausen, D., Boring, S., and Greenberg, S. The Unadorned Desk: Exploiting the Physical Space around a Display as an Input Canvas. INTERACT 2013, 140–158.

16. Hausen, D., Richter, H., Hemme, A., and Butz, A. Comparing Input Modalities for Peripheral Interaction: A Case Study on Peripheral Music Control. INTERACT 2013, 162–179.
17. Hausen, D. Comparing Modalities and Feedback for Peripheral Interaction. Ext. Abstracts CHI 2013, 1263–1268.
18. Healy, G., Lawler, S., Thorp, A., Neuhaus, M., Robson, E., Owen, N., and Dunstan, D. Reducing Prolonged Sitting in the Workplace. Human Factors and Ergonomics Society, 2012.
19. Karam, M. and schraefel, m.c. A Study on the Use of Semaphoric Gestures to Support Secondary Task Interactions. Ext. Abstracts CHI 2005, 1961–1964.
20. Karam, M. and schraefel, m.c. A Taxonomy of Gestures in Human Computer Interaction. 2005.
21. Levine, J. a. Non-Exercise Activity Thermogenesis (NEAT). Best Practice & Research Clinical Endocrinology & Metabolism 16, 4 (2002), 679–702.
22. Mankoff, J., Hudson, S., and Abowd, G. Interaction Techniques for Ambiguity Resolution in Recognitionbased Interfaces. UIST 2000, 11–20.
23. Montero, C. and Marshall, M. Would You Do That? – Understanding Social Acceptance of Gestural Interfaces. MobileHCI 2010, 275–278.
24. Norman, D. Natural User Interfaces Are Not Natural. Interactions 17, 3 (2010), 6–10.
25. Oakley, I. and Park, J. Designing Eyes-Free Interaction. In Haptic & Audio Interaction Design. 2007, 121–132.
26. Probst, K., Lindlbauer, D., Greindl, P., Trapp, M., Haller, M., Schwartz, B., and Schrempf, A. Rotating, Tilting, Bouncing: Using an Interactive Chair to Promote Activity in Office Environments. Ext. Abstracts CHI 2013, 79–84.
27. Quek, F., McNeill, D., Bryll, R., and McCullough, K. Multimodal Human Discourse: Gesture and Speech. Transactions on Computer-Human Interaction 9, 3 (2002), 171–193.

28. Shen, X. An Evaluation Methodology for Ambient Displays. Journal of Engineering, Computing and Architecture 1, 2 (2007).
29. Springer, T. The Future of Ergonomic Office Seating. Knoll Workplace Research, 2010.
30. Streitz, N., Geißler, J., Holmer, T., Konomi, S., MüllerTomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-LAND: An Interactive Landscape for Creativity and Innovation. CHI 1999, 120–127.
31. Tan, H., Slivovsky, L., and Pentland, A. A Sensing Chair Using Pressure Distribution Sensors. Transactions on Mechatronics 6, 3 (2001), 261–268.
32. Van Uffelen, J., Wong, J., Chau, J., van der Ploeg, H., Riphagen, I., Gilson, N., Burton, N., Healy, G., Thorp, A., Clark, B., Gardiner, P., Dunstan, D., Bauman, A., Owen, N., and Brown, W. Occupational Sitting and Health Risks: A Systematic Review. American Journal of Preventive Medicine 39, 4 (2010), 379–88.
33. Weiser, M. The Computer for the 21st Century. Scientific American 265, 3 (1991), 66–75.
34. Wexelblat, A. Research Challenges in Gesture: Open Issues and Unsolved Problems. GW 1997, 1–11.
35. Wobbrock, J., Aung, H., Rothrock, B., and Myers, B. Maximizing the Guessability of Symbolic Input. Ext. Abstracts CHI 2005, 1869–1872.
36. Zheng, Y. and Morell, J. A Vibrotactile Feedback Approach to Posture Guidance. HAPTIC 2010, 351–358.
37. Bailey, B., Konstan, J., and Carlis, J. The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. INTERACT 2000, 593–601.
38. Hudson, S.E., Harrison, C., and Harrison, B. Whack Gestures: Inexact and Inattentive Interaction with Mobile Devices. TEI 2010, 109–112.
39. Pohl, H. and Murray-Smith, R. Focused and Casual Interactions: Allowing Users to Vary Their Level of Engagement. CHI 2013, 2223–2232.

40. Weiser, M. and Brown, J.S. The Coming Age of Calm Technology. In Beyond Calculation: The Next Fifty Years of Computing. 1997, 75–85.
41. Alpern, M. and Minardo, K. Developing a Car Gesture Interface For Use as a Secondary Task. In Proc. CHI EA ’03, ACM Press (2003), 932–933.
42. Daian, I., Van Ruiten, A.M., Visser, A., and Zubić, S. Sensitive Chair: A Force Sensing Chair with Multimodal Real-Time Feedback via Agent. In Proc. ECCE ’07, ACM Press (2007), 163–166.
43. Jacob, R.J.K. Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces. In Advances in Human-Computer Interaction (1993), 151–190.
44. Kapoor, A., Mota, S., and Picard, R.W. Towards a Learning Companion that Recognizes Affect. In Proc. Emotional and Intelligent II, AAAI Press (2001).
45. MacKay, B., Dearman, D., Inkpen, K., and Watters, C. Walk’n Scroll: A Comparison of Software-based Navigation Techniques for Different Levels of Mobility. In Proc. MobileHCI ’05, ACM Press (2005), 183–190.
46. Nanayakkara, S., Wyse, L., and Taylor, E.A. Effectiveness of the haptic chair in speech training. In Proc. ASSETS ’12, ACM Press (2012), 235–236.
47. Owen, N., Bauman, A., and Brown, W. Too much sitting: a novel and important predictor of chronic disease risk? In British Journal of Sports Medicine 43, 2 (2009), 81–83.
48. Probst, K., Leitner, J., Perteneder, F., Haller, M., Schrempf, A., and Glöckl, J. Active Office: Towards an Activity-Promoting Office Workplace Design. In Proc. CHI EA ’12, ACM Press (2012), 2165–2170.
49. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K.E., and Ansari, R. Multimodal Human Discourse: Gesture and Speech. In ACM Transactions on Computer-Human Interaction 9, 3 (2002), 171–193.
50. Saffer, D. Designing Gestural Interfaces. O’Reilly Media, 2008.

51. Schrempf, A., Schossleitner, G., Minarik, T., Haller, M., Gross, S., and Kurschl, W. PostureCare - Towards a Novel System for Posture Monitoring and Guidance. In Proc. IFAC '11, Elsevier (2011), 593–598.
52. Vanhala, T., Surakka, V., and Anttonen, J. Measuring Bodily Responses to Virtual Faces with a Pressure Sensitive Chair. In Proc. NordiCHI '08, ACM Press (2008), 555–559.