HUMAN ACTIVITY TRACKING AND RECOGNITION USING KINECT SENSOR

ROANNA Z. LUN

Bachelor of Science in Mechanical Engineering
Tsinghua University
June 1985

Master of Science in Computer Science
Ohio University
December 1989

submitted in partial fulfillment of requirements for the degree
DOCTOR OF ENGINEERING in COMPUTER ENGINEERING
at the
CLEVELAND STATE UNIVERSITY
MAY 2018

We hereby approve the dissertation
of
Roanna Lun

Candidate for the Doctor of Engineering degree.
This dissertation has been approved for the Department of

Electrical and Computer Engineering

and CLEVELAND STATE UNIVERSITY
College of Graduate Studies by

_____

Wenbing Zhao, Dissertation Committee Chairperson – Department & Date

_____

Chansu Yu, Dissertation Committee Member – Department & Date

_____

Yongjian Fu, Dissertation Committee Member – Department & Date

_____

Lili Dong, Dissertation Committee Member – Department & Date

_____

Haodong Wang, Dissertation Committee Member – Department & Date

_____

Student's Date of Defense

_____

Chandra Kothapalli, Doctoral Program Director

To my family who made this possible.

ACKNOWLEDGMENTS

Pursuing Engineering Doctoral Degree is a long journey for me, often along with many uncertain paths. During this journal, I have experienced successes, disappointment, joys, prolonged process, and most important the self-satisfying of accomplishments.

This dissertation is the result of collaboration, discussion and support from many great people. My deepest gratitude goes to my adviser Dr. Wenbing Zhao for his endless support, encouragement and patience through years of studies. He provides many opportunities, guidelines and supervision for me to strengthen research experience, enhance computer programming skills, and most importantly to mature me to be a better person. I am very grateful to have Dr. Chansu Yu, Dr. Yongjian Fu, Dr. Lili Dong, and Dr. Haodong Wang as dissertation committee members, and their academic support and input, and personal cheering are greatly appreciated.

In addition, I would also like to thank Dr. Ye Zhu, Dr. Dan Simon, Dr. Xiong, Dr. Chansu Yu and other faculty and staff in Electrical Engineering and Computer Science department for providing me with a Doctoral Research Assistantships. Without these opportunities, I would not have completed the research work.

I would like to thank my husband, sons and daughter, my father and sister who always encourage me to pursue my dream, and always believe in me. I thank them for the sacrifices they have made. My sincere thanks to Dr. Hua Cai who was my lab-mate and classmate in many courses we took together, and she always is my go-to person even she has successfully graduated from Doctoral program. I would also like to thank my dear friends, Dr. Colleen Kelley, Dr. Yunkai Liu, and Dr. Lin Deng for motivating me through this journey and expressing their own support through my tenacity and diligence.

HUMAN ACTIVITY TRACKING AND RECOGNITION USING KINECT SENSOR

ROANNA LUN

ABSTRACT

The objective of this dissertation research is to use Kinect sensor, a motion sensing input device, to develop an integrated software system that can be used for tracking non-compliant activity postures of consented health-care workers for assisting the workers' compliance to best practices, allowing individualized gestures for privacy-aware user registration, movement recognition using rule-based algorithm, real-time feedback, and exercises data collection. The research work also includes developing a graphical user interface and data visualization program for illustrating statistical information for administrator, as well as utilizing cloud based database system used for data resource.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The computer vision-based human body movements tracking and analysis are fundamental principles in many applications in the areas of computer animation, graphic gaming industry, security surveillance, health care exercises, and retail industry. The population of aging baby boomer has rapidly grown in the United States according national census bureau report [3] that states that the population of over age 65 will be doubled in next 20 years. With such large pool of aging population, thereby the demands of health care services undoubtedly will grow enormously in the coming years. In order to meet the increasing demands of health care services, and also to reduce the cost of services, the medical providers nowadays are more often looking for the computer program and other equipment that can assist them for services provided to patients.

The approach to reduce health-care cost, which is to allow users to perform post-injure or post-surgery rehabilitation exercises at home or community-based non-medical facility, has been studied extensively. However assurance of quality the exercise and the correctness of movements is a big challenge. Incorrect body movement performed by patients may be harmful than healing. Certain exercises may require to perform in the exact way what the doctor instructs. Therefore using computer application that can track and monitor activities for the patients, especially for those who need physical therapy and rehabilitation care, people with disabilities, and the elderly,

becomes emergent.

The increasing cost of health care services is not only driven by the growth of aging population, but also driven by increasing compensation paid to injured health-care workers while they are taking care of patients [4] . The nurse assistants and nurses have the highest injury rates of all occupations examined according a report from United States Centers for Disease Control and Prevention [5]. In this report, collecting data identifies that handling, pushing, pulling and lifting patients [6], can be directed to contribute to occupational traumatic injuries. Furthermore it examines the fact that injures from patient handling, slipping, tripping, and falling are made up a substantial portion of all occupational injuries in the health care sector. The researchers have been looking for computer-aided system for preventing such occupational injures. Hence computer application that can tracking and monitoring activities for the health care providers certainly becomes desired.

The applications of body motion recognition, detection, and monitoring technology have expanded from traditional visual gaming realm to non-traditional consumers and enterprises markets according to the survey report published by California Institute of Technology [7]. The most intense demand perhaps is from the health care industry. The body motion recognition technology has been widely used by developing software application for physical therapy and rehabilitation exercises, monitoring patients' activities, and many other aspects in recent years. However existing motion analysis tools used in those applications are intrusive, for example, patients either have to wear body markers or attach inertial sensors. We are motivated to design and develop a system that is non-intrusive, convenient to use, inexpensive, and at meantime is able to monitor and analyze the human body movement in real time.

In our research study, a computer-vision based application for tracking and monitoring human body movements is developed for aiming at reducing health care cost. We envision an easy-use, non-intrusive, in-home computer application that can be used for monitoring the patient's activity and providing real-time feedback about any non-compliant movements could be one of feasible

solution. The system is required to be affordable, efficient, user friendly, and most importantly to be able to provide real-time feedback of movements for either users or medical professional providers.

The release of low-cost image tracking sensor Kinect by Microsoft in 2010 makes possible for developing affordable and interactive real-time human body movement tracking and analysis application. The advanced Kinect technology enables computer application to track, detect and recognize human body movement dynamically in real-time. Applications of Microsoft Kinect have been extended to many fields beyond video games, including security surveillance, health-care, human computer interaction, humanoid robots, sign languages, and other areas [8]. Furthermore, there has been intensively studying in algorithms, mechanisms, and techniques used for fundamental principles of human motion tracking and analysis utilizing Kinect sensor. Our work not only applies to monitor patients performing physical and rehabilitation exercises, but also provides secured and private identification for health-care givers, such as nurses, nurse assistants, and physical therapists.

During the work presented in this dissertation, we have major contributions shown as followings:

- conduct a profound survey on tracking and recognizing human body movement techniques and algorithms
- study different machine learning algorithms used for recognition human body movement
- design and develop user interface system that can protect person privacy on image captured by Kinect server
- design and develop an integrated application that can provide real-time feedback to user during course of performing exercises
- design and develop cloud-based data collection system
- design and develop graphical visualization system for illustrating statistical data of user's activities

First we conduct a profound survey on tracking and recognizing human body movement. The integrated system includes computer server, Kinect sensor, mobile device, wearable device and a cloud-based server. The system focuses on 1) qualification of data collected in real-time from the Kinect sensor, including skeleton frames, depth frames, and RGB camera frames, to achieve data unification regardless of the physical size of the users; 2) program settings to protect privacy of health care providers; 3) automated and interactive tracking of user body movement; 4) real-time detection of non-compliant body movements; 5) the performance of motion/gesture recognition process in feedback system; 6) logging information that can be reviewed locally or remotely

I.1 Significant Contributions

The significant contributions in developing the integrated human activity tracking and recognition using Kinect sensor are:

- tracking and monitoring human body movement using Kinect sensor
- privacy-aware user registration and identification protection
- real-time feedback through wearable device
- data collection on cloud based data system or local device
- data visualization program

I.2 Publications

During the period of pursuing degree of Doctor of Engineer, I have authored the publications shown in Appendix A. The highlights of publications are title of "**A Survey of applications and human motion recognition with Microsoft Kinect**" that has been cited 70 times since published in May 2015, and title of "**A Human-Centered Activity Tracking System: Toward a Healthier Workplace**" that is published on IEEE Transactions on Human-Machine Systems.

CHAPTER II

BACKGROUND AND RELATED WORK

In this chapter we provide background and related research work for major Kinect technology used our system. The focus is its current applications for tracking and monitoring human body movement. We also discuss the techniques and algorithms used in these applications. Furthermore the current development of protecting privacy of health care givers using Kinect sensor, how to provide real-time feedback, and the importances of privacy-aware identifying user are reviewed. The advantages of using cloud-based server for data analysis of logging information is also described.

II.1 Kinect Device

In November 2010, Microsoft Corporation launched a revolutionary gaming motion sensing input device Kinect 1 [18] shown in Figure 1, which is the first-generation of Kinect sensor. This game control device enables Xbox 360 game players to control and interact with console without touching a controller. It communicates simply with the game console by gesturing and speaking.

With enhanced quality, Microsoft released second generation of device - Kinect v2 [19] in 2014. The later sensor can detect as many six complete skeletons compared to two of the original sensor, and 25 body joints compared to 20 of the original sensor. The tracked positions are more

Figure 1: Microsoft Kinect Sensor: Kinect 1

accurate and stable and the range of tracking is wider. Improved design allows to see much smaller objects. And all objects are more clearly with 3D visualization. Even with intriguing features, this device is offered at a very affordable price. Kinect sensors are sold in the market ranging from $30 to $200. Being such low cost but reliable device has enabled software engineers to fully develop its application for everything from user-following retail shopping, detecting elderly falling, home-based security monitoring, to navigation systems for the blind, compared to using traditional more expensive, high performance and quality sensing devices.

In addition to advanced hardware design, Microsoft released Software Development Kit (SDK) for Kinect senor in late 2011 for free. With Kinect SDK, the computer programmer are able to develop sophisticated computer-based vision tracking applications. The applications allow users to control and interact with computers and video game console through human body gesture, motion, and speaking commands without peripheral equipment. The SDK supports C# and C++ programming language interface for users to develop computer-based human body and gesture tracking applications. The SDK programming model [20] enables the developers to access raw image data from Kinect sensor including an RGB camera and a depth sensor. Coupling with SDK, the Kinect sensor can capture human full body motion with three streams of image frames, color stream, depth stream and skeleton stream. The Figure 3 shows a depth image of joint tracking and skeleton image shown in Figure 2. With all intriguing features, Kinect sensor has attracted enormous interests in research and developments [8].

Figure 2: Kinect Depth Image

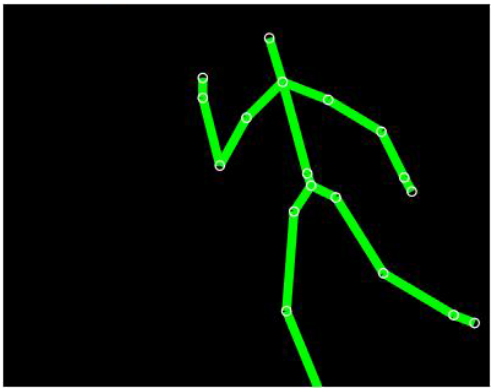

Figure 3: Kinect Skeleton Image

II.1.1 Advanced Features

Kinect sensor has many intriguing features. The wide usage and attribution of Kinect sensor is due to its advanced hardware design. At a very affordable price ranging from $30 - $100, it contains three major components, color VGA video camera, depth sensor, and multi-array microphone. The color camera collects red, green and black image data for facial recognition and other detection features. It supports a maximum resolution of 1280 x 960. The depth sensor is built with an infrared projector and a monochrome complimentary metal-oxide semiconductor sensor, which can produce maximum resolution of 640 x 480. These two pieces can work together to directly fetch the depth of objects, which is the third dimension in addition to X and Y positions regardless of the lighting condition. In addition to depth data, Kinect sensor API produces skeleton image that can transform depth image data to build the positions of various articulated joints (20 joints for Kinect v1 and 25 joints for Kinect v2) of the human body. With skeleton image combining with the raw depth image data, it make easier to track and monitor human body movements in real-time. An array of four microphones allows the user to interact with computer or game console with human voice. The user voices can be isolated from room noises, even very effective at a few feet way.

II.1.2 Original Design

As shown in Figure 1, Kinect sensor is initially developed for game console Xbox 360. It can detect a human gesture, body movement and voice. Due to its affordable price and robust performance, it has widely been used in other area. In order to capture human body movements, it utilizes three visual components, RGB color camera, infrared projector, and infrared camera. The depth image is used to generate human body skeleton data including 20 joints (Figure 3) without using any inertial sensors or markers.

Microsofts Kinect sensor system was originally designed and developed by PrimeSense technology. The product was developed by patented technology that makes 3D depth sensing image. It generates coded scene with near-IR light that is invisible to the human eye and a standard

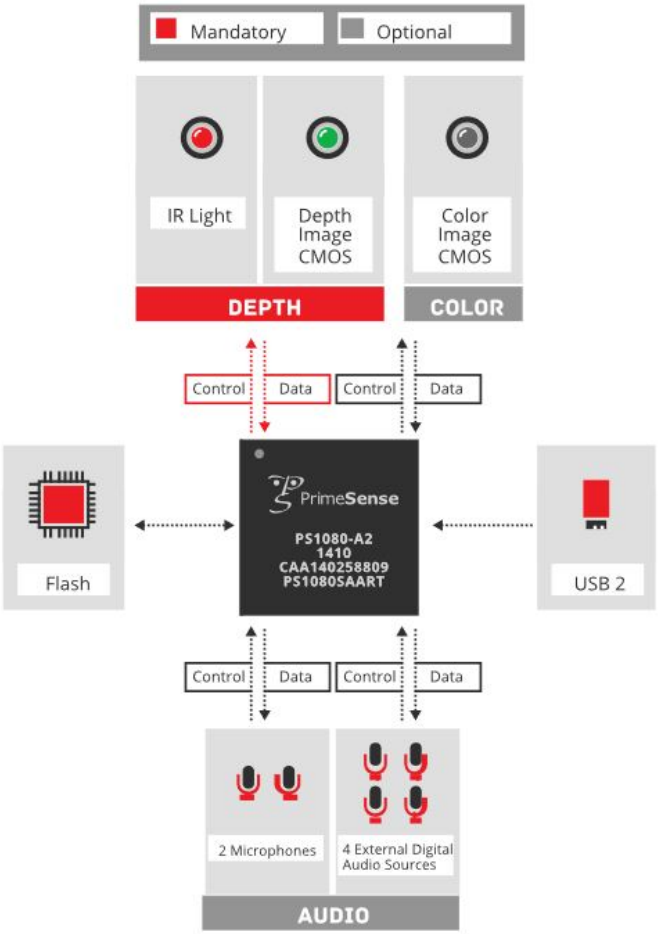

Figure 4: PrimeSense PS1080 Microcontroller Block Diagram (source from [1])

off-the-shelf CMOS sensor that can read the coded light back from the scene. This process enables building depth image that detects objects with high accuracy. A sample of block diagram of PrimeSense device is shown in Figure 4.

Kinect uses depth sensor produced by PrimeSense. However in the earlier technical documents from PrimeSense, how exactly it works is not obvious. People are speculating that PrimeSense sensor is using time-of-flight depth camera. In fact PrimeSense explicitly states they are not using time-of-flight, but something they call light coding. Later researcher found out that Kinect

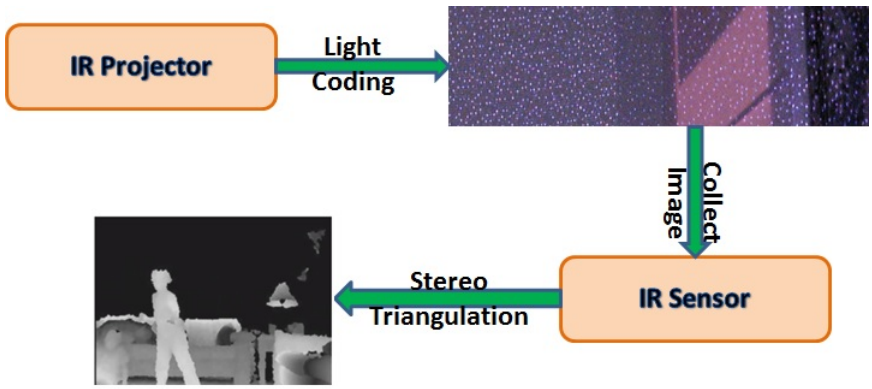

Figure 5: Kinect Processing Technique: How It Works

sensor projects static pattern of specifications on the scene or objects environment using IR sensor. It uses second sensor collecting projected light pattern image. Discovered by Daniel Reetz [21], the depth image is produced by stereo algorithm from image. Figure 5 shows the processing technology of how Kinect generates a depth image.

II.1.3 Stereo Triangulation Algorithm

The depth image is extracted by stereo triangulation algorithm. The theory behind this approach is that Kinect deploys a hidden pattern image with its chip logic. The second one is the image of the specifications captured by IR sensor. Those images are not equivalent, i.e. the distance between laser and sensor is different. Therefore images are corresponding to different camera positions. It can then calculate each specification depth using stereo triangulation.

Kinect sensor applies stereo triangulation algorithm to compute object depth in an image. Figure 6 shows how stereo analysis computes the depth of points in an image. Let us consider the geometry of stereo cameras with parallel optical axes first. Camera 1 and camera 2 are placed at O1 and O2 respectively. The focal length of camera is denoted as d, baseline as B. The object P has focus point 1 and focus point 2 with corresponding image points (X1, Y1, Z1) and (X2,Y2, Z2). Object P is at 3D (X,Y,Z) position reference in world coordinate. The baseline B is the distance between two lens centers. XZ is where the optical plane lies, X axis equals the baseline. Y axis is

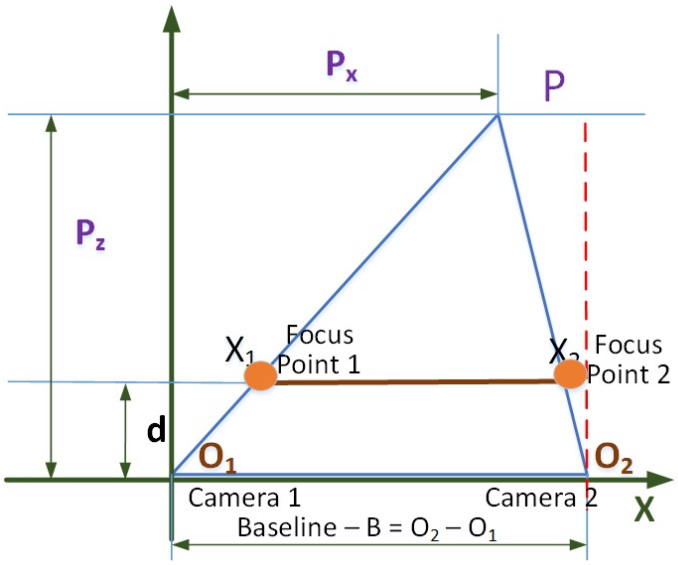

Figure 6: Stereo Triangulation Algorithm

perpendicular to the optical axes.

In this simple model that optic axes of two cameras are parallel, depth Z can be determined from disparity of cameras. This method is called triangulation, which can be shown by equations II.1.

$$\begin{aligned} Z &= (B * d)/(X_2 - X_1) \\ X &= X_1 * Z/d \\ Y &= Y_1 * Z/d \end{aligned} \tag{II.1}$$

The unique hardware design and algorithm used for producing special images, along with Software Development Kit (SDK) provided by Microsoft,the total package of Kinect sensor enables software engineers to develop sophisticated computer-based human activity tracking applications in many area of real world.

II.2 Human Motion Tracking and Recognition Using Kinect Device

Since debuted in 2010, Microsoft Kinect becomes one of the most popular game controllers in past years. Kinect provides an user interface that can be naturally interacting with game console by human body gestures or voice commands. Such unique functionality and feature have attracted many researchers to explore its applications beyond video gaming, especially in detecting and monitoring human activities and movements.

Based on fundamental principles in computer vision-based motion tracking and monitoring techniques, Kinect sensor is built with its unique and advanced hardware design. In addition to many intriguing features, Kinect also is very affordable and cost is in range from $39 (Kinect I) to $199 (Kinect II). It is equipped with an RGB camera, infrared depth sensor, and multi-array microphone. The depth sensor is able to parse the depth data from an object in 3D coordinates by using additional infrared laser projector and a monochrome sensor. Four microphones also allow users to control computer or a game console with human voice command.

Releasing first version of Software Development Kit (SDK) for non-commercial use in 2010, Microsoft provides a free programming interface. The SDK enables users to develop application using C# and C++ programming languages for tracking human body movements. In 2014, Microsoft releases SDK 2.0 and new adapter kit for Kinect v2 sensor, which allows developers commercially deploy Kinect applications for Windows development platform for the first time [22]. Through the SDK, computer software developer are able to access Kinect color RGB, depth image and skeleton image to create applications that enable people to interact with technology by gesture and speaking. The cutting-edge Kinect technology from both hardware design and programming interface SDK make it possible to detect, track and recognize human body movement robust and dynamically in real-time. Applications of Microsoft Kinect have been extended to many fields beyond video games, including human computer interaction, healthcare, security

surveillance, humanoid robots, sign languages, and other areas [8]. Furthermore, computer developers and researchers have intensively studied algorithms, mechanisms, and techniques of human motion tracking and analysis using Kinect sensor.

In this section, we review the recent studies on human body movement tracking and analysis using Kinect sensor in (1) applications of the technology used in human computer interaction (HCI), human gesture and body motion, health-care, robots, and other areas; (2) algorithms used for image classification and segmentation; (3) techniques used for tracking, detecting, and extracting moving objects.

II.2.1 Applications of the Kinect Technology

The applications of human body movement recognition, detection, and monitoring technology have expanded from traditional visual gaming realm to non-traditional consumers and enterprises. The research work of vision-based body motion tracking and analysis have focused on tracking and labeling each part of body that performs some kind of actions. The body motion recognition technology has been used for physical therapy and rehabilitation exercises, monitoring patients' activities, and many other fields. However existing motion analysis tools for those applications are intrusive, for example, patients either have to wear body markers or attach inertial sensors. We are motivated to design and build an application that is non-intrusive, convenient to use, and inexpensive and at meantime is able to monitor and analyze the human motions in real time. Therefore it is very important to discuss those applications of Kinect technology used in computer vision-based human motion tracking and recognition.

Since introduction of Kinect sensor and free SDK libraries in 2010, the Kinect technology opened a huge door for developing computer applications beyond Xbox games. Software developers boost the development of various applications including the human computer interaction, healthcare, security surveillance, humanoid robots, sign languages, and other areas [8] in recent years. In this section we review the most literature regarding applications using the Kinect technol-

ogy.

II.2.1.1 Human Computer Interaction

The Human Computer Interaction (HCI) studies how people interact with computers. The goal of HCI research is to make computers interact with human in more practical, natural, and responsive, as if an interaction is between humans. Therefore incorporating human gestures is very important subject in HCI. The input and output techniques used by computers, interaction styles, and human factors are most challenging issues in this study. In the past years, there has been significant improvement of HCI process with employing touch-free device controller technology, such as that behind Microsoft Kinect. Low-cost Kinect sensor can facilitate natural communication between computers and human through body gestures, body movements, even with voice commands. In this section the focuses are reviewing applications in categories: (1) natural interaction and control applications, and (2) virtual reality gaming and real-time animation applications.

II.2.1.1.1 Natural Interaction and Control Applications The importance of HCI study is to make computers interact with human in more practical, natural, and responsive. Microsoft Kinect sensor provides a great opportunity to allow computer interacting with human by using body gesture, body movement, and verbal commands. There are many natural interaction and control applications emerged recently using Kinect sensor.

The HCI applications using Kinect sensor have grown rapidly in recent years. The primary technology used is to use human gestures to control computer without hand touching. The Villaroman *et al*. [23] proposed system that interacts with computer in classroom instruction on user gesture. Farhadi-Niaki *et al*. [24] proposed an input system that performs typical desktop tasks using arm gestures. Raj *et al*. [25] presented a different approach manipulation 3D on a desktop display. Francese *et al*. [26] presented a 3D navigation user interaction application.

Applying advanced HCI technology to Kinect sensor is also studied by many applications. Chen *et al*. [27] proposed an Interactive Music conductor Generation system allowing the music

arranged from music conductor's hand gestures in real-time. Freitag *et al*. [28] presented a low-cost solution to integrate feed-forward and enhanced real-time feedback for multi-touch applications. Bragdon *et al*. [29] built a device combines multi-touch screen, mobile devices and Kinect sensor to access, control, and share information through different hardware devices for a group meeting.

II.2.1.1.2 Virtual Reality Gaming and Real-time Animation Another major interest in Kinect technology is in virtual reality gaming and real-time animation. The recent innovative human body movement tracking and recognition technology allows users to interact with augment objects freely in real-time for computer-based virtual reality applications and augmented reality games using Kinect.

Aitpayev *et al*. [30] applied the Kinect technology to make the human body a physical part of augmented reality for interaction without wearing a special suit with infrared LEDs or attaching special markers. Tong *et al*. [31] studied an advanced computational algorithms to compute join rotation angles from Kinect for skeleton animation. Franke *et al*. [32] proposed mathematical foundations for using Kinect depth images for mixed reality applications. Tong and his colleagues [33] demonstrated a system that can scan 3D full human body shapes by using multiple Kinect sensors in a more convenient way.

Kinect technology has also been widely used in popular augmented reality games. A computer game developed by Nakachi *et al*. [34] can express "individuality" in their proprietary software package using Kinect. Hai *et al*. [35] developed an interaction system for treadmill games based on Kinect depth maps. The HoloDesk [36] is an interactive augmented reality system combining an optical see-through display and a Kinect to create the illusion that users can directly interact with 3D graphics. The Wizard-of-Oz [37] is a *guessability* game to examine child-defined gestures using Kinect.

II.2.1.2 Human Gesture and Body Motion Recognition

The Kinect technology has provided ample opportunities for accurately, robustly, and interactively tracking the human body without adhesive markers. In this section we focus on reviewing recent research in hand gestures, human pose tracking, and body motion recognition and construction in real-time using Kinect.

II.2.1.2.1 Hand Gesture Recognition The research on hand gesture recognition based on Kinect has drawn great attention. The techniques of hand gesture recognition using Kinect was studied as early as in 2011 in [38, 39]. Doliotis *et al*. [40] proposed an approach of detecting hand gestures using Kinect depth images, which is named Dynamic Time Warping (DTW) for translating and scaling invariants that are required for many human-computer interface systems. Ren *et al*. [41] proposed a distance metric that can measure the dissimilarity between hand shapes with only matching the finger parts, not the whole hand in order to get better recognition results. Chen *et al*. [42] presented an integrated system using both head and hands tracking algorithm that aims to achieve robustness and real-time efficiency in the presence of occlusion, disturbance, and the appearance variation in Kinect images.

There were also some studies on the benefits of combining Kinect with other devices for hand gesture recognition. Frati *et al*. [43] presented an approach of combining Kinect and wearable devices. The results showed using of Kinect technology helps solve the poor sensing problem in wearable haptics devices. Trindade *et al*. [44] proposed a system that can reduce the search space and enable a robust and less computation-intensive recognition process. Wan *et al*. [45] investigated a gesture recognition system to control the movement of a mobile robot aiming to recognize several predefined gestures by using a sequence of Kinect depth images.

II.2.1.2.2 Human Posture Recognition There have been applications for recognizing human postures using Kinect sensor. A group of researchers [46] validated the use of Kinect for human posture recognition using a multi-camera 3D coordinates to build a ground truth with 20 human

subjects. Three postural control tests are forward reach, lateral reach and single-leg eyes-closed standing balance. Kondori *et al*. [47] proposed a system of recovering the six degrees of freedom (DOF) of head motion from Kinect depth image frames. Monir *et al*. [48] proposed an initiative to challenge privacy violation using depth image or RGB image for recognizing human postures with objective of protecting the privacy of the target objectives. This learning-based approach stored features that can be used later for searching matched objects in the testing environment. Zainordin *et al*. [49] presented a novel approach for tracking human poses using skeletal and color images to build a training-based system, which can identify different human poses and actions.

II.2.1.2.3 Human Body Motion Tracking Tracking human body movement in real-time is a challenging task. There were intensive studies for exploring opportunities to utilize low-cost Kinect sensors to accurately track human body movement in real-time. Chaves *et al*. [50] presented an approach that is not only to identify users' movement, but also to provide feedback on the context of the movement based on some predefined checkpoints. Miranda *et al*. [51] introduced a method for real-time gesture recognition using Kinect skeleton data, which can recognize each pose that is represented by angular skeleton joint. Wu *et al*. [52] presented a study using one-shot learning gesture recognition system with Kinect depth and color images. This system can classify gestures from learning example.

Xiao *et al*. [53] proposed a system to recognize human postures using Kinect, which can measure human body position in physical coordinates, and extract human features from depth images and skeletal data for robot control, behavior monitoring, and driving behavior learning. Iwane *et al*. [54] developed an application for simulating arm movement in Japanese flag semaphore. This application consists of technique of detection arm position, and discrimination of a semaphore to detect and identify a Japanese Katakana character from a sequence of semaphore movements.

II.2.1.3 Health Care Applications

With fast growing aging population, the need of affordable healthcare services is in high demand. The idea of using Kinect sensor to monitor physical rehabilitation exercises and other patients' activities has drawn great attention. Traditionally, motion analysis tools are intrusive because patients either have to wear body markers or attach inertial sensors before Kinect sensor is available. With the introduction of Kinect, it is possible to provide markerless full-body tracking. In this section we review the applications of the Kinect technology in healthcare focusing on physical rehabilitation exercises, fall detection and prevention, and medical operating room assistance.

II.2.1.3.1 Physical Rehabilitation Exercises In general physical therapy, rehabilitation training programs involve extensive, repetitive range-of-motion and coordination exercises, which require medical professionals supervise patients' movements and assess their progress to make sure the exercises movement is correct. It is no doubt such services are costly. In order to meet increasing demands and reduce the cost, physical therapy and rehabilitation providers are looking for computer technology that can assist them to provide services to patients in home-based, low cost and user-friendly environment. An basic requirement for technology is that it should help patients learn and perform exercises patterns repetitively and correctly. The affordable, portable, and easy-to-set-up Kinect sensor has provided such opportunities for researchers.

A research group at the University of Southern California Institute for Creative Technologies (ICT) has successfully applied Kinect technology to study virtual reality simulation technology for clinical purposes since 2011. Through extensive evaluation, assessment, and analysis, the scientists at ICT have proved that the Kinect technology can make a major contribution to the quality of traditional intervention physical rehabilitation exercises that specialize in mental health therapy, motor skills rehabilitation, cognitive assessment and clinical skills training [55–59]. Their results show that Kinect can achieve competitive motion tracking performance compared to the traditional, more expensive system. The system developed overcomes the limitation of Kinect when testing

subjects who are out of camera range or whose upper extremities are occluded by their body.

Some physical rehabilitation programs target specific patient groups. Rahman *et al*. [60] presented an interactive rehabilitation system for disabled children. This system utilizes Kinect to record rehabilitation exercises performed by a physiotherapist or a disabled child. The exercise session can be synchronously played in an on-line virtual system, which provides patients with visual guidance for performing correct movements. Chang *et al*. [55] described a study that assesses the possibility of rehabilitating two young adults with motor impairments using a Kinect. Cervantes *et al*. [61] presented their work on cognitive rehabilitation for Alzheimer's patients using a Kinect-based video game. The interactive body motion controlled game increases patients' motivation to participate exercises. In a similar research work, Saini *et al*. [62] aimed at increasing patients' motivation for therapy using a Kinect-based game for stroke rehabilitation. They studied the feasibility and effect of new game technology to improve the accuracy of stroke exercises for hand and leg rehabilitation. Gotsis *et al*. [63] demonstrated a mixed reality game for upper body exercise and rehabilitation using Kinect.

Pedro *et al*. [64] proposed to use Kinect in conjunction with rehabilitation robotics. The benefit of combining the Kinect sensor with a robot is the reduction of hardware cost when multiple cameras are needed to overcome occlusions.

Although Kinect has been proven to be a good replacement for the commonly used inertial sensor for tracking human body movement, the combination of using both devices can achieve more satisfying results. Bo *et al*. [65] proposed a method to combine Kinect with portable sensors, such as accelerometers, gyrometers, and magnetometers, for measuring human motion for rehabilitation purposes. In this study, Kinect was used to temporarily correct the overall estimate and to calibrate the inertial sensors for long-term operations.

II.2.1.3.2 Medical Operating Room Assistance: The growing use of advanced imaging devices and image guided procedures in surgical settings has imposed an increasing need for interaction under high sterile conditions among medical professionals. Kinect provides an appealing

opportunity to control medical images or image-guided devices without free touching approach. Some researchers have investigated gesture recognition systems based on Kinect to address needs in medical surgical rooms.

Bigdelou *et al*. [66] proposed a Kinect-based intra-operative medical image viewer, which is can be used in a surgical environment. The system provides a group of reliable training datasets for the most appropriate operating related gestures. Experimental results derived from evaluations of several types of feature representations show that their system could be viable in practice. Allowing interactive exploration of medical digital images, such as CT, MRI, or PET, Gallo *et al*. [67] developed an open-source system using Kinect in operating rooms. The programming interface utilizes hand and arm gestures to execute basic computer tasks such as selection, zooming, translating, rotating and pointing, and some complex tasks such as the manual selection and extraction of a region of interest as well as interactive modification of the transfer function used to visualize medical images.

II.2.1.3.3 Fall Detection and Prevention Kinect technology has been also used to detect and prevent falls and other dangerous activities for elderly people in a number of studies.

Mastorakis *et al*. [68] introduced a Kinect-based real-time fall monitoring and detection system that can automatically detect a range of falls including forward, backward, and sideways. The detection is carried out without knowledge of the floor plane coordinates or predefined particular body parts. Rougier *et al*. [69] proposed an approach to address the occlusion issue in detecting human body falls using Kinect. Bian *et al*. [70] presented an approach to detect falls by extracting skeleton data from Kinect depth images based on the fast randomized decision forest (RDF) algorithm. Zhang *et al*. [71] proposed an approach that employs two Kinect sensors to detect human falls. This approach is based on statistical information about how the person moves during the last few frames. The major contribution is that it collects all training data from a specific viewpoint, and then collects all the test data from a different viewpoint. Stone and Skubic [72] developed a system for capturing the variations of stride-to-stride gait in home environments for elderly adults

using Kinect. By measuring the changes in gait, falls can be predicted. In the research work of Parra-Dominguez *et al*. [73], a system is proposed to detect falls and other abnormal events on stairways instead of at flat level using Kinect. Ni *et al*. [74] developed a Kinect-based system to prevent potential falls in the hospital ward environments. This system automatically detects the event of patient getting up out of a bed.

II.2.1.4 Humanoid Robotics Applications

Kinect technology has been also used to control humanoid robots. In recent studies, traditional robotics controlling sensors including laser, ultra-sonic and radar sensors, have been either directly replaced by or integrated with Kinect. In this section we review the applications of Kinect technology in navigating and controlling mobile devices, interactively and remotely controlling robotic devices.

II.2.1.4.1 Navigating and Controlling Robotics Humanoid robots are coming and they have gradually entered our life impacting human life in many ways, performing house chores, assisting elderly people, providing education, and completing tasks in severe conditions. How to naturally navigate this human-body shape device effectively becomes a big challenge for us. The emerging Kinect technology provides an ideal interface to accomplish this goal without using wearable devices.

El-Iaithy *et al*. [75] proposed an application to navigate an indoor robot using Kinect with inertial sensors to optimize navigation accuracy, particularly for obstacle detection and avoidance. The experiments show that Kinect is ideal for indoor robotic applications, but not suitable for outdoor applications or when the robot is under strong lighting sources. Hoilund *et al*. [76] conducted the feasibility study for using gestures to control industrial robots using Kinect. The experimental results show that Kinect enhances a mobile robot with the ability to interpret human actions. However Kinect data is more noisy than more expensive motion capture systems. In additional to using hand gestures to control robot devices, Nguye *et al*. [77] presented a human imitation system that

can map different kinematic structures using Kinect sensor. This system can reproduce imitated human motions during continuous and online observation of a humanoid robot. The experimental results show that the system can feasibly adjust robot's motion to satisfy the mechanical constraints and dynamics consistency.

II.2.1.4.2 Interactively Controlling Robotics Study of interactively control robotics using Kinect technology has gained attention from scientists recently. The advances in the Kinect technology on human computer make it very attractive and practical for interactively controlling humanoid robotics.

Xu *et al*. [78] proposed a system that can navigate a robot using dynamic hand gestures in real-time using Kinect sensor. The system recognizes human hand gestures based on a Hidden Markov Model algorithm and converts them to control commands for the robot. The proposed system can work effectively in the complex environment with an average real-time recognition rate of 98.4%. Cheng *et al*. [79] developed a Human-robot interactive demonstration system that provides a visual interpretation of gestures and sends it to robots to achieve natural interaction between a human and robots.

II.2.1.4.3 Remote Controlling Robotics The Kinect technology can remotely to manipulate the robotic arm with human gestures. A number of applications has been emerged in past years, such as a mobile robotic motion capture system, a mobile robot tracking system, and a quadrotor helicopter controller.

Boyd *et al*. [80] developed a training speed skating system by placing Kinect on a mobile robotic device to capture motion in situ. The system enables a speed skater on the ice to capture the full-body motion. Machida *et al*. [81] introduced a tracking system with Kinect on-board of a mobile robot. The Kinect is used to control the velocity and altitude of the mobile robot via human gesture. The experiments show that recognition and tracking are effective and robust. Stowers *et al*. [82] proposed a system that utilizes Kinect's depth images to control the altitude of a flying

quadrotor helicopter. The system is capable of maintaining a constant altitude during flight of the quadrotor helicopter in dynamic environments. Wang *et al*. [83] developed an application to control Aldebaran NAO humanoid robot using Kinect. The testing results show that the system is robust and flexible enough to recognize various human motions. Zuher *et al*. [84] proposed an approach that a humanoid robot can be remotely controlled through the recognition of human motions (such as neck, arms, and leg movements) using Kinect focusing on two major tasks: (1) a real-time recognition of human movements; (2) action to make the robot perform.

Above applications demonstrate that the low cost, sufficient frame rate and depth accuracy of Kinect can make it attractive for controlling real-time robotic actions.

II.2.1.5 Other Applications

In addition to the applications reviewed in above sections, the Kinect technology has been used in various areas, such as speech recognition, sign language translation, retail surveillance, 3D mesh reconstructions, arts performance, workspace safety training, and Internet tools.

II.2.1.5.1 Retail Services The Kinect technology can be very beneficial in retail services due to its recognition robustness and accuracy.

Popa *et al*. [85] proposed a system for analyzing human behavior patterns related to shopping interaction, such as browsing through a set of products, examining, picking products, trying products on, interacting with the shopping cart, and looking for support by waving one hand. Kinect was used to capture the motions that would help assess customers' shopping behavior and detect when there is a need for support or a selling opportunity. This application aims to increase customer satisfaction and improve services productivities. Wang *et al*. [86] proposed an application that allows the users to virtually try on different handbags at home. The users can interact with the virtual handbags naturally, such as sliding a handbag to different positions on their arms and rotating a handbag to see it from different angles. The users can also see how the handbags fit them in different virtual environments other than the current real background.

II.2.1.5.2 Speech and Sign Language Recognition Human natural language recognition has significant impacts on our society. Taking advantage of the Kinect technology, researchers extended existing work to use advanced imaging information for improving the robustness of speech recognition.

Galatas *et al*. [87] incorporated a speaker captured by Kinect as a third data stream in an automatic speech recognizer using facial depth data. The results demonstrate that the system performs better due to the depth modality, and the accuracy is increased when using both visual and depth modalities compared to speech only recognition. Anjo *et al*. [88] developed a system to recognize the Brazilian Sign Language (Libras) in real-time using static gesture. Their work focused on segmentation and classification processes.

II.2.1.5.3 Workplace Safety Training The Kinect technology can also be applied to work place safety training. Martin *et al*. [89] proposed an application that can provide the prevention of back injuries caused by lifting heavy objects with Kinect sensor. The system can also notify the worker about dangerous movements in real-time at the lift location. The system can also be used as a training tool due to its capability of recognizing lift skills. Dutta *et al*. [90] utilized Kinect to record postures and movements for determining the risk of musculoskeletal injury in the workplace. The system shows that Kinect has comparable accuracy versus existing lab based systems. It provides a compact, portable motion capture system allowing workplace ergonomic assessments to be done simply and inexpensively.

II.2.1.5.4 3D Reconstruction It is a challenging task to build geometrically consistent human 3D models because of individual pairwise errors. Chatterjee *et al*. [91] proposed an application to use Kinect for constructing 3D human image reconstruction. A challenge for doing so is that the depth images obtained via Kinect have high noise levels. Farag *et al*. [92] proposed an algorithm that can efficiently calculate a vertex antipodal point for reconstructing a skeleton of a 3D mesh for mesh animation. The algorithm was successfully tested on different classes of 3D objects and

produced efficient results. It has the advantage of producing high quality skeletons as it preserves details, which is suitable for applications where the mesh skeleton mapping is required to be kept as much as possible.

II.2.1.5.5 Others Norrie *et al*. [93] developed a mobile application used for virtual bookshelf. This application supports rapid interaction by using 3D position data from Kinect to simulate a proximity sensor, which associates content on the device with regions of a room. Rodrigues *et al*. [94] introduced a tool called "MotionDraw" for enhancing art performance using Kinect. This tool can track movements of users, which is able artists, performers, dancers and the audience to design, create and control hybrid digital performances. Mora *et al*. [95] presented a method to estimate the eye gaze estimation in 3D space using Kinect. This method is designed to obtain robust and accurate head poses on visual data. It can also rectify the image that exploits the 3D meshes tracking and collect ground truth data. Winkler *et al*. [96] introduced an affordable and non-intrusive solution for automatic camera control for tracking a presenter during speech using Kinect. The approach enables video cameras to automatically follow presenter on podium. Boulos *et al*. [97] developed an application called "Kinoogle" to control virtual globes, such as Google Earth, Bing Maps 3D, and NASA World Wind using Kinect. Liebling *et al*. [98] introduced another Internet application called "Kinected Browser" for Web browsing through touch-free technology. This toolkit can augment web pages with speech input and gesture input via Kinect. Web interactions are designed for new form-factors such as large display walls, and TV sets.

II.2.2 The Algorithms and Techniques using Kinect Device

In above section, we have reviewed applications using Kinect sensor. In this section, we review the algorithms and techniques, i.e. what used to develop Kinect applications for motion tracking and analysis. First we describe the technology used by the Kinect sensor and what the Microsoft Kinect software development toolkit (SDK) provides. Then we provide an overview on the general algorithms and mechanisms for motion tracking and analysis that have been used in

Kinect applications.

The review of the general algorithms and mechanisms is divided into two subsections, one on the representation of the motion information obtained from extracted features in image sequences, and the other on the recognition of objects by matching testing sequence with pre-learned motion models.

II.2.2.1 Kinect Motion Tracking and Analysis Techniques

The Kinect sensor can track and recognizes human body movement through two interactive phases. In the first phase, it collects real-image of objects from built in cameras and computes depth images; in the second stage it extracts human skeleton data from the captured depth images. The motion tracking and analysis techniques and algorithms that enable such processes are introduced below.

Microsoft Kinect consists of three image processing components, a RGB camera, an infrared camera, and an infrared projector. The Kinect depth imaging technology was originally developed by PrimeSense [99]. Later Microsoft acquired PrimeSense technology and was licensed for its Kinect sensor. The infrared projector projects a speckle pattern onto the objects in view using an astigmatic optical element. Subsequently, by using an adapted stereo triangulation method the depth of each pixel can be determined through geometry calculation. Hence the depth image is generated for captured object from infrared camera.

After a depth image is computed from captured human image, the next step is to extract human body skeleton data it in real time. This process must be computational efficient because it is implemented in real-time. A 4-stage process, which consists of background removal, body part classification, centroid classification of body parts, and model fitting, was used to interactively extract skeleton data from a single depth image [100].

Background removal process is to check each depth image pixel to see whether it belongs to a human body part or not. The objective of this process is to effectively identify each body part individually and consistently from frames to frames. In order to make process more robust and

efficient, a randomized decision forest algorithm is used to classify human body pixels, which is introduced by Shotton *et al* [101]. This technique is more efficient than the commonly used temporal segmentation techniques. Each active pixel traverses the trees starting at root. The decision goes either left or right based on an evaluation function. At the end of traverse, the probability of the current pixel that is belong to a particular body part is calculated and saved. The total value of all pixels probabilities is computed and the highest value from a particular body part indicates that pixel is belong to that body. The randomized decision forest classifier can be trained by a large, realistic and highly synthetic set of training images, in which about one million motion depth images of people walking, kicking, jumping, dancing, etc. are stored. The training set also uses synthetic 3D models generated from 15 bodies with different types, such as weight, height, etc.

After tagging each pixel, the next step is to classify body parts. Kinect sensor refines a full image of pixels into 31 human body parts. This intermediate body parts representation is used to computational cost for spatially localized human body joint. Such process also increases recognition accuracy. The centroid of each classified body part is then calculated. In addition, model fitting is carried out to aggregate the centroids into a human skeleton. In model fitting stage, a local mode-finding approach based on mean shift with a weighted Gaussian kernel is used. The 3D body parts are locally accumulated over a small set of parts. For example, the four body parts covering the head are merged to localize the head joint. Therefore, each mode is taken back to the scene to produce a final joint position proposal.

II.2.2.2 Image Segmentation Techniques

Image segmentation is one of the most important techniques used in detecting and tracking human motion. It divides an image into small segments or a group of subsets of pixels for further analysis. The ultimate goal for segmentation algorithms is to separate the interesting objects from their surrounding objects or background. It is also a complex image processing task that requires a lot of computational resources due to the difficulties of segmenting human figures in moving frames. The object is not only moved in position, but also changed with a high degree of freedom

during movement. There are many approaches and mechanisms for classification and segmentation algorithms used in computer-based human body movement analysis. In this section we provide an overview of image segmentation techniques that are used in Kinect applications, particularly used for interactively tracking human body. The focuses are the techniques that have been applied to Kinect application, including Region of Interest (ROI), Support Vector Machine (SVM), Spatio-Temporal Segmentation, Motion Segmentation, Multimodal Mean, and other techniques.

In this subsection, we provide insight review in the techniques and algorithms used for human body movement recognition in computer vision-based, markerless, and real-time environment. The discussion is divided into following subjects: feature extraction, action learning, classification and recognition.

II.2.2.2.1 Feature Extraction Feature extraction is the main vision task in action recognition. It extracts human posture and movement scenes from the video frames that are discriminative for human motions. The extracted features can be used for recognition process later. The features can be represented by a very complicated body parts or a simple silhouette image. The difficulties of performing this task are how to define location position, object occlusion, background, process robustness, and illumination. Those are to be addressed for future study according survey [102].

In order to extract features from image frames, segmentation is one of the most important techniques used in computer vision based tracking human motion. This process is for partitioning an image into segments or a group of pixels so that the desired features can be separated from background or surrounding objects. Computational process requires a lot of computing resources because of the difficulties human body parts in moving frames and being tracked with a high degree of freedom. There are many classification and segmentation algorithms used, we introduce segmentation algorithms that are used in Kinect applications, particularly used for interactively tracking human body. The most popular techniques are Region of Interest (ROI) [85, 87], Support Vector Machine (SVM), Spatio-Temporal Segmentation [66], Motion Segmentation, and Multimodal Mean [103, 104] .

The Region of Interest (ROI) is the region of an image that is specially interested for processing. All of the pixels outside of the ROI are not interested. An important aspect is to select a suitable region of interest for successfully tracking and recognizing a moving object. It is a quite simple process with a particular interesting object in predefined selection. However it is difficulty when an interactive and real-time environment are not predefined. ROI is widely used in the study of speech recognition.

The spatio-temporal segmentation is another technique used in image segmentation process. The spatio-temporal segmentation defines the spatial and temporal coherences in sequential image frames with specific object boundaries and associated objects. Nevertheless this technique can predict new position for each rejoin in moving frames quite efficiently. It has been used in many motion analysis applications, such as network infusion detection, speech recognition, and human activities tracking.

Motion segment algorithm is to divide moving image into featured trajectories. This algorithm was first introduced by Tomasi and Kanade [105] stating that trajectories of motion is a geometric constraint, is able to classified as a multi-dimensional linear manifolds. This algorithm can be used to segment rigid, non-rigid, degenerate, and non-degenerate or any combination of them. In practical, the human body is often classified by rigid segments.

The Multimodal Mean (MM) algorithm extracts the object from background image content [103]. The MM algorithm works better for RGB color based or gray-scale images. But it generates a noisy segmentation when infrared projected images is produced.

II.2.2.2.2 Region of Interest The Region of Interest (ROI) can be defined as the region of an image that has an interest and specially is chosen for image processing. All of the pixels outside of this region is discarded. The region of interest (ROI) is the "available space" for movements. How to select a suitable region of interest is a key factor for successfully tracking and recognizing a moving object [75]. It is a simple and straight forward process with a predefined region associated with a particular interesting object in the image. However in real-time environment, the region,

or interesting objects are not able to predefine. The Kinect applications are able to specify an interesting facial region or an interesting shopping area in real-time.

In a study on speech recognition, Galatas *et al*. [87] proposed an application that incorporates facial depth data of a speaker with Kinect device as an additional data stream for an audio-visual automatic speech recognizer. This approach can detect the human facial mouth region and extracts its ROI from both the video and depth streams. Furthermore, the coordinates of the ROI are computed and normalized by finding the median of each coordinate for the last 10 frames. Result shows this technique reduces abrupt movements of the bounding box; hence it improves accuracy and efficiency of recognition by limiting false detections.

In Popa *et al*. [85] publication, they proposed a system that can recognize shopping related actions. The interested shopping area is segmented into the ROI such as products, passing, pay desk, or resting areas. The customer action in a specific ROI provides a set of data for the semantic modeling, such modeling then is used to analyze shopping behaviors.

II.2.2.2.3 Spatio-Temporal Segmentation The spatio-temporal segmentation is a widely used in motion image process, which is used to exploit the spatial and temporal relationship in sequential image frames. The key fact of this algorithm is to identify object boundaries and associates pixels over frames in time sequence. It has been used in many motion analysis applications, such as speech recognition, human action tracking, network intrusion detection, neuron activity modeling, etc. Although in many recent published literatures, this technique proved to be able to adequately predict new position for each rejoin in moving frames, the prediction is heavily relied on segmentation from previous frame. Wherefore, robust and efficient estimation analysis is based on initialization, which is not a useful for interactive motion tracking and analysis as what the Kinect sensor provides. Nevertheless, several Kinect applications have approved to perform successful motion recognition using this algorithm.

Bigdelou *et al*. [66] proposed a gesture recognition application that allows simultaneous recognition of the type of a gesture as well as the relative poses within a gesture (*i.e.* spatio-temporal

gesture) using Kinect motion data. Result shows great potentials for building 3D gesture-based applications. Wu *et al*. [52] presented an approach using temporal segmentation as a preprocessing step for gesture recognition using only one learning example. In this system, frames that are similar to the beginning and ending in the video sequence are sought, and defined as the interval frames between two gestures in a video sequence. The experimental results show that the accuracy for whole gesture recognition system is upper bounded by this temporal segmentation performance.

II.2.2.2.4 Motion Segmentation Motion segment algorithm is a technique that can be used for segmentation of the feature trajectories in the moving image frames.

Tomasi and Kanade [105] introduced an advanced algorithm that is based on the idea that trajectories of a rigid motion have a geometric constraint, can be casted as a multi-dimensional linear manifold finding problem. This algorithm is widely used to segment a wide range of motions including independent, articulated, rigid, non-rigid, degenerate, and non-degenerate or any combination of them. However the challenges remain for how to reduce the complexity of computing a linear manifolds of higher dimensions for interactive human motion tracking. In motion segmentation process, the human body is often characterized by rigid segments. Thus body movement can be presented by a group of articulated segments that can rotate relatively to each other on a joint. Based on this theory, Rohith and Kambhamettu [106] proposed an application that divides articulated motions into several rigidly moving components. Using this motion segmentation algorithm [107] developed a system that can provide synthesis of rigid structure. The system is able to estimate human motion in 3D directions. The experimental results demonstrate that this method is more general in terms of the feature point distribution and less complex of motion compared with other motion segmentation methods.

II.2.2.3 Using Dataset for Tracking and Recognition Methods

During segmentation process, the features from image of interested object are identified and extracted from non-interesting surroundings objects and background. The next step is to recognize

the identified object in a sequence of image frames that is also considered as classification problem. This process engages matching an unknown sample sequence against predefined sequences that are particularly represented in human actions. Therefore the classification problem becomes learning process from training examples. Tracking and recognition for human body movement techniques are categorized as either using explicit body models or using statistical aggregation data models. We focus on computer vision-based methods that analyze, segment and classify movements for recognizing actions.

Using datasets that includes explicit body models are used by many human body movement recognition algorithms. This approach learn human actions from the pre-record video frames and then recognize actual actions directly from image measure.

For example, KTH [108] dataset contains six most common actions jogging, walking, running, hand waving and clapping, and boxing, which are recorded from 25 different subjects in possible four scenes. All together, there are 2391 video sequences in dataset. The other popular datasets are Weizmann, IXMAS, and CMU Graphics Lab Motion Capture Database [109], according a survey of vision-based methods for action recognition [110]. Although the datasets contain many actions performed by various subjects in different scenarios, and recognition rate is pretty high (above 99% [111]). However building such database becomes difficulty and time consuming for a large amount of samples of body motions or actions required.

Chaquet *et al.* [112] provided the detailed analysis of almost 68 different datasets used for human motion detection and recognition.

II.2.2.3.1 Recognition From extracted body features, the actual actions are detected and recognized by using statistical aggregation data algorithms. *Dynamic Time Warping* (DTW) is a well-known technique to compute the similarity between two temporal sequences. This method can handle both time and speed. It calculates possibility of maximum similarity between tracking objects and training dataset. Traditionally DTW has been widely used for automatic speech recognition to solve speaking speeds and pronunciations. Recently, this method expends to economics, bioin-

formatics [113] and human body movement and actions [114]. The benefits of using DTW is its higher computation performance for recognition process. However a main disadvantage of DTW is it requires high quality of dataset. In reality, it is very hard to obtain perfect human gestures, or actions in uncontrollable real-world environments.

Decision Forest: (DF) is another pattern matching technique for time-varying data. It builds a decision trees using training data. Such tree Decision forest based algorithms have been widely used in computer vision and medical image analysis [115]. The advantage of DF is it integrates classification, regression, density estimation, manifold learning, semi-supervised learning and active learning into a unified framework [116]. This means it only needs learning and optimization implemented once, and can be applied to many interactive real-time applications. As a variation of the DF, the randomized decision forest (RDF) algorithm utilizes randomly trained decision trees, which is used in Kinects software develop kits to produce skeleton images (described in Section II.1). Below we will introduce some Kinect applications using DF algorithm.

The Support Vector Machine:(SVM) is one of the most widely used classifiers. It performs a nonlinear mapping of the inputs vectors from original feature space into a high dimensional feature space, and then optimizes a hyperplane for separating data in such a high-dimensional feature space [117]. The key advantage of SVM is it guarantees maximum-margin separation using relatively little training data. Despite the success of SVM in motion classification, this technique does not extend naturally to multiple class problems [115]. With growing popularity of Kinect sensors, there are some studies on applying SVM to tracking human gestures and motion in real-time environments.

II.2.2.3.2 Using Statistical Model for Tracking and Recognition Methods Using statistical aggregation data models for human motion recognition has gained by Kinect applications, such as Rule-based, Support Vector Machine, Decision Forest, Hidden Markov Model. *Hidden Markov Model* (HMM) has been used in various human gesture and speech recognition applications. HMM uses a statistical time-varying model that describes simple and effective states in time sequences.

This model involves learning training dataset to establish model parameters first. The actual image are classified to discriminative features. By computing likelihood, the motion can be estimated by which action the testing motion image belong to, which is the result of recognition. Traditionally, HMM is used for speech recognition. However lately researchers have utilized HMM in the recognition for human body movement. The HMM can learn and classify time-sequential data without a motion capture database in real-time with achievable high accuracy for recognition. However, HMM does require a larger training dataset.

II.2.2.4 Marker-based Techniques to Track Human Body Motion

There are many motion capture techniques used to track human body movements. The most popular ones are marker-based and markerless [118]. The marker-based technique requires the images provided before tracking. It's much more simple to detect objects. On the other hand, markerless technique recognizes objects that are not provided beforehand, which is much more difficult to implement. In this scenario, the learning models are normally built before recognition process. The detail of implementations will be discussed below.

The marker-based technique is using inertial sensors , such as optoelectronic, stereo-photogrammetric, and other types of LED sensors placed on human body. The skin deformation and body displacement cause marker moving with respect to the underlying bone. Since the image marker can be used to determine the position coordinates, orientation and range of its trajectory system, therefore the corresponding position of joints or body parts that have markers attached to can be defined. inertial sensors that are attached to human body. The detection and recognition procedure includes installing the sensors on body and matching image to the corresponding body parts of a predefined structure. The movements of body parts are estimated based on a training procedure.

The benefit of using reflective markers is the high accuracy of recognition. But this approach is at the cost of ease of deployment and configuration. It is intrusive and very cumbersome to users. Some patients couldn't even place makers on their own body due to imperil difficulty.

The markerless motion-capture technique has gained great interests in last decade, which

has been demonstrated in many interactive systems and prototypes. This method is computer based approach that can actively track and recognize the objects in the real environment without placing markers on human body [119]. It identify patterns, colors or some other ”features” from camera image based on trained models. It allows development of more complicated application in real-time. It is a very attractive solution to problems of marker-based method due to the benefits of (i) no intrusive markers placed on body; (ii) less error caused by misplacing markers; (iii) less disturbance of movement due to markers' placement; (iv) cost effective; and (v) automatic processing in real-time. Nerveless, the disadvantages are difficulty of performing motion tracking and recognition and the ”trained” model must exist before recognition process.

The challenging issues regarding markerless motion capture approach are computing performance, the accuracy of recognition, efficient of training model and size of datasets available for use during estimation process. Even with such uncertainties, markerless is still considered as a promising techniques in regarding of human body movement tracking and analysis.

Our research focus is on using markerless technique, which is focused on computer vision-based algorithms and methods.

II.2.2.5 Motion Recognition Techniques

During image segmentation, the object features interested are identified or derived from non-interesting surroundings region or background. The next stage in motion recognition is to recognize the identified object in image sequences. This process is also considered as a classification. It involves matching an unknown test sequence with known sequences that represent particular human actions. Therefore the classification process is very critical during motion recognition. The motion classification and recognition techniques are categorized as either using explicit body models or using statistical aggregation data models. In this section, we review the techniques and methods used for human motion recognition in the Kinect applications.

II.2.2.5.1 Hidden Markov Model The Hidden Markov Model (HMM) has been widely used in real-time human gesture and speech recognition applications. HMM provides a simple and effective statistical framework for modeling time-varying vector sequences. This algorithm involves learning training dataset and classifying the image features observed from a particular motion. It has been used a large number of continuous vocabulary for speech systems and motion patterns, *ie* gestures, for body movements. Recently HMM has been used in the recognition process for human body movement captured by Kinect. The benefit of using HMM is that it can dynamically learn and classify time-sequential data without a motion capture database. The ability to learn from training data and to develop representations under a mathematical framework make HMM algorithms very attractive compared to other recognition methods. Nevertheless to achieve a high accuracy for recognition, HMM indeed requires a larger training dataset. In this section we particularly review HMM techniques for motion recognition in Kinect applications.

Mansur *et al*. [120] proposed an application method for action recognition using dynamic features of the human body. They used HMM to achieve good classification performance with a small number of training data captured by Kinect. The experimental results demonstrate that this method provides good discrimination compared to the Carnegie Mellon University (CMU) motion capture dataset and the Osaka University Kinect action dataset. Xu *et al*. [78] applied HMM to classify hand gesture sequences for real-time navigation of a robot using human hand gestures. The gesture features are extracted by HMM using a training dataset. This system can quickly detect the human body in front of Kinect and recognize the hand gesture input. The experimental results show robustness of the proposed human-robot interaction system. Zhang *et al*. [121] proposed an application applying the HMM and Neuro-Fuzzy model (HMM-NF) to model and configure golf swing actions. Combining the time-sequence serials modeling method of HMM and the self-learning Neuro-Fuzzy algorithm shows that the HMM-NF method presents better recognition performance than the standard HMM model. Sung *et al*. [122] proposed an application based on an alternative Hidden Markov model called Maximum Entropy Markov Model (MEMM). Sung and colleagues

collected Kinect RGBD images for training a two-layered maximum-entropy Markov model. They tested on activities in five different environments, kitchen, office, bathroom, living room and bedroom, and results show a recognition rate as high as 84.7%.

II.2.2.5.2 Dynamic Time Warping Dynamic time warping (DTW) [123] is a well-used technique to calculate the similarity between two temporal sequences that may vary in time and speed. It is computational algorithm for determining possibility of maximum similarity between unknown tracking objects and predefined data set. For decades, DTW has been widely used for various applications, such as automatic speech recognition with variations of speaking speeds and pronunciations, economics, and bioinformatics [113]. In recent years, DTW has been used for human body movement detection and recognition. The studies show that this method does not need high motion detaills. So it is able to increase computation performance for achieving the goal of robustness for recognition process. However a main disadvantage of DTW is it requires high accuracy of data set for computing similarity between testing objects and training objects. In uncontrollable real-world environments, obtaining perfect human gestures, or actions is hard to achieve.

Doliotis *et al*. [40] proposed an application that solves real-time challenging problems by detecting hands movement using Kinect depth data. They employed DTW to achieve an very important process in HCI for translation and scale invariant. Waithayanon *et al*. [124] used DTW to match two motions where angle are changed over time. Given an unseen motion, the classifier computes the distance between actual motion and known or training motions. The mothion with shortest distance is the motion found among a group of trained motions.

II.2.2.5.3 Decision Forest The Decision Forest (DF) algorithm is another well-known pattern matching technique for time-varying data. It is an algorithm for classification and regression by building a decision trees using training data. It has been widely used in computer vision and medical image analysis [115]. The advantage of DF is that it integrates classification, regression, density estimation, manifold learning, semi-supervised learning and active learning into a unified frame-

work [116]. Hence it only needs learning and optimization implemented once, and can be applied to many interactive real-time applications. A variation of the DF, Randomized Decision Forest (RDF) is an algorithm than utilizes randomly trained decision trees. It is the fundamental technique used in Kinects software develop kits to produce skeleton images (described in Section II.1). In this section, we introduce some Kinect applications using DF algorithm.

Miranda *et al*. [51] applied the decision forest algorithm to identify gestures in real-time using Kinect motion data. A gesture is modeled as a sequence of key poses. During training, a decision forest is constructed based on the key poses. Each path from a leave node to the root represents a gesture, and it is labeled with the corresponding gesture. Gesture recognition is reduced to a simple searching problem based on the decision forest. Several groups of researchers [70,101,125] have used RDF in fall detection. RDF can recognize skeleton shape deformation caused by the human body falling. However, due to changes in the orientation of the body during movement, the accuracy of recognition is reduced. Considering that the human fall is a fast activity, the RDF computational performance is critical. Wei *et al*. [126] applied RDF to train a classifier for automatic labeling of depth pixels. The benefit of using RDF is that it can naturally handle multi-class problems robustly, quickly and most importantly, remaining reasonably easy to train.

II.2.2.5.4 Support Vector Machine The Support Vector Machine (SVM) is one of the most popular classifiers. It performs a nonlinear mapping of the inputs vectors from original feature space into a high dimensional feature space [117]. The key advantage of SVM is that it guarantees maximum-margin separation using relatively little training data. Despite the success of SVM in motion classification, it can not extend naturally to multiple class problems [115]. With growing popularity of Kinect sensors, there are some studies on applying SVM to tracking human gestures and motion in real-time.

Madeo *et al*. [127] proposed an application that can segment a gesture into a sequence of motion data from Kinect sensor. This application transfers gesture analysis problem into a classification task using SVM. In addition, they applied several pre-processing methods to extract

time-domain and frequency-domain features. The study aims at finding the best parameters for a SVM classifier in order to distinguish the rest position from a gesture unit. Miranda *et al*. [51] classified key poses in a sequences of body motion through a multi-class classifier using SVM.The major components of key poses are the pose descriptor, the pose identification and the labels of pose sequences. Patsadu *et al*. [128] studied human gesture recognition using classification methods including BPNN, SVM, decision tree, and naive Bayes. Experimental results have shown that the back propagation neural network method outperforms other classification methods and can achieve recognition with 100% accuracy. Furthermore, the average accuracy of all classification methods used in this study is 93.72%, which confirms the high potential of using Kinect in human body recognition applications.

II.2.2.5.5 Kalman Filter The Kalman Filter is an algorithm that operates recursively on streams of input data to statistically optimally estimate of the underlying states. Specifically, it can be used in applications that involves time series analysis, such as signal processing and econometrics. With rapidly growing interests in computer vision-based human body movement tracking and analysis, the Kalman Filter algorithm has been increasingly used to track moving objects in image frames of time series.

Frati and Prattichizzo [43] proposed an application to animate hand avatars based on human motion data from Kinect sensor. This application focuses all the relevant values obtained from the previous steps are filtered and smoothed by the Kalman Filter algorithm. Then filter can predict the state vector that consists of the point position and velocity. Machida *et al*. [81] applied the Kalman Filter algorithm to reduce the noise and estimate the motion state in a robot application. The proposed application includes a tracking control interface with an on-board Kinect. The human position relative to the mobile robot is estimated from the Kalman Filter. El-laithy *et al*. [75] also listed the Kalman Filter as a potential way to optimize measurements for indoor robot navigation.

II.2.2.5.6 Least Squares Regression Linear regression is one of the most popular techniques in statistical analysis. The algorithm is simple and often outperforms complicated models when the number of training samples is small [129]. Least-squares regression is commonly used linear regression techniques. It is widely used in statistical analysis as well as in computer vision-based motion analysis.

Lui [130] proposed an application using non-linear least squares regression framework on manifolds for the one-shot-learning CHALEARN gesture challenge using Kinect sensor. One of the key contributions of this study is using the least squares fitting algorithm for a simple estimation model. The experimental results reveal that this method is competitive with the state-of-the-art methods. Kondori *et al*. [47] proposed an algorithm for recovering the six degrees of freedom of head motion from a sequence of Kinect image frames. This algorithm utilizes least squares to minimize the difference between the measured rate of depth change at a given point and the predicted rate by the depth constraint equation. Wei *et al*. [126] employed an adapted Lucas-Kanade algorithm [131] to solve the non-linear least squares problem for real-time pose estimation based on Kinect motion data.

II.2.2.6 Other Techniques

There are many techniques that are not mentioned above, which are used to perform image segmentation using Kinect sensor.

The Multimodal Mean (MM) algorithm has been used in image processing for many years. Apewokin *et al*. [103] proposed an application that can accurately separate the foreground of object from background image frames obtained from Kinect sensor. The MM algorithm works well for red, green, and blue (RGB) images and grayscale images. However, it doesn't work well with image obtained from infrared projector, *eg* Kinect's depth images due to noisy produced by segmentation. Brooks *et al*. [104] proposed an approach that can segment RGB images obtained from Kinect sensor using the MM algorithm and maps the pixels to the corresponding position in the depth

image. This method can remove the noise around objects very well.

Anjo *et al*. [88] presented a system that can track and recognize hand gestures using Kinect sensor. They developed a segmentation algorithm called "Virtual Wall" that sets a depth threshold like an invisible wall in front of the user. The position of this wall is calculated based on user's center of mass. This algorithm was applied to regcognition of Brazilian Sign. The experimental results show the recognition rate can be as high as 92%.

Lin *et al*. [132] proposed an application using Action Trait Code (ATC) for learning and recognizing human actions. The ATC uses the average velocity of body parts for segmentation. The human body is divided into several body parts, such as left arm, right leg, etc. The average velocity of each body part in an action sequence is labeled by action elements. The experimental results demonstrate that the proposed approach successfully delivers high recognition accuracy and is applicable to real-time applications.

Sivalingam *et al*. [133] presented an approach based on sparse representation. This approach is tested for human motions and gestures from predefined datasets and Kinect motion data. The experimental results show that the classification accuracy is 90% - 92% from the Australian Sign Language dataset, 98% - 99% from UCF FutureLight human motion capture dataset, and near perfect from the Kinect motion data.

II.2.2.7 Issues and Challenges

Microsoft Kinect sensor provides intriguing opportunities for us to naturally interact with computer systems in real-time. The various applications, algorithms and techniques used in studies involving in Kinect sensor have been reviewed in this chapter. Although Kinect sensors have been used in various applications as shown in Section II.2.1 and fundamental algorithms used by Kinect sensor in Section II.2.2 , there remains a number of challenging issues to resolve. Hereby we discuss the primary obstacles in the technology using Kinect sensor, such as intrinsic limitation of the Kinect technology, real-time motion tracking methods, and interactive feedback mechanism.

II.2.2.7.1 Overcoming Occlusions There are several studies about the limitation of Kinect technology used in real-time human body movement tracking and analysis. The largest obstacle is that Kinect contains large errors in the presence of occlusions [134, 135]. This limitation imposes a severe limitation on applying the Kinect technology to track human body movement because it involves self-occlusions or the use of external objects.

In order to overcome occlusions, better methods must be used to extract correct poses from the depth images. In [134], Shen *et al.* proposed an approach aiming to solve the pose correction problem by leveraging temporal information and by estimating systematic bias. The actual pose correction in the presence of occlusion is achieved by training their model with exemplar poses. In [135], Shum *et al.* proposed another approach that can recognize the initial body parts might be from the depth images in the presence of occlusion due to external objects. The goal is to evaluate the degree of reliability of each tracked body part, and consider such reliability value when the motion database is queried. A number of researchers also proposed to use multiple Kinect sensors to overcome occlusions [136].

Although the great effort of solving occlusion while using Kinect sensor, the issue still remains largely unsolved. In our proposed system, we explore some options for this regard, which will be addressed in the following chapter.

II.2.2.7.2 Interactive Feedback Mechanisms Providing real-time movement feedbacks to users is a very challenging issue for many Kinect applications, especial for those need physical rehabilitation exercise. During exercise, patients are required to perform movements in a specific patten to meet the objectives of the exercises. In order to monitor the quality of exercise, the recognition process should provide real-time feedbacks to patients to inform them about any incorrect movement.

Several feedback mechanisms have been developed for Kinect sensor since its inception. In 2013, Velloso *et al*. [137] presented the *MotionMA* system that can provide real-time feedback of body movement. This system creates a three-step communication loops: an extracted model

of movements; repetition of the movements by following the recorded demonstration during exercises; and automatically generating feedback based on the extracted model. Su [138] proposed an approach using the DTW algorithm and fuzzy logic for ensuring the quality of home-based rehabilitation exercise. First a patient performs a prescribed exercise in the presence of a professional. Then the patient's exercise is recorded for evaluation at home. The evaluation is used as a reference for the patient to validate exercise and to prevent adverse events. Saini *et al*. [62] presented an application that can provide stroke patients with instructions and guidelines to perform rehabilitation exercises. In this application, the determination of correct movement trajectory and speed are based on subjective evaluation of the professional.

Although above applications in some content provide feedbacks, the results show that they are not real-time feedbacks. They are the results of evaluation for performance of exercises. The real-time feedbacks are still not well addressed in the reviewed applications. In our proposed application, the real-time is addressed and developed. The main benefit of this effort is to provide patients the directly communication with the medical professionals for prevention of adverse exercise and improvement for effectiveness of the rehabilitation exercises.

II.2.2.7.3 Real-Time Motion Tracking The another big challenge for many computer applications using Kinect sensor is performance of recognition. For real-time motion tracking and analysis, the algorithm used must be commensurate with the speed of motion, *i.e.* the frame rate of the camera (30 frames per second for Kinect). The key criteria on determining which algorithm to use, is often time the selection of computational complexity. That is also represented by the efficiency of segmentation and image pre-processing.

Miranda *et al*. [51] proposed to track motion by sequentially identifying a set of extreme poses instead of regular sequences for the performance purpose. This approach is based on a decision tree algorithm that stores key poses at root and represents gestures at leaves. Hence the simplified the search or in another word, the shorter path search, is efficient enough to be used for real-time motion tracking. In another study on human body fall detection, Mastorakis and

Makris [68] used Kinect depth images instead of skeleton data to track body motion. The reason of using depth image is that approach outperforms the articulated model that requires significantly more computational power.

In our proposed system, we use many techniques to reduce computational and image processes aiming for improvement of real-time recognition performance.

In this section, we present a comprehensive survey on the applications of the Kinect technology, and the latest algorithms used for human body movement tracking and recognition using Kinect sensor. In Section II.2.1, we review the applications of the Kinect technology in a variety of areas, including human computer interaction, human gesture and body motion recognition, healthcare, humanoid robotics, speech and sign language recognition, retail services, workplace safety, as well as 3D reconstructions. In Section II.2.2, we provide an overview of the algorithms and mechanism used for depth sensing method and skeleton extraction using Kinect SDK, as well image segmentation and motion tracking algorithms that have been used in many Kinect applications. Finally, we discuss a few challenging issues related to the Kinect technology.

II.3 Related Work

The research literature reviewed in this chapter is related to computer vision-based human motion tracking and analysis. Also we are only interested in sequences of images acquired from a single video camera, not multiple cameras. The human body movement analysis is based on analyzing body motion in a larger scale rather than on studying of a local moving pattern such as hand gestures or facial expressions. The mechanisms, techniques, and algorithms reviewed in this section are categorized into 1) clustering and segmentation algorithms; 2) recognition and analysis mechanism used for human body movement; 3) image and data processing algorithms; 4) specific human body movement analysis using Kinect Sensor.

II.3.1 Human Body Motion Recognition Algorithms

The early research work of vision-based recognition of body motion has focused on tracking and labeling each part of body that performs some kind of actions. The tracking task is to determine the location and shape of body parts from image frames using the labels to identify them. In this section the algorithms related to motion or gesture recognition are reviewed.

II.3.1.1 Marker-based Motion Analysis

Marker-based motion-capture has been demonstrated in several interactive systems and prototypes. However these systems require obtrusive retro-reflective markers or LEDs [26] and many expensive camera setups. The marker-based human motion capture has so far mainly been used in the entertainment industry.

Obdrlek *et al*. [139] present an application using marker to detect human body movement. The procedure includes two steps: (a) the markers are manually mapped to the corresponding limbs of a predefined skeleton structure; (b) the limb lengths are estimated based on a training procedure. Then the procedure requires the subject to perform procedure. Then the procedure requires the subject to perform some movements, one joint at time (angles, knees, hips, wrists, etc.) to record the data.

The benefit of using reflective markers is the high accuracy of recognition. But this approach is at the cost of ease of deployment and configuration. Our research focus is on using markerless detecting technique. Therefore the more literature regarding markerless techniques are reviewed in next section.

II.3.1.2 Markerless Motion Analysis

With growing advanced sensor technology development, the computer gaming industry has introduced markerless motion tracking algorithm to provide game players to interact with computer device without wearing markers. The popular devices include the Nintendo Wii Remote,

PlayStation Move, and the Microsoft Kinect.

Change *et al*. [55] present a full-body 3D motion capture system and joint tracking capabilities without markers or handheld controllers. Their study has demonstrated that the certain body movement trajectories can be detected and tracked by Microsoft Kinect sensor in a very closer accuracy to high precision equipment OptiTrack. The experiment results also show that the time latency between outputs of sensor device and body movement of markerless system is in a negligible range with milliseconds.

In the early stage of development of computer version and graphics technology, the steady image sequences captured from video camera are able to be digitized. After that computer software can automatically analyze the sequences of the video image. There are two major tasks in resolving tracking body motion in video image without using markers, first is to detect the correspondences between sequential images, then second step is to track an object in a stream of sequenced images. The approaches used are 1) finding a match in a series action models performed by a person and stored in database; 2) recognizing different body parts, such as arm, legs, head, in a sequences of image. This technique is referred as labeling; 3) defining parameters used to configure the probabilities of an object movement in time frames. This modeling technique has been used mostly for human motion tracking.

Gould *et al*. proposed an application called Trajectory Primal Sketch (TPS) [140] to detect a motion by sudden changes, which are defined by computing the velocity curves v_x and v_y derived from the trajectory of a point. The presentation of contour of a set of velocity curves v_x and v_y can be used to distinguish basic motion like translation, rotation, projectile and cycloid. Modeling of the human body is another technique to study human motion. Leung *et al*. [141] presented approach that provides motion segmentation in a complex scene. This model consists of segments that usually connected body joints of human body. The figure generated from joints is called skeleton of human body. The model can be refined by using collections of component cylinders representing the spatial information of human body. Modeling of the human motion [141, 142]

is also developed to be an effective way to track human motion. The studies have shown that the forward motion is almost constant within a walking cycle. In computer vision, the plot of joint curves for a walking cycle, *i.e.* joint angles in a time series, can be used as a walking motion model. The curves provide sufficient information for determining of human posture. This approach can reduce the searching space during tracking by limiting the possible angle variation between frames.

Other approach for modeling motion is to use a sequence of stick figures, called key frame sequences that can roughly detect movement of the body. This key frame sequence consists of an ordered set of stick figures, each differing from its predecessor and successor, for instance, when a body segment has crossed or uncrossed another body segment.

Above studies provide fundamental base for tracking and recognizing human body movement using markerless approach.

II.3.2 Tracking Activities of Daily Life

The older population in USA grows steadily in recent years. Monitoring and learning of the activities of Daily Life becomes one of the most important area for research and development in health care industry due to the aging population. Using computer and other portable devices can be useful solutions.Recent study [143] made conclusion that vision-based program can help physical disable individuals in the realization of their daily activities, hence increasing their mobility and self-confidence.

One type of studies is to utilize mobile devices, such as smart phone, wearable watch, GPS sensor, and fingerprint sensor, for supporting personal health and social care solutions. Pires *et al*. [144]reviewed some existing applications in this study. The advantages of using mobile devices are data can be acquired from anywhere and at anytime, and the cost of devices are relatively low due to the wide availability of such devices (*i.e.* smart phone, smart watch, ...) already exists in the daily life of older people. Grünerbl *et al*. [145] introduced a smartphone sensor-based system that

use smart phone for assisting mental and psychiatric care. Their application specially addresses the need for a practical and collaboration of activity recognition and mental care. Altini *et al*. [146] proposed an application especially for daily activity related to physical fitness and rehabilitation, *e.g.*walking and walking speed using smart phone. The system demonstrated the effectiveness of position recognition by combining accelerometer and gyroscope data with smart phone location. The evaluation results show that the system can personalize walking speed estimates and reducing walking speed estimation error by calibrating sensor with smart phone usage. Yi *et al*. [147] presented another application using smart phone to detect falling of elderly people. The system can be used to enhance the reliability for remote health diagnosis.

Dimitrievski *et al*. [148] proposed an application with focusing on non-invasive sensors for the privacy of monitored individuals during their normal daily routines. The main contribution of this work is the technology for fusion of data originating from different sensors for recognizing activities and states in the environment. Dr. Heinz *et al*. [149] suggested potential computer based applications that can be used to support elderly people in independent living environments. They explored an approach using Skype technology to support older adult independent living. The Skype technology can add important visual information about how and what the seniors are doing to traditional verbal communication. The real-time interaction with healthcare professions are also provided in this system, which is called Skype chat. Lee *et al*. [150] presented an exploratory study that explores various algorithms to detect individual's walking of free-living in uncontrolled environment. Specifically, the algorithm can accurately recognize fast walking, such as descending stairs, or slow walking, such as ascending stairs, or slow walking of transitional activities.

Although the existing technology and algorithms are designed to recognize a set of multiple activities in above studies, the focuses are on specific activities, such as walking, climbing staircase, and etc. The sensors are used in previous studies are also limited to certain types, such as smart phone. Our research work is aiming on the integration of different devices, including Kinect sensor, smart phone, and wearable devices.

II.3.3 Real Time Feedback

It is very critical for the healthcare professionals to remotely track receive exercise updates or feedback. The functional status of exercise performed by post-injury and post-surgery patients, and people with disabilities and the elderly can help care provided about how the exercise is carried out, how efficient it is and what the impact it has on patient's progress towards healing. The real time monitoring can also ensure the technology is robust, reliable and accurate.

The research work conducted by Hagedor *et al*. [151] showed a remarkable increase in training performance if computer feedback is provided. The improvement that was up to 400% in the training specific performance. Furthermore, The data analysis from Parker *et al*. [152] study revealed that computer feedback, such as accuracy, metastability, rewarding, adaptability, and knowledge are all the key factor in theory-driven mechanisms during exercise based medical intervention. The application provided computer feedback via real-time 3-D images; a qualitative chart and a graph that is displayed on a lap-top computer.

Popović *et al*. [153]investigated the impact of feedback on quality of exercises. Scientists conducted intensive study the efficiency of feedback-mediated method compared to no-feedback exercise. The results demonstrated a higher level of acceptance by patients and positive effects of treatment from feedback-based program.

Although many studies have focused on providing assessment of exercises to patient or health professionals, there is lacking of study addressing the real-time feedback of how exercise performed and whether it is performed correctly or not. The system developed in this dissertation can provide real-time feedback.

Providing interactive feedbacks to users is an important part for many Kinect applications, especial those for physical rehabilitation purposes. During rehabilitation training, patients are required to perform movements in a specific manner to meet the objectives of the exercises. The recognition process should provide feedbacks to patients to inform them about any incorrect movement.

Several feedback mechanisms have been developed in Kinect research. Velloso *et al*. [137] presented the *MotionMA* system that provides real-time feedback. This system creates a three-step communication loop: 1) experts (medical professionals) demonstrate movements and the system extracts a model for such movements; 2) novice users (patients) repeat the movements by following the recorded demonstration during exercises; and 3) *MotionMA* automatically generates feedback based on the extracted model to help patients assess and improve their performance in real-time.

In order to reinforce the quality of home-based rehabilitation exercise, Su [138] proposed an approach using the DTW algorithm and fuzzy logic for ensuring that the exercise is performed correctly. First it allows a patient to perform a prescribed exercise in the presence of a professional. Then the patient's exercise is recorded as a base for evaluation at home. The evaluation can be used as a reference for the patient to validate his or her exercise and to prevent adverse events. A summary report of the evaluation may also be uploaded to a cloud service for physicians to monitor the patient's progress and to adjust the prescription if necessary. The DTW algorithm is used to compare two sequences that have different time lengths in order to determine the similarity between the supervised and at home exercises. To accommodate the fuzziness of the correctness criteria, fuzzy logic is employed to build a fuzzy inference of the physician's subjective evaluation.

For the rehabilitation exercises, the determination of correct movement trajectory and speed is traditionally based on subjective evaluation of the professional. Saini *et al*. [62] presented a system that provides stroke patients with instructions and guidelines to perform rehabilitation exercises. A centralized system stores all the patient's data that will be used to analyze the patient's movements and activities. The biofeedback system allows the user to give feedback to the therapist, and the therapist will evaluate the patient's movement and give further instructions. The biofeedback is displayed to the patient as well as the therapist. The main benefit for biofeedback is that patients can directly communicate with the therapist to improve the effectiveness of the rehabilitation exercises.

II.3.4 Cloud Computing

Cloud computing technology is one of the most popular Internet technology in recent years and its applications have grown tremendously among computer software development. The benefits of using cloud computing are significant for the proposed system for a number of reasons: (1) globalization allows people to access a centralized computing server or services worldwide as long as there is an Internet connection; (2) allowing remotely access to real-time data collecting from local server (3) the cloud computing environment is on the Internet; therefore, users access the cloud application through a computer Internet browser, not from the desktop computer operating system.

The development of cloud computing has provided a great opportunity for end-user moving individual resources to a centralized system [154]. The trend is to allow application to store, process, and execute on anywhere regardless of its whereabouts, where most cases the computing resources are scarce or are so difficult to maintain. The advantages of using cloud service are generally seen flexibility and scalability of operation cost, reducing system hardware cost, and improving productivity.

Cloud computing offers several basic models,including Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The SaaS provides software services as needed, which is type of service we uses for cloud-based database. In a PaaS model, the operating system is provided as needed. The users have to install or configure their own system. The users of this service can then build their software using the platform that is offered. However IaaS model, only computing system is offered as the service in the term of physical or virtual machines. On this system, the user has install the operating system of their choice, or configure their own computing environments and run software on them.

In our system, a cloud-based database server(SaaS) is utilized for storing, processing, and graphically illustrating user logging information. An mySQL instance from Amazon Web Services is used for this application. The local Kinect server needs makes a connection to mySQL and sends

data through client API by TCP communication channel. The detail design will described later.

II.3.5 Rule-based Motion Recognition

Motion recognition in general involves two process. First is extracting the features from data source. Second process is to understand semantic of the motion based on extracted features, which is also called human motion recognition process.

Machine learning methods have attracted great attention in performing human motion recognition. Most techniques used in recognition require very large training and validation data sets, which sometimes are manually turned. Such processes are computation expensive and unintuitive. However for simpler activities and gestures, algorithmic (i.e., rule-based) methods are often used [155], which the rules are defined in a text file. In this study, we focus on rule-based recognition method for its robustness and simplicities.

Rule-based method in human motion recognition is to build a set of rules for key poses and the sequencing of the poses. Hence definitions are used to identify each gesture or activity. It become very popular in healthcare and gaming applications in which the human body movements are often well defined and relatively simple [15, 156].

Hachaj *et al*. [157]proposed an approach to use knowledge rules to recognize the human body gestures and motion. The rules are defined in text file and are interpreted by gesture description language (GDL). Using Kinect based image frames, they tested their approach on several gestures, such as waving hands, clapping, hips motion. The experimental results have approved that their approach is reliable tool for recognizing human body static poses and body gestures.

In our study, the ultimate goal is to provide a computer vision-based system that can tracking and monitoring human body movement in real time using rule-based recognition technique with interactive feedback.

CHAPTER III

INTEGRATED HUMAN ACTIVITY TRACKING AND MONITORING SYSTEM

The ultimate goal for research presented in this dissertation is to develop a computer vision-based application that can track and monitor human activity using rule-based recognition technique in real time with interactive feedback. In this chapter, we present the detailed design and implementation. Tracking of the activities of daily living (ADL) plays a very important role in our life, especially for elderly and disable people. Those people normally are under a constant surveillant or monitoring system from professional health care-giver. Using computer based system allows health-care professionals to monitor their activities and instantly receive updates regarding the functional status of their daily life. In order to accomplish this goal, we specially developed a computer application that can monitor health-care worker activities in the room with patents and provide interactive feedback of assessment about their performance. The detail of system design and implementation is presented in rest of this chapter.

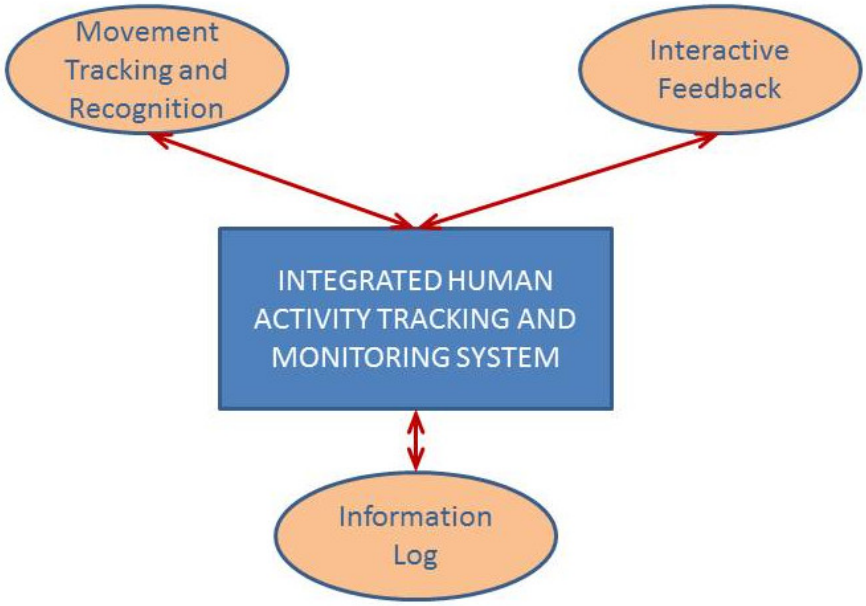

Figure 7: Integrated Human Activity Tracking and Monitoring System Architecture

III.1 System Overview

The developed integrated computer software can provide service to track and monitor human activity and interactive feedback for health care professionals. In this section, we present overview of system design and implementation. Our system consists three major components: human activity recognition, real-time feedback, and data collection program. The Figure 7 shows a high-level overview of the software architecture of integrated human activity tracking and monitoring system. The main components are activity tracking and detection, interactive feedback and information logging.

III.1.1 Human Activity Recognition

Human Activity Recognition is the core component in our system. It is a computer server that connects to Kinect sensor. It consists functionalities used to 1) track human body movement from image frames collected by Kinect sensor; 2) analyze image and detect body movement; 3)

recognize the movement based on predefined rule.

The Kinect sensor, a motion sensing input device, contains three major components, color VGA video camera, depth sensor, and multi-array microphone. The color camera collects red, green and black image data for facial recognition and other detection features. The depth sensor is built with an infrared projector and a monochrome complimentary metal-oxide semiconductor sensor. These two pieces work together for directly fetching the depth of users, in X, Y and Z positions regardless of the lighting condition. With Kinect SDK, the computer software developers are able to develop sophisticated computer-based vision tracking applications. In this processing component, data stream from one or more Kinect sensors is received through Kinect SDK library. Then it starts recognition process to find out if movement is performed correctly or not based on predefined rule.

In the second phase of image process in *Human Activity Recognition*, the skeleton image collected from Kinect is processed. The angles between body limbs are computed and compared to the rules for a specific movement. Hence the any non-compliant movement can be determined according the angle tolerances defined in the rule. The result of detection is communicated in real-time with user.

III.1.2 Interactive Feedback

Interactive Feedback component consists functionalities used to 1) allow user to register with Kinect sensor; 2) notify user if incorrect movement detected. The registration is a process that identifies and tracks a consented worker. It has a specific communication process between wearable device, mobile application and Kinect sensor. The real-time human activities are detected for proper movements. If any non-compliant activity is detected, a notification is generated and sent back to user through wearable device. The application program interface produces a vibration on device for alerting the user.

III.1.3 Information Log

Information Log component is the logging process consists two forms of information about human activities. One form is to store logging information into Extensible Markup Language file, and other form is mySQL database. The information can be saved to either local computer or cloud-based database. The log data files can also be integrated into a centralized database. A particular log includes the information about non-compliant activities, such as when a session starts, session duration, activity starting time stamp, activity duration, user id, etc. The XML file resides locally on the same location with Kinect server. However the cloud-based database resides in AWS, it requires a remote TCP connection established by Kinect server.

In addition to data log files, this component provides a graphic user interface for visualization of activities logging information. It shows the statistical information about activities, such as on what particular day, how many non-compliant activities found, average duration of activities, during which time period. The information on selected user or on all users can be reviewed as well. This program is running on a centralized server, therefore the user activities logged from different Kinect server can be centralized and monitored by an authorized administrator.

III.1.4 Hardware Components

There are three primary devices used in our system, Kinect sensor, wearable devices (smart watches), and mobile device(smart phone) that can communicate with wearable device. The Kinect sensor captures image of body movement with frequency of 30 frames per second. The skeleton image that contains 20 human joints (described in II.1) can be obtained via Kinect SDK. The user registration and interactive feedback are handled by smart phone and wearable device (i.e. Pebble device in our system).

The communication process of our system consists the two types of inputs, one from Kinect server, and the other from HTTP requests between wearable device and mobile phone. One important component of our system is the mobile app, which bridges the communication between the

Kinect server and the watchapp, and in fact, it integrates the Kinect sensor and the wearable device. In addition, we describe in details of the programming operation of mobile app. Integrating a cloud-based database with Kinect sever provides more reliable and secured data storage for logging information of detected non-compliant activities.

In following sections, we present the detailed system design and implementation for hardware and software components. The primary hardware devices used in our system are Kinect sensor, smart phone and wearable device.The software components are human body movement recognition, real-time feedback and interactive data logs.

III.2 Software Components

The proposed integrated human activity tracking and monitoring application, it primarily consists three components: human body movement recognition, real-time feedback, and interactive data logs.

The tracking and monitoring human activities is core functionality of our system, which is rule based gesture recognition. It is responsible to detect human body movements based on predefined rule. Each movement rule defines a sequence of how body segments move during activities, which is determined by key body segments. The movement recognition is based on error of key segments between rules and actual movements. The design detail is demonstrated in below Section III.2.1.2.

The real-time feedback provides information regarding non-compliant human body movements detected by Kinect sensor. The feedback system is interactive during body motion, and the quantities and duration of such activities are also reported back to user as well. Real-time notification is generated and sent back to the client end device through a smart-phone type of device, which will produce an action-aware event on the wearable device(e.g. Pebble watch) to alert the user.

The real-time logging process can be performed either on local server or on remote server.

Particularly, information about non-compliant activities is called a *log* that includes activity session, user information and duration. Logs can be saved into cloud-based database system and a local file with XML format.

III.2.1 Human Body Motion Recognition

The human body movement recognition process is implemented on server component that is connected with Kinect sensor. The Kinect server maintains a communication process with Kinect sensors through (1) initiating a device instance for each connected Kinect sensor; (2) receiving skeleton frames from sensor; (3) recording user's biometrics; and (4) tracking active user's status. The programming flow control of Kinect input handling is shown in Figure 8.

III.2.1.1 Programming Interface for Kinect Server

The programming interface for Kinect server is a program module that implements communication between Kinect sensor, server, and wearable device. It receives image data from Kinect sensor, and sends data messages to wearable device for invoking services or querying status. It provides the interface suitable to the Kinect device so that its image frames can be processed in server, as well as the interface to wearable device that can receive or send message description through smart phone application. The communication calls are based on network data messages description.

The primary programming interface for Kinect server consists two components: a) communication between the Kinect sensor and server; b) HTTP interface between wearable device (i.e. a Pebble Watch in our system) and server through a mobile device, which is a smart phone in our system.

The programming interface is embedded in a program class called *KinectCoordinator* that is used for communication between *KinectDriver* and Kinect server. *KinectCoordinator* class consists of a list of instances of *KinectDriver* class for each Kinect sensor connected to server, a list of *User*

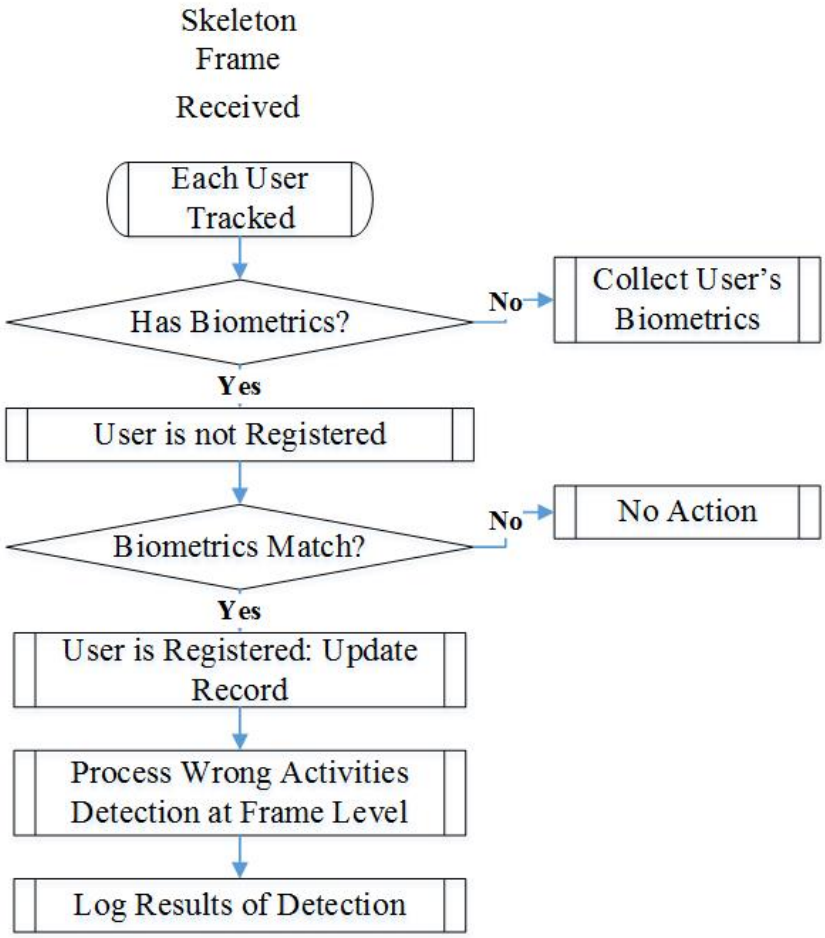

Figure 8: The flow control for Kinect input process.

class that is used for Kinect sensor capturing each user's biometric information, an *ActivityLog* class that is for tracking and storing user's activities. Figure 9 shows the diagram of Unified Modeling Language (UML) of *KinectCoordinator* class.

KinectDriver is an object class designed to manage communication between server and Kinect sensor. This class of four major components: 1) server interface; 2) Kinect sensor object; 3) user object; 4) data logging object. And component is also an independent class object that implements specific tasks, which will describe in below sections.

1. The server interface object in *KinectDriver* implements communication between Kinect sensor and server object, i.e. *KinectCoordinator*. The Kinect server sends user's skeleton image data to *KinectCoordinator*. Server enumerates the Kinect sensors that are connected to server i.e.

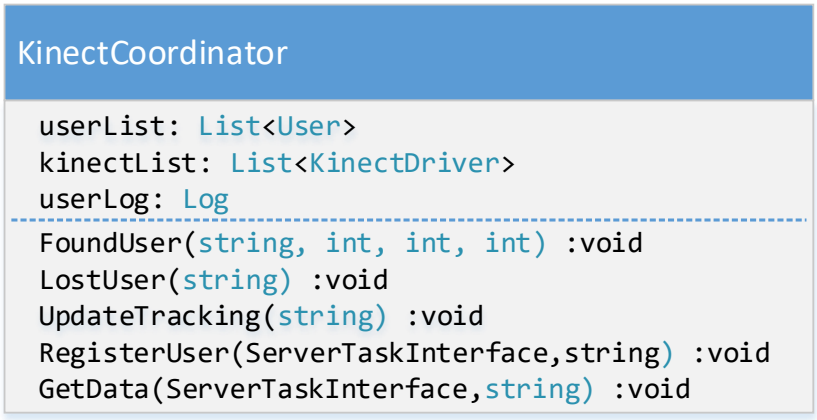

Figure 9: Kinect Coordinator Class Interface

computer using the *KinectDriver* class. This class is initiated when a Kinect sensor is connected with server that is started by this application. Each Kinect sensor initiates one object. The static property of Kinect sensor API contains the collection of Kinect sensors that are connected with server. Because this property is static, the collection of Kinect sensors is available when server is running. Hence if multiple Kinect sensors are connect to server, then a list of *KinectDriver*s is created.

2. *KinectDriver* is created by system during initialization process. Hence the object is able to receive human body skeleton image frames from Kinect sensor using programming application interface provided by Kinect Software Development Kit. The function consists of two parts: first it finds an attached Kinect sensor, then initializes it and prepare to read image data from it. One important step in the initialization process is to enable data streaming. The Kinect sensor API provides four types of image data that can be streamed out color, depth, skeleton, and infrared. To enable each of the image streams, the stream enable method must be called with specification of the image data format, the frame rate, and the resolution of the pixel data. For example, below shows how to enable Kinect to generate color stream data with RGB data, at a 640x480 resolution, at a frame rate of 30 frames per second. To stream depth data call the enable method with same specification as color image: the image data format, the frame rate of 30 frames per second, and

the 640x480 resolution. There is no specification needed to stream skeleton data, which enables the Kinect to generate skeleton data. The stream data is activated by defining following statements in the initialization process.

- sensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30): this statement enables color stream with solution of 640x480 and speed at 30 frames per second.
- sensor.DepthStream.Enable(DepthImageFormat.Resolution640x480Fps30): this statement enables depth stream with solution of 640x480 and speed at 30 frames per second.
- sensor.SkeletonStream.Enable(): this statement enables skeleton stream.

Once initialization process is completed, in our application*KinectDriver* calls Kinect sensor.Start() that is provided by Kinect sensor API (Kinect SDK version 1.8). It starts the sensor and associated threads. And this call starts streaming data out.

3. User object is used to track all information regarding an user. This class consists of user activity result, Kinect sensor's index that user is tracked by, tracking status, biometrics data, and activity logging data. The tracking is pursued by two steps: 1) only track temporary behavior for responding to mobile app request; 2) track aggregated data at server side through a blue tooth designed id. The wrong activities are tracked by a simple binary finite state machine that is alternated by two values, "normal" and "abnormal" states. The implementation details for tracking and monitoring human activity process are provided in Section V.

4. Data logging object is used to track and store user's activities logging information. Particularly, it tracks user's starting and ending time for a registered session, type of activity, and an activity's starting and ending time. This class object also can export the activities to a text file or send information to a cloud-based database. Again the detail of implementation of logging process is described in future section.

III.2.1.2 Tracking and Monitoring Human Activity

In our system, using the rule-based approach to track and monitor the human activity is the primary functionality. It detects and assesses individual human body frames and movements with the pre-dened exercise rules. Each exercise movement rule defines a sequence of body segments during during activities. And the sequence is associated by a key body segment. The correct movement recognition is evaluated upon the variance between rule data and actual movements.

We use Kinect SDK programming interface for developing recognition module. The image frames provided by Kinect SDK program are color frames, depth frames and skeleton frames. In our system, we are particularly interested in skeleton frames. On receiving a new skeleton frame, the Kinect runtime dispatches the frame to a designated thread for processing. The *KinectDriver* first makes a copy of the skeletons received, and then examines each user who is currently tracking. If a user is registered and his or her biometrics recording has not been completed, the *KinectDriver* adds this user to the list. Subsequently for registered user, the average values of the biometrics are stored for further processing, not just a single frame value. Using average values instead of a single frame for computing user's biometric information can increase the accuracy of recognition.

If the user is registered and biometrics has already been taken, the *KinectDriver* examines skeleton with all other users who are not currently tracked to see whether their biometrics are matched or not besides current registered and actively tracked user. If a user is identified this way, the status of that user is changed.

After a user is identified as registered, the *KinectDriver* then computes the single-frame level wrong activity detection by invoking the *Activity Recognizer*. What activity is considered to be non-compliant is defined in the Configuration file. The result of detection is reported back to *KinectDriver*. Each user being tracked has a corresponding *User* class, which in turn performs an activity-level wrong activity detection. All detected wrong activities are recorded in cloud-based database and XML format file, which process will be described in Chapter VII.

The HTTP interface between wearable device (i.e. a Pebble Watch in our system) and server

through a mobile device, which is a smart phone in our system. HTTP communication for Mobile and Wearable Devices)

As shown in Figure 10, the communication between server and wearable device is performed via HTTP requests and responses. It includes a mobile app that handles request and response between server and smart phone, and a watch app that handles interaction between smart phone and wearable device. The wearable device (i.e. Pebble Watch) provides an API program allowing bi-directional communication between phone apps and watchapps. The user initiates registration request by pressing a button on watch. The mobile app receives the request and sends it to server. Once registration request is received by server, a *ServerTask* class object then is created to process the message. The *ServerTask* distributes HTTP request to another class *Kinect Coordinator* object. First *Kinect Coordinator* determines whether the user has been previously registered or not. If so registration request is forwarded to the *User* class to identify. The use's biometrics is used to examine. Upon completion of identification, a success response is sent back to mobile app immediately. Otherwise, a gesture recognition is performed using the most recent copy of the skeleton. The user gets notification of registration status from wearable device, such as displaying the message on watch face.

In the scenario of registration request is for a new user, gesture recognition is performed by *KinectDriver* class. A notification is sent back to requester regardless of the detection result through mobile app and watch app.

Our system is designed in such a way that same registration request can be used for registering a new user and periodically query the status of a previously registered user. The mobile app sends a query to server periodically. The response received from server is parsed. If the message is any of status changing from registered to unregistered, from unregistered to registered, or alert, the mobile app sends the notification to watch app that shows the message on watch face.

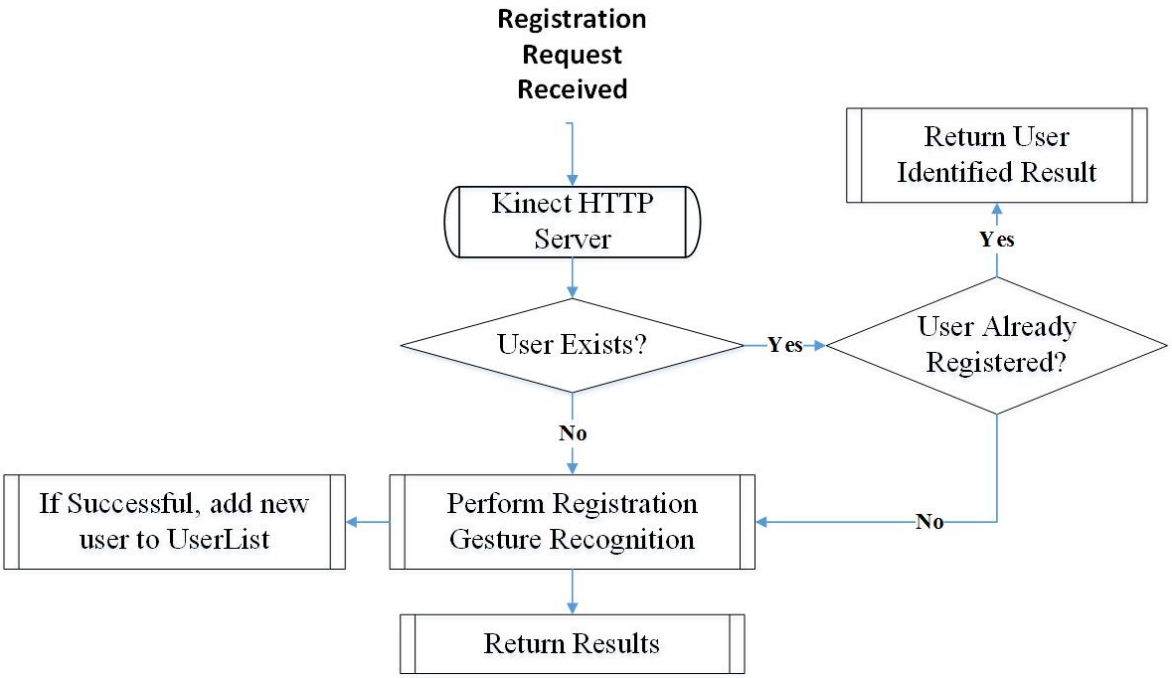

Figure 10: The communication processes handled by Kinect server.

III.2.2 Real-time Feedback

The real-time feedback provides interactive message exchanges between Kinect server and user's wearable device. It is implemented by two communication components: 1) HTTP protocol communication between Kinect server and a mobile device (i.e. a smart phone); 2) a mobile APP to wearable device (i.e. a Pebble Watch).

The information of feedback typically includes user's registration status, user's request of being registered, user's tracking status, and detection status of any non-compliant activities. An alert notifying user about how many times non-compliant activities have been detected is also generated by program.

The Kinect server first initializes a HTTP server with its local IPv4 address and a port (default is 8181). Then the mobile application can make an authorized connection with server. Once the connection is established, the message exchanges between Kinect server and mobile application are carried out dynamically.

The wearable device provides mobile application that allows exchanging messages between a mobile application and device. In our mobile application plays a role in bridging a way to connect Kinect server with wearable device thereby for communication between two devices.

The detail program design about real-time feedback is addressed in Section VI

III.2.3 Interactive Data Logs

Our application provides a functionality that can capture user's activity logs at different time frame and save them to a local device file or on-line data storage. This feature allows users or administrator to monitor and evaluate what participants perform during a certain time frame. The logging data also provides some insight of the users exercises do and statistical information regarding activities. Meantime, this application captures a screen shot where user's skeleton image is displayed when a non-compliant activity is detected. This feature provide a detailed information, typically a interest only when diagnosing problems, as well as of future improvements for detection accuracy.

The detail of context regarding interactive data logs is addressed in Chapter VII

CHAPTER IV

USER REGISTRATION AND PRIVACY PROTECTION

One of important objectives of our application is to monitor and track privacy-aware activity. However the Kinect SDK software provided by Microsoft is capable of tracking no more then six persons at a time, but only two persons actively. Allowing a designed user to sign-in with system before being monitored and tracked is needed so that only one person can be identified in Kinect sensor view range should be for tracking. In addition to protect the user's privacy and identification, we also provide a functionality that can obstruct facial identity on display screen.

In this chapter we address the importance of privacy-aware registration process and protection, overview system design, and details of implementation for each programming components.

IV.1 Importance of privacy-aware registration and protection

Our application is tracking and monitoring activity performed by a designed user. However the Kinect SDK software is capable of tracking multiple persons at a time. How to correctly identify a desired person to track and monitor and how to protect identity of that person in the view range of Kinect sensor become a challenging issue in our application.

The activity monitoring and tracking reveals a private information about a person. Such

information shouldn't be shared or kept without permission of the performer. In other word, using Kinect sensor to track one's activity must have a consent or being acknowledged from whoever participates in such activity. Thus before our application starts tracking activity detected by Kinect sensor, a consented agreement is required. The process of obtaining consent for tracking activity is called registration, which allows user to notify Kinect server that he or she is willing to be tracked.

Another aspect of protecting the privacy of a user is obstructing facial image on display screen during active section of tracking. The Kinect SDK provides three images: color, depth and skeleton images. The depth image is in gray scale. The skeleton image illustrates human body joint locations that can be shown as connected lines. Both depth and skeleton images do not show any actual human physical figurations. However the color image represented by RGB format, which shows actual image of captured players, including body and facial figurations. The requirement of critical data in our application is body movement, not facial, thereby the facial image is designed to be obstructed for the purpose of protecting an identity of a user.

The details of system design and implementation of privacy-aware registration and protection are addressed in the following section.

IV.2 Design Overview

The basic software components of a privacy-aware registration process are user registration request, evaluating registration gesture, and notification. The identity of user is obtained by Kinect server through a private self request from a wearable device first, and then user must perform a specific gesture that is predefined with rule. Kinect server then evaluates the received gesture via Kinect skeleton image. If it matches the registration rule, then a notification of successful registration will be sent back to user wearable device. Once user is registered with system, his or her biometric is saved, thus Kinect server can automatically identify user based on the biometrics. Figure 11 illustrates this process.

The registration process is to identify a user that Kinect sensor tracks. Only the registered

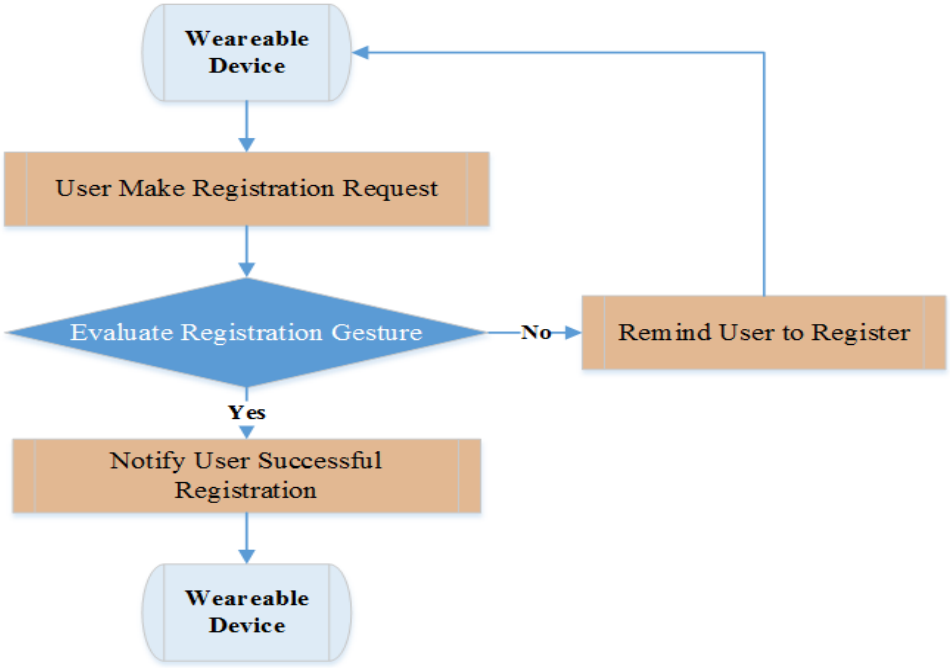

Figure 11: Registration Process

user is tracked and monitored by system. The one mechanism used in our system is *one-time registration* meaning the user only requires to register once with application. If user re-enter into the view sight of Kinect sensor, he or she can be recognized automatically.

The facial image displayed by Kinect's color image is obstructed for purpose of protecting the privacy of consented user who has registered with system. A gray color transparent rectangle is used to display on screen before user is successfully registered with system without showing any users' facial images. Once a consented user is confirmed, only user's skeleton image is shown, which does not have reflecting personal image.

The details of this feature are described in more detail in following Section IV.3.

IV.3 System Components

In our application, we developed a functionality that can protect user's privacy through consented registration and personal physical figure. There are two major components to accomplish

it: 1)privacy-aware registration process; 2) and protection of facial image. In this section, we present some details of implementation for this functionality.

IV.3.1 Privacy-aware User Registration

The privacy-aware user registration is the process that allows users to identify themselves and privately sign in with system. It involves with two stages, first time registration and automatically recognition. First the user must send Kinect server a request for registration from wearable device. If the registration gesture is recognized, this user's biometric is saved with system, which can be used for auto-register during active section. The first stage is called *explicit registration*, and second stage is called *biometric auto detection*. The second mechanism used in our system is *one-time registration* meaning the user only requires to register once with application. If user re-enter into the view sight of Kinect sensor, he or she can be recognized automatically.

IV.3.1.1 User Registration Request

The privacy-aware user registration process is initiated by user make a request from wearable device. A predefined registration gesture must be performed to evaluate it against registration rule. There are two status states designed for wearable device, which we choose Pebble Watch. Status states are defined as 1) registered; 2) unregistered.

On wearable device application, we provide a registration button for user to make request. This action is called *explicit registration*. After request is made, if the registration gesture is successfully confirmed and recognized, the user is completed registered. Thus the registration state becomes *registered*. In this state, the user does not need to register again even he or she walks away and comes beck the view range of Kinect server.

On receiving a registration request from the watchapp, the message is relayed to the Kinect server for evaluating the user's registration gesture. However after registration request and gesture is not recognized, the state then becomes unregistered. While in unregistered state, a reminder of

registration is displayed on wearable device for 10 times. The notification can be terminated by other three cases: 1) user selection; 2)user is successfully registered; 3) auto-registration occurs.

Making a registration request is the first step in the process of privacy-aware user registration.

IV.3.1.2 Register Rule

After request is made, user must perform a predefined gesture. This gesture is evaluated using rule-based recognition algorithm. The key component of rule-based body movement recognition is a predefined gesture rule that is called *RegisterRules*. In our application, register rule is used to describe the body joint. movement.

The registration process is to recognize a user with system. The user must perform a gesture that is defined by *RegisterRules*. *RegisterRules* is defined in term of bone orientation, which describes gesture's type, body joints, and angles between joints. In this particular application, a left arm gesture is used for registration. The left wrist and left elbow is forming a 90^0 angle.

The registration process is for recognizing a user who must perform gesture that is defined by *RegisterRules*. This rule describes the gesture that includes the type, body joints, and angles between joints. In this particular application, a left arm gesture is used. The left wrist and left elbow is forming a 90^0 angle.

The registration rule includes the following parts.

- Type of Gesture: Bone Orientation
- Downstream Joint: Wrist Left - joint
- Upstream Joint: Elbow Left - joint
- Transverse Angle between joints: 90^0

The XML format is used, which is represented with corresponding tags. A sample of XML format for registration rule is shown in following Figure 12:

```
<RegisterRules>
  <Configuration>
     <Type>BoneOrientation</Type>
     <AlertLevel>1</AlertLevel>
     <AlertBody/>
     <AlertVibration>0</AlertVibration>
     <DownstreamJoint>WristLeft</DownstreamJoint>
     <UpstreamJoint>ElbowLeft</UpstreamJoint>
     <FrontalAngle>-1</FrontalAngle>
     <SagittalAngle>-1</SagittalAngle>
     <TransverseAngle>90</TransverseAngle>
     <FrontalAngleTolerance>-1</FrontalAngleTolerance>
     <SagittalAngleTolerance>-1</SagittalAngleTolerance>
     <TransverseAngleTolerance>20</TransverseAngleTolerance>
  </Configuration>
</RegisterRules>
```

Figure 12: Sample of Registration Rule Defined in XML Format

The *RegisterRules* must be defined prior to before system starts. The user registration is very first step in our application. Without consenting from user, Kinect server can not start tracking for privacy concern.

IV.3.1.3 Evaluating Registration Gesture

As discussed above, after registration request is made, user must perform a predefined gesture. This gesture is evaluated using rule-based recognition algorithm. *RegisterRules* describes the body joint. The Kinect server evaluates received skeleton image with *RegisterRules*. The successful registration is evaluation result in the range of tolerance.

According to the rule defined in *RegisterRules*, the registration is achieved by recognition of a left arm gesture, which is the left wrist and left elbow is forming a 90^0 angle. This angle is computed by left wrist vector and elbow left vector, shown in Figure 13, and it requires to be 90^0.

Vector $\vec{a}$ is bone segment between points left shoulder and left elbow, and vector $\vec{e}$ is bone segment between points left elbow and left wrist. The Kinect server captures skeleton image at 30 frames per second. The skeleton includes human body 20 joint points that are in 3D coordinates.

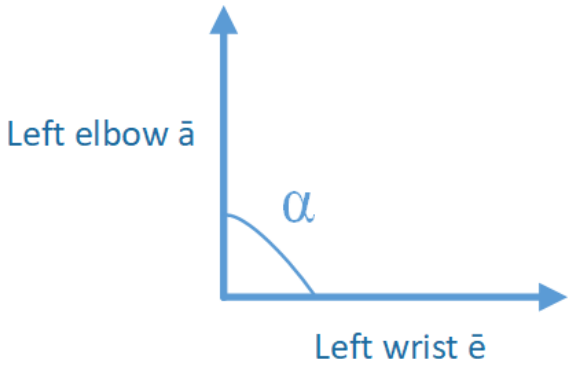

Figure 13: Angle Calculation Between Two Vectors

Given joints 3D points of left shoulder:*joint1*, left elbow:*joint2* and left wrist:*joint3*, the angle α is computed by (IV.1():

$$cos(\alpha) = \frac{\vec{a} \cdot \vec{e}}{\|\vec{a}\| \cdot |\vec{e}\|} \tag{IV.1}$$

where $\vec{a} \cdot \vec{e}$ is the dot product of vectors $\vec{a}$ and $\vec{e}$. And $\|\vec{a}\|$ is the length of $\vec{a}$, and $\|\vec{e}\|$ is the length of $\vec{e}$. They can be represented byIV.2:

$$\begin{aligned} \|\vec{a}\| &= \sqrt{(joint1.x - joint2.x)^2 + (joint1.y - joint2.y)^2 + (joint1.z - joint2.z)^2} \\ \|\vec{e}\| &= \sqrt{(joint2.x - joint3.x)^2 + (joint2.y - joint3.y)^2 + (joint2.z - joint3.z)^2} \end{aligned} \tag{IV.2}$$

If the α is in the tolerance range of 90^0, the gesture is recognized, otherwise return a false status. The registration gesture is the beginning phase of gesture recognition. The evaluation result is sent back to user's wearable device for displaying status. The notification process is addressed in following section.

IV.3.1.4 Notification of Registration Status

After evaluating the registration gesture, Kinect server can send the result back to user by displaying a message on wearable device. Thus user receives the notification for either registration is successful or not. There are three types of messages sent back to user for notification: 1)time out; 2)in correct registration gesture; 3) successful registered.

The Kinect server performs evaluation process of received user's registration gestures against predefined *RegisterRules*. Upon the evaluation result, Kinect server utilizes HTTP communication channel to send resulting messages to mobile application that transmits these messages to wearable device's APP.

The types of feedback messages for registration are listed as following.

- case 1: registration is successful
- case 2: registration request is failed due to timeout
- case 3: registration is failed due to unable to recognize gesture

If registration is failed, the mobile APP sends a series of reminder in a time interval until user to make another registration request or the number of reminders has reached a maximum count. The reminder time interval increases in multiplication steps, which is shown in IV.3 according a detailed explanation in [15].

$$remindertimeinterval = registrationremindercount * remindertimeinterval \quad \text{(IV.3)}$$

The detail implementation of this real-time feedback process is discussed in Section VI.2.3.

IV.3.2 Automatically Identify Previously Registered User

We provide a more sophisticated approach in our application for privacy-aware registration process, which is considering the scenario that a consented user might leave the sight of Kinect server and later come back. Should user initiate a new registration request or should system automatically recognize user's registration status in this situation? Obviously the auto recognition makes more sense. Considering how cumbersome or very impractical to require the user to make request and perform the registration gesture every time, *one time registration* strategy is a better practice in real time. Therefore we provide a feature that Kinect server can immediately recognize user once he or she has previously registered with system.

The automatically identifying an user who has registered with our system is implemented by a non-intrusive mechanism of using body biometrics according [158]. This approach records user's body physical information provided by skeleton image from Kinect sensor.

The SDK of Kinect provides image frames at a rate of 30 frames per second. The skeleton image composes of points in spatial coordinates of human twenty body joints. Each user has distinctive physical shape that are represented by lengths of body parts. Therefore biometric used to identify user can be said to compute lengths of body parts.

The lengths of body parts are calculated as a distance of two joints that are given from image frame of Kinect sensor. We captures 90 frames (i.e. 3 seconds) data for a stable user biometric data source. The average length of body parts over this time period of performing registration gesture is saved for user's biometric.

The used body parts are:

- neck: distance(joint(head),joint(shoulder center))
- spine: distance(joint(shoulder center), joint(spine))
- left arm: distance(joint(left shoulder), joint(left elbow))
- right arm: distance(joint(right shoulder), joint(right elbow))
- left forearm: distance(joint(left elbow),joint(left wrist))
- right forearm: distance(joint(right elbow),joint(right wrist))
- left thigh: distance(joint(left hip),joint(left knee))
- right thigh: distance(joint(right hip),joint(right knee))
- left leg: distance(joint(left knee),joint(left ankle))
- right leg: distance(joint(right knee),joint(right ankle))

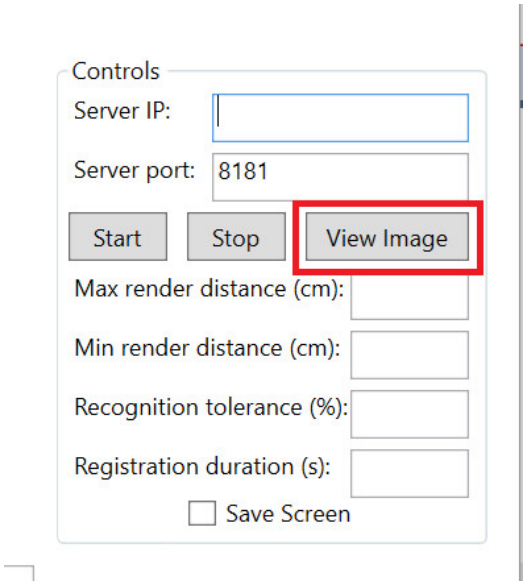

Figure 14: An Option of Display Personal Image

Each user's biometric information is recorded with system after registration gesture matches *RegisterRules*. In the scenario that user is out of sight of Kinect server and comes back again, a process of checking if this user has been previously tracked or not is automatically initiated. If this user's biometric is found, he or she would be auto sign-in. This process is also called *biometric auto detection*.

IV.3.3 Protection of Privacy of Registered User

In our application we focus on protecting what we present in a public area as a crucial part of maintaining personal privacy. Kinect sensor delivers the color stream data in RGB format, which shows a real image of anyone in sight of camera. Some users may have a concern of showing their personal image in a public platform without consensus. Considering the protection of privacy of a registered user, we provide a dynamic option for user to choose viewing the real personal image or not. This functionality is represented by providing a selection button, shown in Figure 14. The system starts with default setting of not showing graphic image. However images can be shown by clicking on *view image* button.

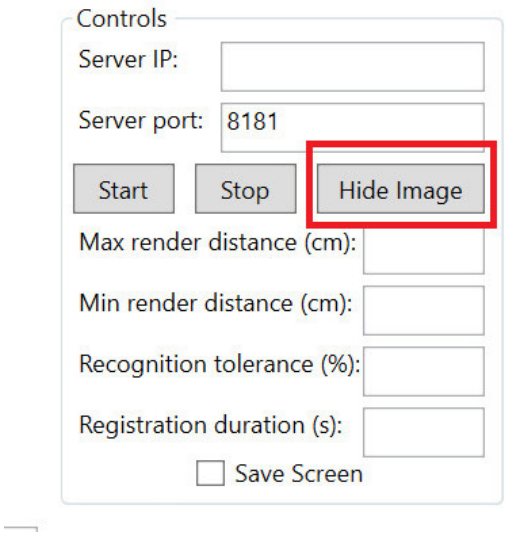

Figure 15: An Option of Hide Personal Image

After the user is registered with system, by default the user is shown as a single-color silhouette instead of color image. User has a choice to view real image or not by clicking on *view image* button. Then the button becomes *hide image*, shown in Figure 15. The image can be hidden by clicking on button again.

The image overlay is implemented by laying an additional image object over color image provided by Kinect sensor on screen. The overlay image is defined with same dimension as display area.

The registration process is to allow system to identify a user to sign-in with Kinect sensor. Since Kinect can track multiple users in the image, therefore our system needs to know which user should be tracked and monitored. Kinect SDK provides functionality to record user's biometrics, such as facial features and body shape. So that it can distinguish a registered user from other unregistered users when multiple people shown in front of device. Once a user is registered with device, tracking and monitoring of that specific user is implemented by using a user id provided by Kinect sensor application interface, and the registered user will be identified with same user-id in system permanently. Hence the user can be tracked and monitored automatically whenever he or

she walks into device image capturing range.

In this chapter we describe the privacy-aware registration process. It requires a predefined rule for performing registration gesture. The gesture is called *Registration Gesture* that is defined by *Registration Rule*. The *Registration Rule* defines human body joint movement for registering user's biometrics with system. The successful registration is confirmed with recognition of correct *Registration Gesture* performed after a designed request. Privacy-aware process tracks and monitors only registered user and allows an automatically recognition of user every time he or she re-enters back to sight of Kinect sensor, which is called a one-time registration mechanism. To prevent a personal image published on a shared working environment, we provide a feature for user to choose if the real image displays or not, which is fulfill the complete requirements of privacy protection.

CHAPTER V

ACTIVITY DETECTION AND MONITORING

The research of tracking and monitoring human activities using computer software has attracted broad attentions in recent years due to increased aging population and higher demand of better services in rehabilitative health care. The tremendous interest also motivates us to develop software system that can be used to efficiently track and monitor human activities. The software system presented in this dissertation is an application that can meet such demands.

Tracking and monitoring human activity process is one of major functionalities in our system. We adopt the rule-based approach to accomplish this task. The idea is to detect and assess individual human body frames and movements according the predefined exercise rules. Each exercise movement rule defines a sequence of body segments during activities. And the sequence is associated by key body segments. The correct movement recognition is evaluated upon the variance between rule data and actual movements.

In this chapter the design strategy of accurately and efficiently tracking and monitoring activity is reviewed in detail. The rule-based algorithm that is adopted for activity recognition is also described. In order to ensure the quality of exercise or activity movement, we develop a real-time feedback system that can notify user about any non-compliant activities. The principle of detection for ”abnormal” movement is explained in this chapter as well.

V.1 Design Overview

In this section we describe the design of the module that can track and monitor human activities based on predefined rules. The component that implements tracking and monitoring human activities is designed as a class object in our system. It detects and assesses individual human body frames and movements with the predefined exercise rules. Each exercise movement rule defines a sequence of body segments during activities. And the sequence is associated by a key body segment. The correct movement recognition is evaluated upon the variance between rule data and actual movements.

This module contains four main components: user skeleton data, activity logging information, and rule configuration. The user skeleton data contains 3D position data for human skeletons, which 3D coordinates are the body axes (x, y, and z) of the depth sensor as shown in Figure 16. This is a right-handed coordinate system that places a Kinect at the origin with the positive z-axis extending in the direction in which the Kinect is pointed. The positive y-axis extends upward, and the positive x-axis extends to the left. Placing a Kinect on a surface that is not level (or tilting the sensor) to optimize the sensor's field of view can generate skeletons that appear to lean instead of being standing upright [2].

The activity logging information is used to track and store user's activities logging information. It tracks user's starting and ending time for a registered session, type of activity, and an activity's starting and ending time. The finite state machine maintains a status that indicates that detected activity is "normal" or "abnormal." Lastly, the rule configuration data comes from IO stream of text file that defines exercise rules.

V.2 Detection of Non-Compliant Activities

In our application, the tracking and monitoring of the user's activity are to detect any non-compliant movements performed. This process is persistent for a registered user. The detection is

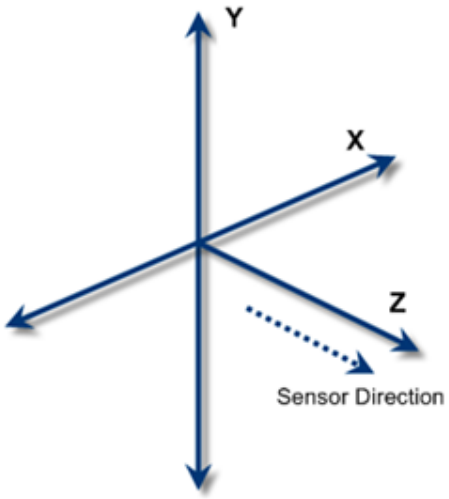

Figure 16: Skeleton Space, provided by [2]

based on the rule-based algorithm. How to use the rule-based algorithm for assessing the correctness of human body movements has been studied by Zhao *et al.* in several literatures [15,156,159]. In this section we discuss the detail of using this approach to track and monitor body movements.

According to research work from Dr. Zhao *et al.*, the rehabilitation exercises rules are defined in three categories: dynamic rules, static rules, and invariance rules generally. The latest is our focus in this study. An invariance rule defines that moving human body segments must be matched during each iteration of the exercise [156]. The rule is particularly described as the relative angle between the moving body segment and the frontal plane, sagittal plane, or horizontal plane.

The process of detecting consists following steps:(1) retrieve rule configuration; (2) receive body joint points; (3) calculate angles; (4) calculate variances; (5) determine the correctness of movements.

V.2.1 Rule Configuration

The rule used in detecting non-compliant activity is defined in a markup language Extensible Markup Language (XML) format, which is called *InvarianceRules*. The XML format is a universal structured data format. The advantage of using XML format for rule configuration is its unique

features of easy to understand, extensibility, and readability. XML format also makes it easier to compatible with other system because it stores data in plain text format. It provides software and hardware independent way of sharing, transporting and storing data. It can be easily expanded or upgraded to new operating systems, new applications, or even new browsers, without losing data.

Comparing to other format storing configuration data, the XML has very special character that is it has no predefined tags. The rules used for exercises can generated different tags according movements specifications. The rules defined in XML can be extensible without application software modifications. The new data are able added to rule configuration as needed, even exchangeable with older version of software. Therefore XML format is an ideal for configuring exercises rules.

There two major XML definitions used to define exercise rules in our application. The first one is *RegisterRules* and second one is *InvarianceRules*, which are described in detail in following sections. The *RegisterRules* is described in a large detail in Section IV.3.1. The *InvarianceRules* is used to evaluate the correctness of activity. Both rules are defined in terms of the body bone orientation.

The rule that describes the tracking gesture is non-compliant, is in *InvarianceRules*. This rule includes the type, body joints, and angles between joints. In our particular application, two body gestures:1)bend; 2) bend and twist, are used.

The bend gesture is defined as the body joint of hip center and body joint of shoulder center are aligned straightly. It means if the angle between hip center and shoulder center is beyond 180^0, that gesture is considered as incorrect or *invariance*.

Similar, the bend and twist gesture is defined as the body joint of hip center and body joint of shoulder center are aligned straightly with additional twisting rule of the body joint of hip left and body joint of hip right are aligned straightly. It means if the angle between hip right and hip left is beyond 180^0, that gesture is considered as twisting and is *invariance*.

Thereby, the invariance rule includes two *InvarianceRules* described in Figure 17. Given

predefined rules, next it is the process for using such rules for detection of non-compliant gestures performed by users.

V.2.2 Gesture Recognition

The detection of non-compliant activity is implemented by a process of gesture recognition. The focus of this section is to demonstrate how to use user's skeleton data to calculate joint angles that are aligned with rules defined in configuration file. The process of detection consists following steps:1)retrieving predefined rule configuration; 2) receiving body joint points from Kinect sensor image stream; 3) calculating angles of body alignments; 4) calculating variances; 5) determining the correctness of movements.

The *InvarianceRules* used in detection of non-compliant activity is predefined in a XML configuration file, which is described in above Section V.2.1. This file must resides in the same location with programming executable code. The rule configuration is parsed using *XDocument* function provided in .Net framework. The XML elements are parsed into structured data. All attributes of rule configuration are saved in configuration class for gesture recognition.

The Kinect SDK software library provides object that can directly interacts with the camera data steams. One of the most important functions of the Kinect SDK is tracking the human skeleton. It can track up to twenty joints in a single skeleton and return joint positions for two of the tracked players. The software receives image at a rate of 30 frames per second. Each joint position is represented by 3D (x, y, z) coordinates.

To access skeleton joint data, the SDK provides event handlers for image frames. Particularly, we uses *AllFramesReady* for retrieving skeleton joints coordinates. The *AllFramesReady* event carries color, depth and skeleton images data in three data objects respectively for access. For our gesture recognition mechanism, only *SkeletonFrame* object is needed. This object contains skeleton data object for each tracked user who is the registered with our system. Each skeleton consists twenty joints 3D coordinates, which can be obtained by applying *JointType* enumeration

to skeleton. A sample of obtaining left shoulder and right hip joints 3D coordinate is shown in Fig 18.

The most important function for gesture recognition is evaluating gesture with predefined rule. The angle betweens specified body parts are used for measurement. According rule defined in *InvarianceRules*, the non-compliant gestures are bend and bend twist. Therefore once a detected gesture that matches the angle defined by rule is considered as wrong. Similar to the gesture recognition for user's registration process, the detection is achieved by recognition of examining an align between hip center joint and shoulder center joint for bending, and an align between left shoulder joint and right shoulder joint for twisting. The examining angle requires to be 180^0.

Vector $\vec{a}$ is bone segment between points hip center and shoulder center, and vector $\vec{e}$ is bone segment between points left shoulder and right shoulder.The angle α is computed by IV.1. If the α is not in the tolerance range of 180^0, the non-compliant gesture is detected. The process is persistent during an active session. The evaluation result is sent back to user's wearable device for displaying status.

In this section, we describe how to detect non-compliant activity. For our particular application, bending or bend twisting is considered as wrong activity, which is required to detect during active session.

V.3 Overlaying Skeleton Image on Color Image

In our application, there is another very interesting feature we provide for illustrating activity detection and monitoring is laying the skeleton image on a gray-out body color image. Figure 19 shows an example.

The layering skeleton image over real-time color image from a tracked user can illustrate how human body joints are formed for a particular gesture. This feature can help us to validate efficiency of recognition, to make sure the detection is correct. The implementation is carried out by drawing skeleton joints on the graphic canvas that is drawing object of color image.

V.3.1 Skeleton Layout

Skeleton layout is a graphic object we use for drawing human body joints. It is an inheritance of a .Net framework grid object that allows having child elements. Each element can represent a graphic shape, such as rectangle, triangle, polygon, etc. A grid object ban be laid on a panel that is a basic graphic component.

Tp draw a human skeleton, the skeleton data captured by Kinect sensor is divided into several body parts, and one part is considered as one child of grid. In our application particularly, body joints are grouped in five parts, just for convenience of drawing. These five graphic elements and the joints included in are listed below:

- head and main torso figure element, includes head, shoulder center, left shoulder, spine, right shoulder, and shoulder center.
- left leg figure element, includes hip center, left hip, left knee, left ankle, and left foot.
- right leg figure element, includes hip center, right hip, right knee, right ankle, and right foot.
- left arm figure element, includes left shoulder, left elbow, left wrist, and left hand.
- right arm figure element, includes right shoulder, right elbow, right wrist, and right hand.

Each figure element can be added to skeleton layout through children of grid. The sample of coding is *skeletonLayout.Children.Add(figure element)*

After defining a skeleton layout, next step is laying it over color image.

V.3.2 Overlay Color Image

The skeleton layout consists of graphic drawing of body parts. Laying it over a color image can make human body skeleton captured by Kinect sensor transparently display on top of another graphic image. Kinect sensor can capture both color and skeleton images simultaneously, therefore overlaying skeleton layout on color image are unified.

The first step of overlaying process is creating a coordinating mapper and use it to map a point from skeleton space to color space. Kinect SDK provides coordinates that are represented in pixels translation functions for mapping points from one image frame to another. In our application, a skeleton image is required to map to color image, hence event *MapSkeletonPointToColorPoint* of *CoordinateMapper* is used. The mapping process should apply to each skeleton point.

The 2D coordinates that are represented in pixels of mapping point is multiplied by ratio of frame sizes of skeleton layout and color image frames. The actual coordinates of point are shown in Equation **??**.

$$actual\ point.X = actual\ point.X * (int)(skeleton\ layout\ width)/ \\ (color\ stream\ frame\ width) \tag{V.1}$$

$$actual\ point.Y = actual\ point.Y * (int)(skeleton\ layout\ height)/ \\ (color\ stream\ frame\ height) \tag{V.2}$$

The mapped points coordinates are used to draw joint line on graphic canvas that is laid on top of color image. Figure 19 shows an sample of overlaid image for bending posture.

In this chapter, we describe how the non-compliant activities are monitored and tracked. It includes overview of design architecture and detailed programming implementations.

```
<InvarianceRules>
  <ConfigurationGroup>
    <AlertLevel>2</AlertLevel>
    <AlertBody>Bend and twist.</AlertBody>
    <AlertVibration>1</AlertVibration>
   <Configuration>
      <Type>BoneOrientation</Type>
      <AlertLevel>1</AlertLevel>
      <AlertBody>Description</AlertBody>
      <AlertVibration>0</AlertVibration>
      <DownstreamJoint>HipCenter</DownstreamJoint>
      <UpstreamJoint>ShoulderCenter</UpstreamJoint>
      <FrontalAngle>0</FrontalAngle>
      <SagittalAngle>-1</SagittalAngle>
      <TransverseAngle>-1</TransverseAngle>
      <FrontalAngleTolerance>20</FrontalAngleTolerance>
      <SagittalAngleTolerance>-1</SagittalAngleTolerance>
      <TransverseAngleTolerance>-1</TransverseAngleTolerance>
    </Configuration>
    <Configuration>
      <Type>BoneOrientation</Type>
      <AlertLevel>1</AlertLevel>
      <AlertBody>Description</AlertBody>
      <AlertVibration>0</AlertVibration>
      <DownstreamJoint>HipLeft</DownstreamJoint>
      <UpstreamJoint>HipRight</UpstreamJoint>
      <FrontalAngle>0</FrontalAngle>
      <SagittalAngle>-1</SagittalAngle>
      <TransverseAngle>-1</TransverseAngle>
      <FrontalAngleTolerance>30</FrontalAngleTolerance>
      <SagittalAngleTolerance>-1</SagittalAngleTolerance>
      <TransverseAngleTolerance>-1</TransverseAngleTolerance>
    </Configuration>
  </ConfigurationGroup>
  <Configuration>
      <Type>BoneOrientation</Type>
      <AlertLevel>1</AlertLevel>
      <AlertBody>Bend.</AlertBody>
      <AlertVibration>1</AlertVibration>
      <DownstreamJoint>HipCenter</DownstreamJoint>
      <UpstreamJoint>ShoulderCenter</UpstreamJoint>
      <FrontalAngle>0</FrontalAngle>
      <SagittalAngle>-1</SagittalAngle>
      <TransverseAngle>-1</TransverseAngle>
      <FrontalAngleTolerance>20</FrontalAngleTolerance>
      <SagittalAngleTolerance>-1</SagittalAngleTolerance>
      <TransverseAngleTolerance>-1</TransverseAngleTolerance>
    </Configuration>
  </InvarianceRules>
```

Figure 17: Sample of Invariance Rule Defined in XML Format

```
float x1 = skeleton.Joints[JointType.ShoulderLeft].Position.X;
float y1 = skeleton.Joints[JointType.ShoulderLeft].Position.Y;
float z1 = skeleton.Joints[JointType.ShoulderLeft].Position.Z;
float x2 = skeleton.Joints[JointType.HipLeft].Position.X;
float y2 = skeleton.Joints[JointType.HipLeft].Position.Y;
float z2 = skeleton.Joints[JointType.HipLeft].Position.Z;
```

Figure 18: Retrieving Joint Point's 3D coordinates from Skeleton Frame

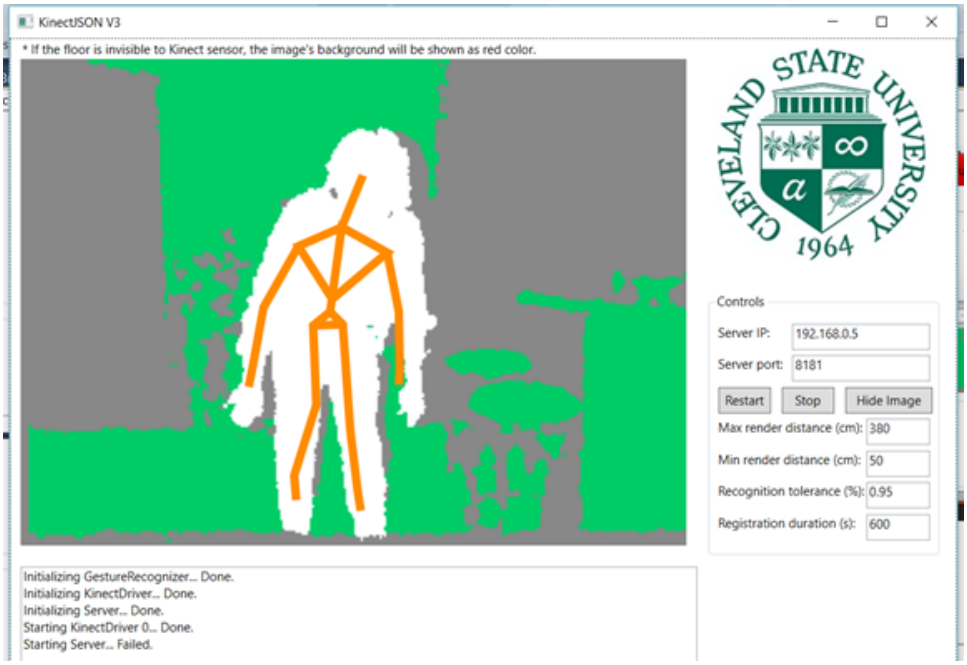

Figure 19: Laying Skeleton Image on Tracked Color Image Showing Body Bending Posture

CHAPTER VI

REAL-TIME FEEDBACK

It is very critical for the user to receive updates during exercise. The real time monitoring can also ensure the exercise performed is reliable and accurate. In our Integrated Human Activity Tracking and Monitoring System, we provide real-time feedback to user through a wearable device, particular a Pebble Watch.

The real-time feedback provides information regarding non-compliant human body movements detected by Kinect sensor. The feedback is interactive during body motion, and the quantities and duration of such activities are also reported back to user as well. Instance notification is generated and sent to the client end device through a smart-phone type of device, which will produce an action-aware event on the wearable device(e.g. Pebble watch) to alert the user.

The process of providing real-time feedback is relatively straightforward, which is similar to the handling of the registration request. It sends a response back to user about registration status upon user making a request. It also sends user notification message by checking if any non-compliant activity being detected. Or a null record is returned for indicating if no wrong activity has been detected.

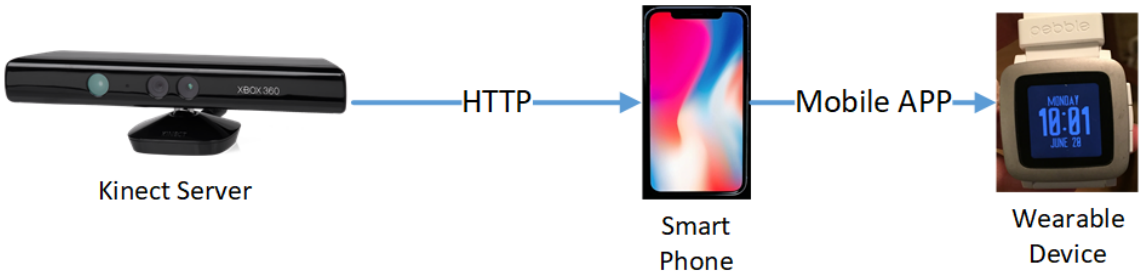

Figure 20: Communication Transport Model Used in Real-time Feedback

VI.1 Design Overview

The wearable device used in our system is Pebble Watch, which provides mobile application that allows exchanging messages with a mobile device. In our mobile application plays a role in bridging a way to connect Kinect server with wearable device thereby for communication between two devices.

The following Figure 20 shows the communication architecture of implementation for providing real-time feedback for user.

VI.2 Communication Between Kinect server and Mobile Application

The communication between Kinect server and mobile application is implemented through a HTTP transport model. The .Net framework that is used in our application provides a simple, programmatically controlled HTTP protocol listener for responding to HTTP requests. This listener is kept active for during application running.

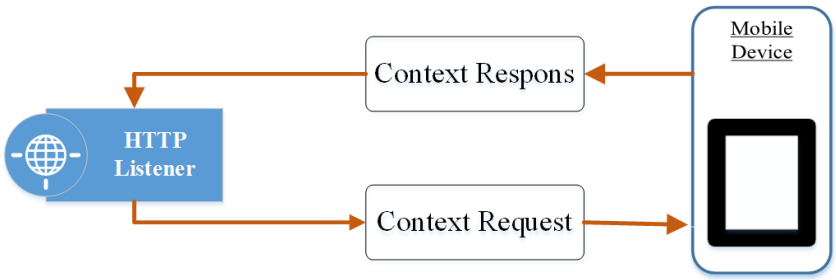

Figure 21: HTTP Listener Used for Communication Between Kinect Server and Mobile Device

VI.2.1 HTTP Listener

HTTP listener in .Net framework is used for building communication between Kinect server and mobile application. It is built on top of HTTP protocol that is in the kernel mode for handling all HTTP traffic for Windows operating system.

To use HTTP listener, a new instance of the class of HTTP protocol listener should be created with supplying a local IPv4 address and a port number. In order to create message exchanges between client and server, an object context must be created through active listener's *GetContext* method.

The client request can be obtained by*request* class of listener context, thereby server response can be sent by *response* class of listener context. Figure 21 illustrates the architecture of HTTP listener.

VI.2.2 Receive Message from Client

The process of receiving message from a client (i.e. a smart phone application) is implemented by *request* class of HTTP listener context. After listener starts, it is waiting for any message sent from authorized connection. The message received by context request object is parsed in a specific format, which is *http://ip:port/?type=MESSAGETYPE&userID=USERID*. The definitions of request message are listed as:

- ip:port - ip address and port number used for transmitting data

- type - an integer number represents what kind of messages. Currently only two types of messages are accepted.
  - 100 - registration request sent by wearable device
  - 120 - data request sent by wearable device
- userID - user registered ID

VI.2.3 Send Message to Client

The process of sending message to a client (i.e. a smart phone application) is implemented by *response* class of HTTP listener context. An IO stream object is needed for sending server message to client. The sending message is configured to a JSON response message.

JSON is an interchangeable data communication format that is based on a subset of JavaScript. JSON can represent simple or complex structured data. It is strictly used only for transporting data, not for function or expression. A JSON message text can easily be converted into a JavaScript value, which makes it a very convenient format for use in HTTP protocol. In our application, the message sent to client consists of four elements, type, title, body and vibe. *type* is an integer number represents what kind of messages being sent to client. The following types are used in our application:

- type 101 and vibe is 2 - success to registered
- type 103 and vibe is 1 - recognition of registration gesture failed
- type 110 and vibe is 0 - notify user is out of view range of Kinect sensor
- type 110 and vibe is 2 - notify user is unregistered
- type 121 and vibe is n - notify user how many times (n) non-compliant activities have detected

The messages are sent out as output stream of response context. To prevent the active server keeping sending out message, output stream of response context must be flushed and closed each time. The steps of sending message to client through HTTP listener are:

- create an instance response object from HTTP listener context: *response = HTTP.listener.context*
- create an output stream instance: *output = response.OutputStream*
- write message to output stream: *output.Write(message, 0, message.Length)*
- flush output stream: *output.flush()*
- close output stream: *output.close()*

Above sample coding shows how to send a simple and basic message from Kinect sever to mobile application using HTTP protocol in our real-time feedback module.

CHAPTER VII

LOGGING ACTIVITIES

The Logging activity programming module is designed to capture user's activity at different time frame and save them to a local device file or on-line data storage. This feature allows users or administrators to monitor and evaluate what participants do. The logging data also provides some insight of what the users do and statistical information regarding activities. Meantime, this application captures a screen shot on which user's skeleton image is displayed when a non-compliant activity is detected. This feature provides a detailed information, typically for interest when diagnosing problems is required, as well as of future improvements for detection accuracy.

VII.1 Introduction

The activity logging is one of major components developed in our Privacy-Aware Human Activity Tracking System. It provides functionality to save user's activity's attributes local device or onto a cloud based database, as well as functionality to capture screen shot with user's skeleton image.

Analyzing data generated from worker's activity is one way to ensure tasks are performed compliantly with defined rules. The collected data can help administrator or users to know who,

when, and where such activities are engaged currently, previously or histrionically. The captured user's skeleton image shows actual body movement when non-compliant activity is detected. The graphical image can help user to identify where, how, and why the movement goes wrong and also can verify the accuracy of detection. Displaying data visually can make it easily understandable and simplifying the data features. Our logging module includes identifying when to collect data, where to store data, and what type of activities are required to log.

The Logging activity programming module includes two major components, 1) collecting user's activity data; 2) capturing real-time user's skeleton image when a non-compliant movement is detected. The activity data can be stored either on a local or remote data storage. The captured image are stored locally. This section focuses on logging activity data and capturing screen image, and data visualization system will be addressed in Chapter VIII.

VII.2 User's Job Activities Log

the logging activity programming module handles all implementation regarding job activities in an encapsulated programming object called *Log* class. It saves a series of data about a user's job activity, which includes structured data for a particular user regarding date, time and duration for sign-on session, as well as data, time and duration for a non-compliant activity. The information is stored in *Job Activity Log*. This *Log* class exclusively handles all attributes and methods regarding activity log. The design strategy is utilizing benefits of object-oriented programming structure. The OOP structure makes programming code much easier to maintain and modify.

VII.2.1 Members of Log Object

The *Log* class maintains a constructor that automatically loads log parameters from a configuration file, an enumerator for types of activities, and an inner class *UserLog* that is an instance of log job activity for a particular user. The *Log* class diagram in the Unified Modeling Language (UML) is shown in Figure 22

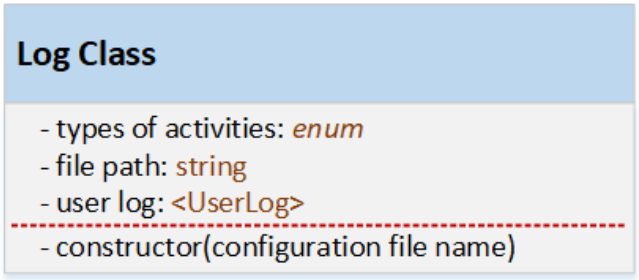

Figure 22: UML of Log Class

VII.2.1.1 Type of Activities

The type of activities that are tracked and saved are defined in an enumerator variable in log class. This includes eight elements. Each element defines a type of activity that user performs on job duty. Only defined type of activity is tracked and saved in log. Such activities are described as follows.

- Registered - user has registered with system
- ReRegistered - user has been registered before, now he or she registers again
- UnRegistered - user unregistered with system
- FoundUser - a previously registered user is just found in the view range of Kinect sensor
- LostUser - user walks away or is out of view range of Kinect sensor
- WrongActivityDetected - a non-compliant activity detected by Kinect sensor
- WrongActivityEnd - system detecting the ending of a non-compliant activity

VII.2.1.2 Retrieving Parameters from Configuration File

In order to provide a flexibility for selecting where to save the log file and name of file, our application use a configuration file for initial settings and the parameters used by log file. The configuration parameters are used to identify who is the user to track, where the log file is located,

and what the log file name is for a specific user. The job activity is associated with a specific user, therefore the user's activity log must have a user's identification.

The Kinect sensor provides a long integer *player id* that contains about 60 characters, which is a unique identifier representing tracked/lost person. This id is unique during a given time period for a person seen by the Kinect. Its value is assigned by Kinect hardware for matching and connecting with Kinect sensor data such as the player's location, joints, and image pixels.

Each user has a personal log file throughout the system, meaning each user has one log file. The log file location and name are maintained dynamically, meaning they are defined in a configuration file instead of hard-coding. Hence The administrator can change location log file or move old log file to set a new one.

The configuration parameters are set by program initially. In other words, if there isn't log file parameters found in configuration file, the system will add default ones automatically. If existing information is found, then parameters will retrieve and use for saving user's log data. Particularly, there are three elements required for configuration parameters: user id, log file path and log file name. The configuration parameters are in XML format and their labels are defined as following.

- User ID - a unique Kinect player ID, default is the current tracked player ID.
- Log File Path - log file path, default is "logs" folder in current working directory.
- Log File Name - log file name, default is "yyyymmdd" and a two digits sequence number with file extension of ".xml."

The sample of XML log configuration is shown in Figure 23

The loading function reads log parameters from XML configuration file and saves these parameters to class attributes. Shown in Figure 23, the information of user log configuration is ranged by XML name tag *¡UserLog¿*. The data specifically related to a user is defined by XML

Figure 23: Log Configuration

name tag *¡Use¿*. User id is under *¡ID¿*, file path is defined by *¡Path¿*, and file name is defined by *¡Filename¿*. User id, file path and file name are saved in user's *¡UserLog¿* object.

VII.2.1.3 User Log Object

The *Log* class had an inner class called *UserLog*, which is an object describing variables and methods required by saving a user's job activity log. The detail of this class will be discussed in the following Section VII.2.2

VII.2.2 UserLog Object

The *UserLog* is an object member of *Log* class. It is the centralized programming module to handle all functionalities regarding logging a user's job activities. It includes user's activity of type, data structures attributes of session and activity, as well as operational members for performing writing log to local device. The major operational methods are a constructor, and a control procedure to handle request for logging user's activity. The class diagram in the Unified Modeling Language (UML) is shown in Figure 34, which describes the structure of *UserLog* class by showing the attributes, operations and the relationships among class members.

VII.2.2.1 Data Members

The data members of user's log is categorized into two groups, session and activities. Therefore two data objects are structured to two entities: session and activity. Each describes what type activity the worker performs on duty, starting time, ending time, and duration of activity.

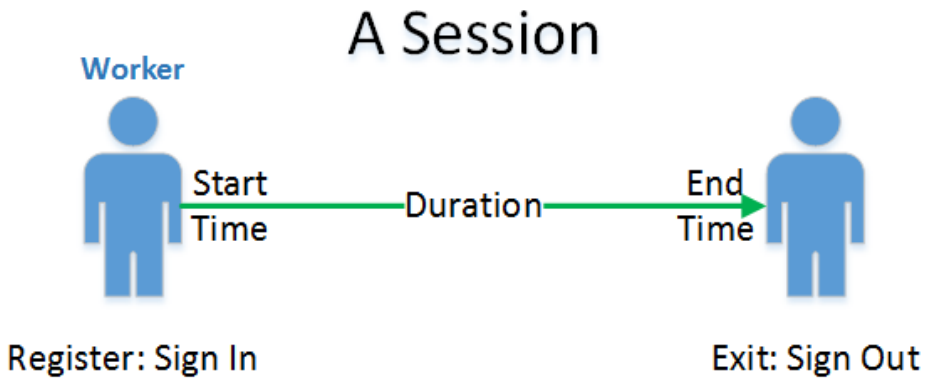

Figure 24: Characteristics of Session

- **Session** A session is defined as it starts when a worker is registering with system and it ends when a worker is walking off or signing off. One session is associated to a specific worker during a designed time frame. Each user can have many sessions during a certain time period. And during a session, there could be one or more activities detected during one session. The session log tracks worker id, the start time, and end time that is used to calculate session duration.

  * Session ID - a sequential integer used to identify a session.
  * Session Start Time - a date time stamp when the worker is registering with system.
  * Session End Time - a date time stamp when the worker is signing off with system. It is also used to calculate session duration.

  Figure 24 shows characteristics of session.

- **Activity** An activity is defined as it is a non-compliant one that are performed by a worker during a session. The normal or compliant activities are not recored in the logging process. The attributes of an activity log includes session id, activity description, activity starting time and activity ending time. The activity duration is computed by time elapsing between starting time and ending time.

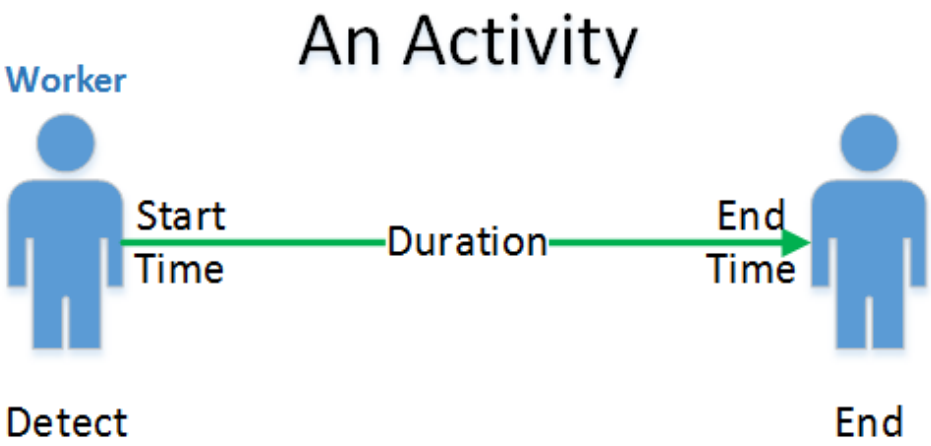

Figure 25: Characteristics of Activity

* Session ID - during which session an activity captured.
* Activity Description.
* Activity Start Time - a date time stamp when activity is detected.
* Activity End Time - a date time stamp when activity is ended.

Figure 25 shows characteristics of an activity.

VII.2.2.2 Operational Members

The *UserLog* class has only one operational member to handles all types of activities. This controller function is more than able to using the minimal input from different types of activities. Thus it can make code more maintainable and easier identifying the source of errors because the implementation is self-contained (*encapsulation*).

The controller function is a dispatcher used for determining which action needs to execute based on input parameters. To better illustrate this process, the following flowchart Figure 26 shows how it works with input parameters.

The input parameters to controller are a user id that is a standard player id generated by Kinect sensor, and an activity type. Based on the type of activity, controller dispatches to different

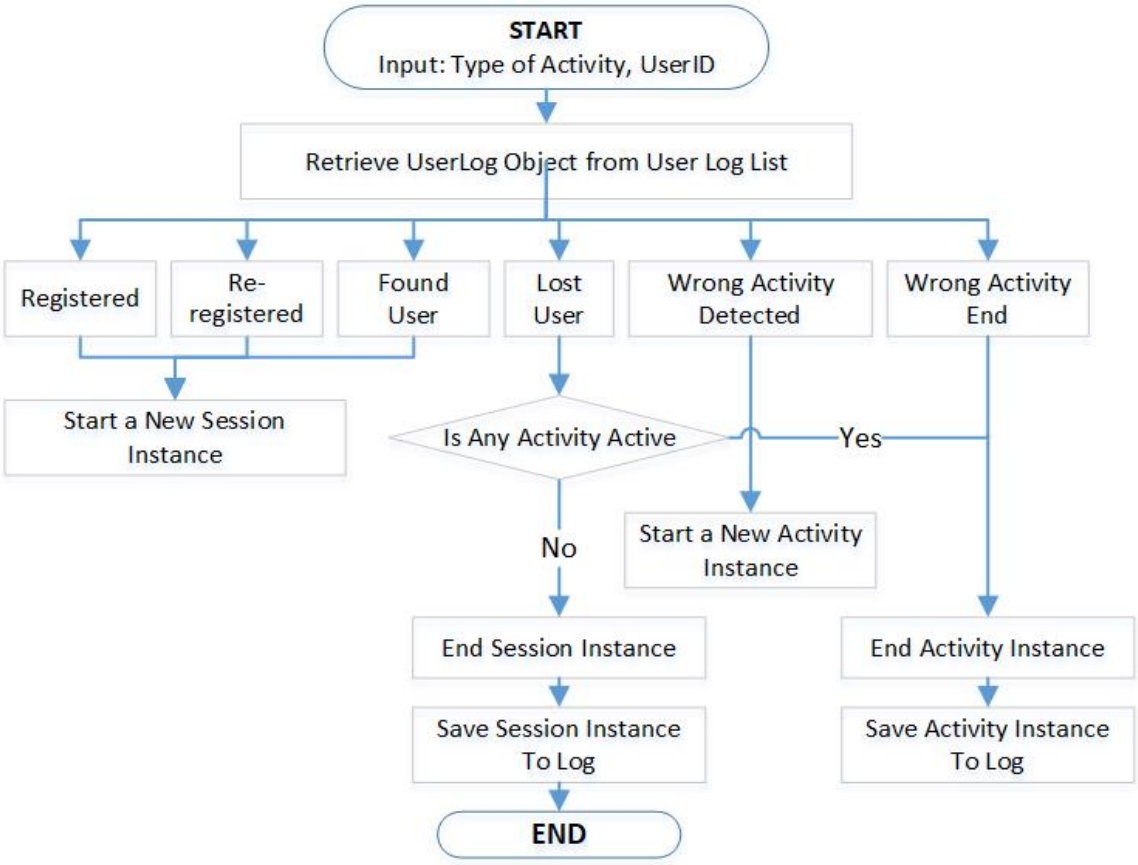

Figure 26: Implementation of Programming Controller to Handle User's Activity Log

actions. The basic actions include initiating a new session, closing/ending a session, initiating a new activity and closing/ending an activity, saving a session to device, and saving an activity to device.

The activities for initiating a new session in *UserLog* class are: 1) registered; 2) re-registered; 3) found-user. Basically, any sign-in activity needs a new session. A session is closed whenever the user is lost from Kinect sensor tracking. An activity object is initiated when a non-compliant activity is detected, and it is ended by a compliant one found.

The session and activity logs are flushed to device storages when instances are closed. We provide two types storages for saving activity logs, one is local device, and the other is cloud-based database, which will be addressed in detail in next section.

VII.2.3 Storage of Activities Log

We provide two types of data storages for saving user's activity logs, one is on local device, and the other is on cloud based database. The benefit of using local data storage is simple, easier maintain and most important fact is ease of the security concern regarding data breaches. On the other hand, cloud based storage allows real time data sharing, viewing, monitoring, and as well as being centralized data visualization.

VII.2.3.1 Local Data Storage

The information of user's job activity can be saved in on a local device. As described in above Section VII.2.1.2, the log file name is defined in log configuration file, which is a concatenated string of values from name tags of *path* and *filename*. With thus flexibility of having file path and name defined dynamically, it is much easier to move file location or use other file name.

In addition to cloud-based database, the logging information is saved to a locally resided XML file. The attributes of XML file are in hierarchy data model. The local device log file consists a standard data record. A particular log file includes the following elements:

- User ID - user identification who performs activities in this log.
- Date Value - date range tag, which can include multiple sessions.
- Session - a session has start time and end time can include multiple activities.
- Activity - an activity tag includes starting time, type of activity, and duration.

Figure 27 illustrates a sample of local log file. In this sample, it shows user's activities during two days. Each data record keeps a starting time, type of activity and duration of that activity.

VII.2.3.2 Cloud Based Data Storage

The logs can be saved both the locally in an XML log file and remotely to the a cloud-based database in Amazon Web Services. In this section we focus on saving log of user's activities

```
<UserID>f81924d6ed39525a0fd3f1b7e47b61a01e28b1f9af7ea5415ca1c8ec7e59b6f4</UserID>
<Date Value=" 12/18/2017">
  <Session Start=" 12/18/2017 23:41:47">
    <Activity Start=" 12/18/2017 23:41:47" Activity="Registered"/>
    <Activity Start=" 12/18/2017 23:41:52" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/18/2017 23:41:55" Duration="2.760524" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/18/2017 23:41:56" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/18/2017 23:41:59" Duration="2.8056363" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/18/2017 23:42:01" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/18/2017 23:42:03" Duration="2.5488687" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/18/2017 23:42:06" Duration="2.5488687" Activity="Lost User"/>
  </Session>
</Date>
<Date Value=" 12/22/2017">
  <Session Start=" 12/22/2017 14:54:56">
    <Activity Start=" 12/22/2017 14:54:56" Activity="Registered"/>
    <Activity Start=" 12/22/2017 14:55:01" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/22/2017 14:55:05" Duration="3.6224342" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/22/2017 14:55:07" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/22/2017 14:55:11" Duration="3.7837682" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/22/2017 14:55:14" Activity="Wrong Activity Detected"/>
    <Activity Start=" 12/22/2017 14:55:18" Duration="3.3414465" Activity="Wrong Activity Ends"/>
    <Activity Start=" 12/22/2017 14:55:21" Activity="Lost User"/>
  </Session>
```

Figure 27: A sample of local log data record

remotely in a cloud based data storage. The advantage of logging data to cloud-based database is that all information from clients (i.e. different Kinect servers) can be kept in a centralized place. Utilizing centralized cloud-base database, the administrator of organization can review, analyze and collect data from anywhere at anytime. Chapter VIII details the design of our data visualization system.

We utilize cloud based relational database to save data record, particularly open source mySQL database provided by Amazon Web Services (AWS) is used. Amazon Relational Database Service (Amazon RDS) is easy to set up, operate, and scalable in the cloud. It provides low cost and scalable capacity as far as for maintenance for administration tasks. It gives applications the fast performance, high availability, security and compatibility as they need.

The data log is structured in hierarchical relational data model in which information is organized into a tree-like structure. The top of tree is user who has children of sessions, each session has children of activities. Figure 28 shows data model structure. A data log composes two types of information: (1) session that includes the time when the user registered,re-register, or found again, and when the user walked away or disappear; (2) non-compliant activity that includes the time it is

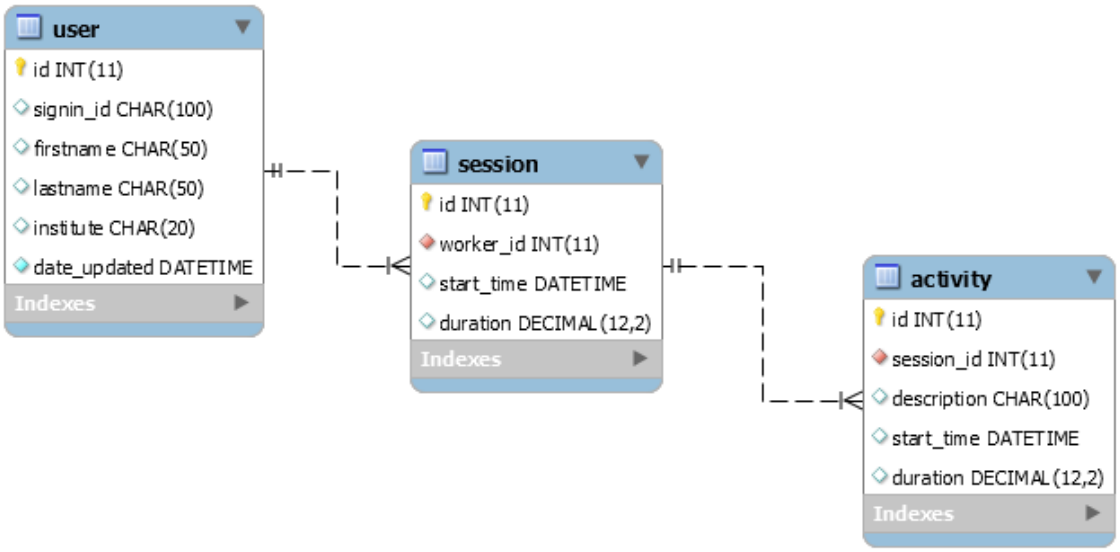

Figure 28: Cloud-based database schema diagram

detected, type of activity and activity duration.

The schema of log data consists entity of user, session and activities. The each data entity (i.e. table) includes a foreign key field that links to ancestor, such as session table has identity field for primary key field in user table (one user may have many sessions), user table has identity field for primary key field in session table (one session may have many activities). A database programming interface is created to handle all data retrieval and update between Kinect server and cloud database. It is designed to be an inheritance hierarchy in object model so each database table has a corresponding programming object that matches each data attribute with programming variable. The data access including insert, update, and retrieval. These tasks are performed by stored procedure with proper input and output parameters. The benefits of using stored procedure are: (1) reduce the network traffic between client and server; (2) stronger security that multiple users and client programs can operate on underlying database objects without direct permissions to those underlying objects; (3) easier maintenance and reuse of code.

VII.2.3.3 Dual Data Storage

We provide the dual-modality in logging user's activities to ensure high reliability of our framework. Even if the Amazon Web Services is temporarily unavailable, no data is lost because a local file can be a backup. We develop a function that can transfer logging information from XML file to cloud-based database.

VII.3 Tracking Image of Real-time Movement

In our application, there is an important feature that allows user to track image of real-time non-compliant activity. If user selects tracking, once a wrong activity is detected, an image of that movement is automatically captured and stored. The image is an instant copy of pixel screen on where the body skeleton is displayed.

Our application provides an option for user to select if image of incorrect movement needs to be saved or not. The implementation of saving image consists of providing a check box that users can select to indicate they would like to save an image of screen shot when an incorrect movement is detected, and a function that can capture screen pixels made from skeleton image.

The flow chart of this process is shown in Figure 29

VII.3.1 User Selection

On the main screen of application, there is a check box for option of taking screen shot if incorrect movement is detected. If user marks this option, a status variable is used to save the selection. During the process of validating user's movement, capture screen function is called once an incorrect movement is detected and saving screen shot option is selected. Figure 30 shows user selection.

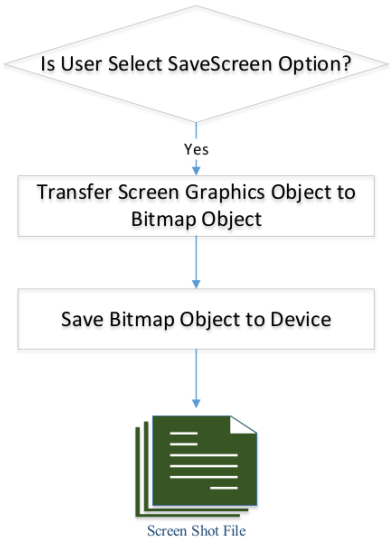

Figure 29: Process to Save Screen Shot

VII.3.2 Capturing Active Windows Image to a Bitmap Object

The purpose of saving screen shot is for administrator to track, review and analyze the user's movement when system detects it is incorrect. The user's skeleton image is displayed after user's is registered. Therefore tasking of saving skeleton image becomes how to save screen image to a file.

Miscrosoft Kinect SDK provides skeleton image in format of pixels. But the graphic file format is device-independent bitmap (DIB). The challenge is how to capture the active window in pixels and save it in BMP format.

The technique we use in this application is to transfer active window pixels in bitmap scale first and then save bitmap scale to device. Active screen window where a user's skeleton image is displayed, is defined by window's work area width and height, which are represented by System.Windows parameters. The image pixel is an integer of unit of screen color in a computer image.

The process to copy active screen image to a graphic object consists following steps.

- Retrieving number of pixels from window working area of width and height

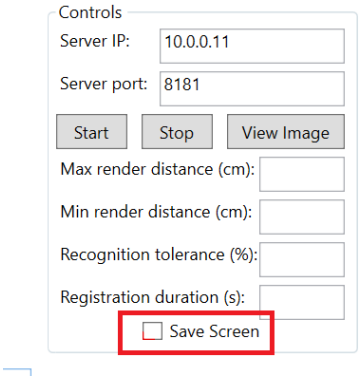

Figure 30: User Option for Capturing Image of Incorrect Movement

The number of pixels of active screen window can be obtained by System.Windows parameters. In .Net framework, width and height pixels are set by an integer value of Systems.Windows.SystemParameters.WorkArea.Width and and integer value of Systems.Windows.SystemPa respectively, shown in Equation VII.1.

$$\begin{aligned} \textit{imageWidthPixel} &= (Integer)(SystemParameters.WorkArea.Width) \\ \textit{imageHeightPixel} &= (Integer)(SystemParameters.WorkArea.Height) \end{aligned} \tag{VII.1}$$

- Creating an image bitmap object of active screen's number of width pixels and number of height pixels in RBG color format

  An bitmap image object can be initiated by Bitmap component in drawing class.

  bitmap image object = new System.Drawing.Bitmap(image width pixels, image width pixels, color arbg)

- Creating a graphic object for image bitmap object by inheriting system drawing image class

  A graphic object is used for drawing surface. For our case, a image of active windows can be copied to this graphical object. In order to transfer graphical object to bitmap image object, the graphic object must inherit bitmap object. Such connection can be initiated as following.

  graphic screen shot = System.Drawing.Graphics.FromImage(bitmap object)

- Copying active window image from screen to graphic screen shot object The active window image can be copied to a graphic image object by defining window's origin point and size of windows. Microsoft .Net framework provides a function for copying active window image from the upper left corner to the right bottom corner. In our application, the origin point in (0,0) and size of capture area is defined by window's width pixels and height pixels. It is represented by: *graphic screen shot.CopyFromScreen(Point(0, 0), Size(imageWidthInPixel, imageHeightInPixel))*;

VII.3.3 Saving Bitmap Object

The captured graphical image object can be saved to device by implementing a bitmap image object implementation member called *Save*. The *Save* function requires two parameters, save to file name and file type. The file name consists date, time, and activity number that matches the one is saved in log file. The file type is JEPG graphic file. the bitmap image object is defined in above section. Its implementation is carried out by the following code.

bitmap image object.Save(file name, ImageFormat.Png)

Being able to track and save a screen shot when incorrect movement is detected can help administrative user to identify movements and verify the detection accuracy. The screen shot files can be tracked with logging information saved in log file with matching activity ID. We believe this feature has great benefits in term of application usability, and future detection accuracy improvements.

CHAPTER VIII

DATA VISUALIZATION SYSTEM

The real-time data visualization of user's job activity is one of major components developed in our Privacy-Aware Human Activity Tracking System. It provides not only functionality to save user's activities information on local device or on a cloud based database, capturing screen shot with user's skeleton image for non-compliant activity, but also a graphical interactive user interface to allow health care workers and administrators to review real-time and historical statistic information of activities.

Reviewing and analyzing users' activities are a way to make sure they perform tasks correctly with requirements. The visualization data system provides the statistical data in a graphic model for users. Thus helps users to visually know who, when, and where such activities are engaged currently, previously or histrionically. The captured user's skeleton image also shows actual body movement when non-compliant activity is detected. The graphical image can help users to identify where the movement goes wrong and also can verify the accuracy of detection. Displaying data using graphics can make it easy for users to understand and can simplify the data features.

In this chapter we focus on addressing how data analysis interface process works. we starts with discussion of the importance of good visualization, then illustrating the design architecture of application interface, and finally with a sample of real-time data.

VIII.1 The Importance of Data Visualization

Graphically, artistically, even visually can help us see data in an effective ways. It is said the simpler graphs might be more effective [160]. The key is to present data in a way that is visually allowing the data to speak for itself.

One of the key benefits of our data visualization is that it allows user to access large amount of data with integrated view in real time, as well as the simple and powerful data graphs. Our visualization data system transforms the numerous of users activity data into meaningful information. It provides data displays of quantities by using grouped points, which can visualize relationships and patterns between measured data.

Another benefit of data visualization is to identify on emerging trends faster. Using visual graphic to show data trends can be much more effective than a long written sentence for explaining what data represents. Many studies report that data visualization tools are effective in portraying complex or vast quantities of data [160] [161] [162]. Our visualization data system illustrates how many times at which time period users' non-compliant activities are detected. Thus data visualization results can provide a very important trend of users' non-compliant activities during their duty.

Therefore, the design strategy of visualization interface program extensively reflects the importance of data visualization.

VIII.2 Overview of System Design

The data visualization program is a stand alone, integrated application that connects to a mySQL database server. It also includes a Graphical User Interface (GUI) for users to interact with application. The data of users' activities is stored in mySQL database, and user interface allows user to select statistical aggregation data.

The mySQL is a popular Open Source relational database management system widely used

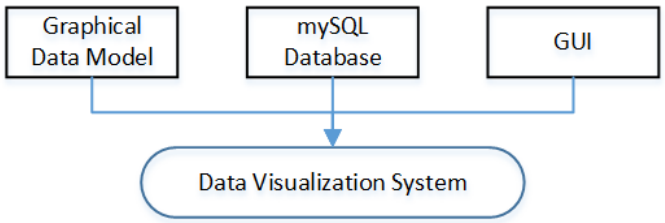

Figure 31: Major Components of Data Visualization System

due to its flexibility, reliability, speed, which equips with Structured Query Language (SQL) for accessing and processing data. To feed data for user interface component, many stored procedures are used for retrieving different aspects of data.

The graphical representation of data is implemented by using *OxyPlot* that is a cross-platform plotting library for .NET framework. It provides graphical symbols such as lines, curves, bars, pie slices, dots, etc for illustrating numerical data in the form of a qualitative structure and important information.

Another important component of our data visualization program is an interactive form that allows user to select statistical data parameters, such as user, date range, hour range, etc. The selected parameters then are sent to database to perform mySQL aggregate functions for retrieving statistical data.

Each programming component described herein is significantly importantly in implementation of data visualization system. The Figure 31 shows system architecture design consisting components: 1) an graphical user interface; 2) a cloud based database used as data source; 3) a graphic library that provides model for visualizing qualitative numeric data.

VIII.3 Major Programming Components

Three components are included in data visualization system: 1) an graphical user interface; 2) a cloud based database used as data source; 3) a graphic library that provides model for visualizing qualitative numeric data. Each of them plays very important tasks in storing, retrieving and

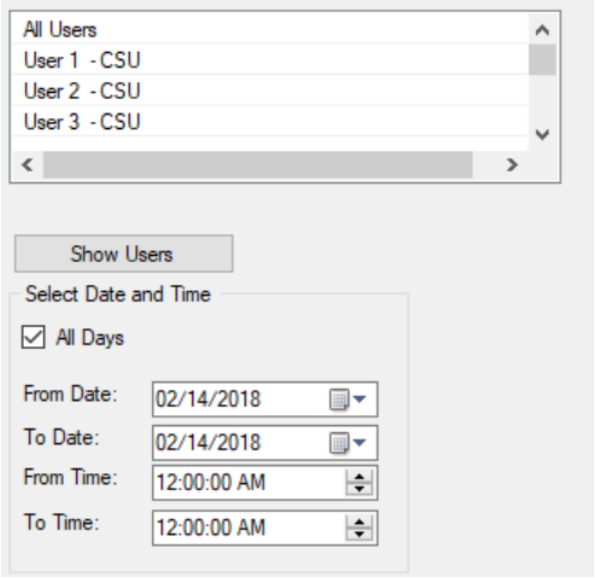

Figure 32: Major Components of Data Visualization System

illustrating numerical statistical users job activities data.

VIII.3.1 User Interface (GUI)

The Graphical User Interface (GUI) includes graphical controls, which enable user using a mouse or keyboard to make selections of certain parameters. A well-designed user interface should provide a ”user-friendly” experience interacting with program in a natural and intuitive matter.

The GUI plays two major roles in our data visualization system. First it provides selections for parameters used to retrieve statistical data from database. Secondly it provides three different type of views through button actions.

VIII.3.1.1 Data Parameters Selection

The GUI used in our data visualization system includes a list box for selecting a particular user or all users. and a date time selections for all days or a particular date or/and time. Figure 32 shows the screen layout for GUI selection elements.

The user interactive control elements on main screen of data visualization system are:

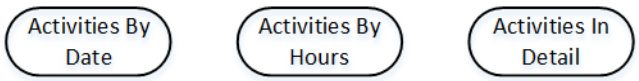

Figure 33: Three Types of Data Views

- user selection list: this drop-down list allows user to select whose activities data will display. The default value is selecting all activities.
- date range: a date picker allows users to select date range, or a specific date when activities are detected. The default value is today's date.
- time range: a date picker allows users to select time range, or a specific time when activities are detected. The default value is 12:00am.

Using standard date time pickers can normalize user's entry to a unified string, reduces programming difficulty on data entry variations. There is also an option for all activities without date time range. The selections of userID, date, and time range are parameters used by mySQL stored procedures, which will be described in detail in Section VIII.3.2.

VIII.3.1.2 Types of Data Views

Our data visualization program utilizes lists and graphs as a simple way to visualize job activities data, which are very useful for illustrating trends over time, or foreseeing future time frame when likely non-compliant activities occur.

Three types of data views are available on GUI, which are showing detail data, statistical abstract of activities by dates, and statistical abstract of activities by hours. The user interactive selections for such action types are presented using three buttons that are illustrated in Figure 33

VIII.3.2 Database Schema Design

We choose mySQL database as back-end data storage for data visualization system due to its great features of flexibility, reliability, speed, and functionality using Structured Query Language

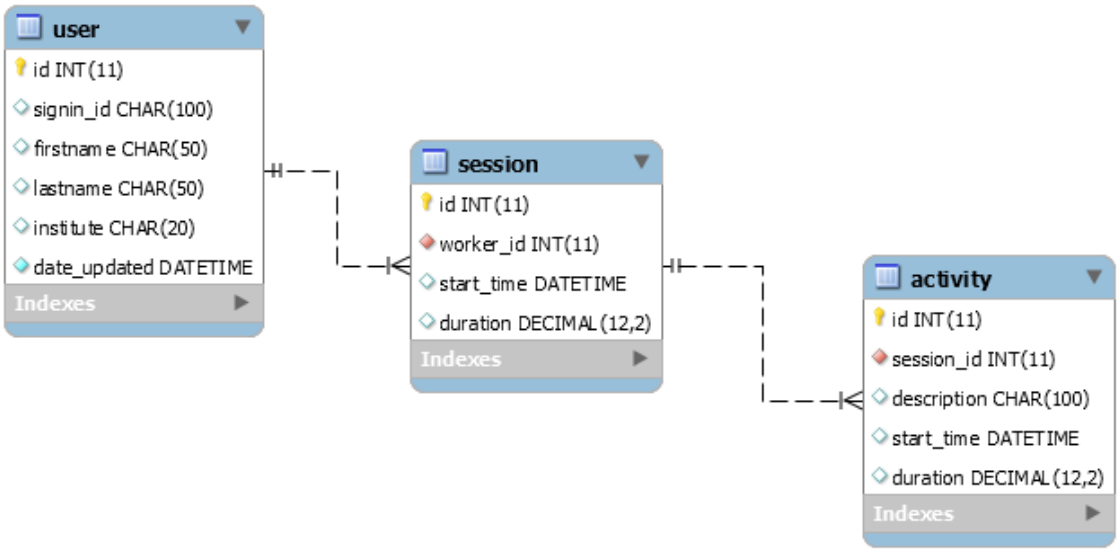

Figure 34: Entity-Relationship(ER) Diagram of Data Visualization Database Schema

(SQL) for accessing and processing data. The primary goal of database schema design is to meet data requirements and to provide sufficient functions that can retrieve aggregation data with input parameters. In addition, table entities lead to fully normalized as well.

VIII.3.2.1 Table Schema

The database schema for data visualization system should provide data entities for worker, session, and activity and the relationships between them. Therefore three data entities are employed in database for that purpose. The result of the logical design schema is shown in the Entity-Relationship (ER) diagram, Figure 34. As shown, relationships in all three data entities are one to many.

At top of the hierarchy of systematic data organization is worker table. Each worker can have many sessions during any date time period, and each session can have multiple activities performed at different time. Figure 35 illustrates data hierarchy and primary properties of tables are listed as following.

- **worker table:**

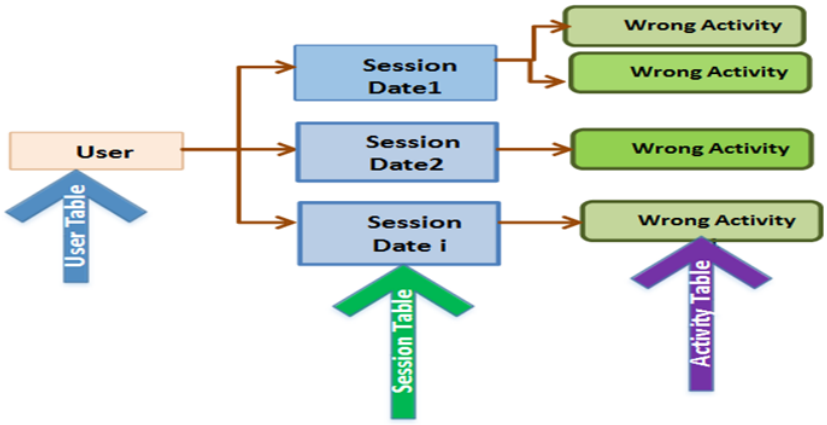

Figure 35: Data Hierarchy of Data Visualization Database Schema

- *id* - Primary Key, is used to link to session table
- *worker sign-in id* - long integer number generated by Kinect sensor
- *first name*
- *last name*
- *name of work place*

- **session table:**
  - *id* - Primary Key (session id), is used to link to activity table
  - *worker id* - Foreign Key linked with id field in worker table
  - *activity description*
  - *session start time*
  - *session duration*

- **activity table:**
  - *id* - Primary Key (activity id)
  - *session id* - Foreign Key linked with id field in session table
  - *activity description*

* *activity start time*

* *activity duration*

VIII.3.2.2 Stored Procedures

data visualization system allows user to select specific parameters, such as a worker or all workers, date range, time range, or all date for reviewing statistical information. As described in Section VIII.3.1, parameters are received from user interface. Thus these parameters can passed to stored procedures in database and retrieve desirable data.

A Stored procedure is a group of compiled statements written by Structured Query Language(SQL). It is encapsulated in a user-defined entity and stored in executable form in a database. The benefits of using stored procedures are improved performance, stronger security access and centralized computing resources for ease of use and maintenance.

The executable code can be automatically cached and accessed by authorized users, and most importantly it can be executed with a single call. This can significantly reduce network traffic between the database server and client program because only a small amount of coding (i.e. a single call) is sent across the network. Without the code encapsulation in a procedure, every individual line of code would have to go through the network, which generates more network traffic and may cause bottleneck performance or congestion. Such process minimizes the use of networks, reduces network traffic, and improves round-trip response time between server and client.

The stored procedure provides one operation on underlying database objects for multiple users or clients. The permission to access stored procedure can grant to users who might not have privilege of direct accessing of database tables, views or other objects. The controls of what processes are allowed are performed in a scalable matter, thus reduces of requirements to grant permission for each object and also manages the security requests in more efficient way to track, audit, respond, and report.

Programmer can generally find out using stored procedures result in much easier coding

maintenance because it is simpler and faster modifying a stored procedure than changing a hard-coded SQL statement inside client program. The stored procedures provide a level of abstraction from the underlying database schema. Therefore the client is isolated from the underlying data objects.

In our data visualization program, there are several stored procedures used for retrieving data from tables. All stored procedures have input parameters of worker id, from date, to date, from time, to time, and a group by key field name. If to date or to time is missing or is null, the date range is from date to the last date defined in table, this rule applies to time as well. The definitions of stored procedures are described as following.

- ***Get Activities By Date:*** retrieve activity information for a specific worker or all workers during a given date time range with aggregation of duration by each date.
- ***Get Activities By Hour:*** retrieve activity information for a specific worker or all workers during a given date time range with aggregation of duration by each hour.
- ***Get Activities Detail:*** retrieve activity information for a specific worker or all workers during a given date time range without aggregation.

Using stored procedures in mySQL database we take advantages them, including increased performance, higher security, ease of use, and better scalability. To retrieve data with selected parameters from database, we utilize several stored procedures to retrieve data to GUI program for illustration.

VIII.3.3 Graphical Data Model

Data Visualization program allows user to select a specific worker or all workers, date range and time range for reviewing activity statistical information. Thus information is represented in a graph object, including number of wrong activities found, average count, average duration. We employ the *OxyPlot* library that is compatible with .Net framework for this purpose.

OxyPlot is a light weight open source plotting library. It possesses features of easy to use, open for extensions and high performance. The package provides many plotting models, such as linear axes,logarithmic axes, line series, scatter series, area series, bar and column series, heat maps, etc. To implement plotting, *OxyPlot* must be downloaded and imported into framework.

VIII.3.3.1 Creating a Plotting Object

The basic requirements of using *OxyPlot* object to plot a graph using specific data are first creating a plot object, and then binding a data series to plot object. A plot object is an instance of PlotModel from *OxyPlot* library. Data series is a list of *DataPoint* type objects. *DataPoint* type object consists of a pair of X and Y point. The steps of creating plotting graph are listed as following:

- ***Initialing a plot model*:** PlotModel myModel = new PlotModel
- ***Creating a data series*:** LineSeries dateSeries = new LineSeries content...
- ***Adding data to data series*:** dateSeries.Points.Add(new DataPoint(x,y))
- ***Binding data series to plot model*:** myModel.Series.Add(dateSeries)

Above describing is the basic requirements for using *OxyPlot* object to plot a graph. Furthermore plot model has many properties that can add more features to graph.

VIII.3.3.2 Properties of Plotting Object

OxyPlot library package provides different types of axes and series. Particularly, in our application we focus on line series. The following features are used and implemented in data visualization program.

- ***Linear Axes*:** user can define axes position, title, title style, label, label style, label's interval, display angle, etc. It is normally used for y-axis

- ***Category Axis*:** user can define label's tinker, text notation, text notation style, display angle, axis position, etc. It can display multiple data series, therefore normally it is used for display x-axis.

Based on experience of using *OxyPlot* library, we feel it is a simple, easy of use, and friendly implemented graphic package.

In this chapter, we describe a standard alone application that allows for both instant administrative viewing and accessing extensive current and histories of statistic information of activities.

CHAPTER IX

CONCLUSION AND FUTURE WORK

In this dissertation research, we develop an integrated application that uses Kinect sensor, a motion sensing input device, for tracking non-compliant activity postures of consented health-care workers to increase the workers' compliance to best practices, individualizing gestures for each user to register by using machine learning algorithm - adaptive boost, monitoring rehabilitation exercise using kinematic modeling with fuzzy inference, and rule-based real-time feedback system for rehabilitation exercises. The research work also includes developing GUI and database system to log human activities detected by Kinect sensor and data visualization system used by administrator.

Motivated by advanced features and image data provided by Kinect sensor, gesture recognition technology, we believe Kinect that is a low cost game-based depth sensors have the potential to revolutionize real-time human body movement analysis research, benefit to various applications for computer vision society.

Summarizing the contributions of computer application we developed are the following:

- an integrated software system
- tracking and monitoring user's real-time activities

- protecting user's privacy
- providing real-time feedback
- collecting real-time activity data
- data visualization system

Future development Issues and Concerns: As we recognized earlier, communication between smart-phone and wearable device is via a blue-tooth connection, and its setup is a manually process. It would be more convenient for users to have an data. We also recognize that there is need for one smart-phone with for multiple wearable devices instead of currently one smart-phone with just one wearable device allowed.

In the current development of real-time feedback methodology, the number of types of messages are very limited, we hope to have more message code feedback to user, including why the gesture recognition fails, what type of non-compliant activity detects in future.

BIBLIOGRAPHY

[1] M. Hughes. Primesense ps1080 microcontroller. [Online]. Available: https://www.allaboutcircuits.com/news/teardown-tuesday-occipital-3d-structure-sensor

[2] Microsoft kinect programming guide, data strams. [Online]. Available: https://msdn.microsoft.com/en-us/library/hh973078.aspx

[3] J. M. Ortman, V. A. Velkoff, and H. Hogan, "An aging nation: The older population in the united states, population estimates and projections," 2014. [Online]. Available: http://www.census.gov/prod/2014pubs/p25-1140.pdf

[4] K. Mullen, M. Gillen, S. Kools, and P. Blanc, "Hospital nurses working wounded: Motivations and obstacles to return to work as experienced by nurses with injuries," *Work*, vol. 50, no. 2, pp. 295–304, 2015.

[5] A. E. Gomaa, L. C. Tapp, S. E. Luckhaupt, K. Vanoli, R. F. Sarmiento, W. M. Raudabaugh, S. Nowlin, and S. M. Sprigg, "Occupational traumatic injuries among workers in health care facilitiesunited states, 2012–2014," *Health Care*, vol. 2012, 2014.

[6] I. Aslam, S. A. Davis, S. R. Feldman, and W. E. Martin, "A review of patient lifting interventions to reduce health care worker injuries," *Workplace Health and Safety*, vol. 63,

no. 6, pp. 267–275, 2015. [Online]. Available: http://whs.sagepub.com/content/63/6/267.full.pdf+html

[7] C. Chen, G. Li, P. Ngo, and S. C., "Motion sensing technology," California Institute of Technology, Pasadena, 2011.

[8] R. Lun and W. Zhao, "A survey of applications and human motion recognition with microsoft kinect," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 05, p. 1555008, 2015. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218001415550083

[9] W. Zhao, R. Lun, C. Gordon, A. B. M. Fofana, D. D. Espy, M. A. Reinthal, B. Ekelman, G. D. Goodman, J. E. Niederriter, and X. Luo, "A human-centered activity tracking system: Toward a healthier workplace," in *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, June 2017, pp. 1784–1791.

[10] R. Lun and W. Zhao, "Kinect applications in healthcare," pp. 5876–5885.

[11] R. Lun, C. Gordon, and W. Zhao, "Tracking the activities of daily lives: An integrated approach," in *Future Technologies Conference (FTC)*, January 2017.

[12] W. Zhao, R. Lun, C. Gordon, A.-B. M. Fofana, D. D. Espy, A. Reinthal, B. Ekelman, G. D. Goodman, J. E. Niederriter, C. Luo, and X. Luo, "Liftingdoneright: A privacy-aware human motion tracking system for healthcare professionals," *International Journal of Handheld Computing Research (IJHCR)*, vol. 7, no. 3, pp. 1–15, March 2016.

[13] R. Lun, C. Gordon, and W. Zhao, "The design and implementation of a kinect-based framework for selective human activity tracking," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, October 2016, pp. 2890–2895.

[14] R. Lun and W. Zhao, "A survey of using microsoft kinect in healthcare," *Encyclopedia of Information Science and Technology, Third Edition. IGI Global*, vol. 3279-3287, 2015.

[15] W. Zhao, R. Lun, D. D. Espy, and M. A. Reinthal, "Rule based realtime motion assessment for rehabilitation exercises," in *Computational Intelligence in Healthcare and e-health (CICARE), 2014 IEEE Symposium on*. IEEE, December 2014, pp. 133–140.

[16] W. Zhao, R. Lun, D. D. Espy, and M. Ann Reinthal, "Realtime motion assessment for rehabilitation exercises: Integration of kinematic modeling with fuzzy inference," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 4, no. 4, pp. 267–285, 2014.

[17] W. Zhao, H. Feng, R. Lun, D. D. Espy, and M. A. Reinthal, "A kinect-based rehabilitation exercise monitoring and guidance system," in *2014 IEEE 5th International Conference on Software Engineering and Service Science*, June 2014.

[18] "Xbox 360 + kinect," 2012. [Online]. Available: http://www.xbox.com/en-us/kinect

[19] "Kinect hardware," 2014. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect/hardware

[20] J. Webb and J. Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.

[21] "Kinect hacking 101: Hack a powershot a540 for infrared sensitivity," http://www.danreetz.com/blog/2010/11/17/kinect-hacking-101-hack-a-powershot-a540-for-infrared-sensitivity/, 2011.

[22] Microsoft, "Microsoft releases kinect sdk 2.0." [Online]. Available: http://blogs.microsoft.com/blog/2014/10/22/microsoft-releases-kinect-sdk-2-0-new-adapter-kit/#sm.00001pqagh1726dohwjollswk3b06

[23] N. Villaroman, D. Rowe, and B. Swan, "Teaching natural user interaction using openni and the microsoft kinect sensor," in *Proceedings of the 2011 conference on Information technology education*, ser. SIGITE '11. New York, NY, USA: ACM, 2011, pp. 227–232. [Online]. Available: http://doi.acm.org/10.1145/2047594.2047654

[24] F. Farhadi-Niaki, R. GhasemAghaei, and A. Arya, "Empirical study of a vision-based depth-sensitive human-computer interaction system," in *Proceedings of the 10th asia pacific conference on Computer human interaction*, ser. APCHI '12. New York, NY, USA: ACM, 2012, pp. 101–108. [Online]. Available: http://doi.acm.org/10.1145/2350046.2350070

[25] M. Raj, S. H. Creem-Regehr, K. M. Rand, J. K. Stefanucci, and W. B. Thompson, "Kinect based 3d object manipulation on a desktop display," in *Proceedings of the ACM Symposium on Applied Perception*, ser. SAP '12. New York, NY, USA: ACM, 2012, pp. 99–102. [Online]. Available: http://doi.acm.org/10.1145/2338676.2338697

[26] R. Francese, I. Passero, and G. Tortora, "Wiimote and kinect: gestural user interfaces add a natural third dimension to hci," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12. New York, NY, USA: ACM, 2012, pp. 116–123. [Online]. Available: http://doi.acm.org/10.1145/2254556.2254580

[27] S. Chen, Y. Maeda, and Y. Takahashi, "Music conductor gesture recognized interactive music generation system," in *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, 2012, pp. 840–845.

[28] G. Freitag, M. Tränkner, and M. Wacker, "Enhanced feed-forward for a user aware multi-touch device," in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, ser. NordiCHI '12. New York, NY, USA: ACM, 2012, pp. 578–586. [Online]. Available: http://doi.acm.org/10.1145/2399016.2399104

[29] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris, "Code space: touch + air gesture hybrid interactions for supporting developer meetings," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser.

ITS '11. New York, NY, USA: ACM, 2011, pp. 212–221. [Online]. Available: http://doi.acm.org/10.1145/2076354.2076393

[30] K. Aitpayev and J. Gaber, "Collision avatar (ca): Adding collision objects for human body in augmented reality using kinect," in *Application of Information and Communication Technologies (AICT), 2012 6th International Conference on*, 2012, pp. 1–4.

[31] X. Tong, P. Xu, and X. Yan, "Research on skeleton animation motion data based on kinect," in *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, vol. 2, 2012, pp. 347–350.

[32] T. Franke, S. Kahn, M. Olbrich, and Y. Jung, "Enhancing realism of mixed reality applications through real-time depth-imaging devices in x3d," in *Proceedings of the 16th International Conference on 3D Web Technology*, ser. Web3D '11. New York, NY, USA: ACM, 2011, pp. 71–79. [Online]. Available: http://doi.acm.org/10.1145/2010425.2010439

[33] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 4, pp. 643–650, 2012.

[34] I. Nakachi, Y. Takeuchi, and D. Katagami, "Perception analysis of motion contributing to individuality using kinect sensor," in *RO-MAN, 2012 IEEE*, 2012, pp. 308–313.

[35] H. Hai, L. Bin, H. Benxiong, and C. Yi, "Interaction system of treadmill games based on depth maps and cam-shift," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 219–222.

[36] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "Holodesk: direct 3d interactions with a situated see-through display," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 2421–2430. [Online]. Available: http://doi.acm.org/10.1145/2208276.2208405

[37] S. Connell, P.-Y. Kuo, L. Liu, and A. M. Piper, "A wizard-of-oz elicitation study examining child-defined gestures with a whole-body interface," in *Proceedings of the 12th International Conference on Interaction Design and Children*, ser. IDC '13. New York, NY, USA: ACM, 2013, pp. 277–280. [Online]. Available: http://doi.acm.org/10.1145/2485760.2485823

[38] K. K. Biswas and S. Basu, "Gesture recognition using microsoft kinect," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, 2011, pp. 100–103.

[39] M. Tang, "Recognizing hand gestures with microsofts kinect," http://www.stanford.edu/class/ee368/Project_11/Reports/Tang_Hand_Gesture_Recognition.pdf, Department of Electrical Engineering, Stanford University, Tech. Rep. 20130423a, 2011.

[40] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 20:1–20:7. [Online]. Available: http://doi.acm.org/10.1145/2141622.2141647

[41] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *Multimedia, IEEE Transactions on*, vol. 15, no. 5, pp. 1110–1120, 2013.

[42] B.-J. Chen, C.-M. Huang, T.-E. Tseng, and L.-C. Fu, "Robust head and hands tracking with occlusion handling for human machine interaction," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2141–2146.

[43] V. Frati and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics," in *World Haptics Conference (WHC), 2011 IEEE*, 2011, pp. 317–321.

[44] P. Trindade, J. Lobo, and J. Barreto, "Hand gesture recognition using color and depth images

enhanced with hand angular pose data," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, 2012, pp. 71–76.

[45] T. Wan, Y. Wang, and J. Li, "Hand gesture recognition system using depth data," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, 2012, pp. 1063–1066.

[46] R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant, "Validity of the microsoft kinect for assessment of postural control," *Gait and posture*, vol. 36, no. 3, pp. 372–377, 2012.

[47] F. Kondori, S. Yousefi, H. Li, S. Sonning, and S. Sonning, "3d head pose estimation using the kinect," in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, 2011, pp. 1–4.

[48] S. Monir, S. Rubya, and H. Ferdous, "Rotation and scale invariant posture recognition using microsoft kinect skeletal tracking feature," in *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, 2012, pp. 404–409.

[49] F. D. Zainordin, H. Y. Lee, N. A. Sani, Y. M. Wong, and C. S. Chan, "Human pose recognition using kinect and rule-based system," in *World Automation Congress (WAC), 2012*, 2012, pp. 1–6.

[50] T. Chaves, L. Figueiredo, A. Gama, C. de Araujo, and V. Teichrieb, "Human body motion and gestures recognition based on checkpoints," in *Virtual and Augmented Reality (SVR), 2012 14th Symposium on*, 2012, pp. 271–278.

[51] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, "Real-time gesture recognition from depth data through key poses learning and decision forests," *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, vol. 0, pp. 268–275, 2012.

[52] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from rgbd images," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 7–12.

[53] Z. Xiao, F. Mengyin, Y. Yi, and L. Ningyi, "3d human postures recognition using kinect," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, vol. 1, 2012, pp. 344–347.

[54] N. Iwane, "Arm movement recognition for flag signaling with kinect sensor," in *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*, 2012, pp. 86–90.

[55] C.-Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. Sawchuk, and A. Rizzo, "Towards pervasive physical rehabilitation using microsoft kinect," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, 2012, pp. 159–162.

[56] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. Rizzo, and M. Bolas, "Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor," pp. 1831–1834, 2011.

[57] B. Lange, S. Koenig, E. McConnell, C. Chang, R. Juang, E. Suma, M. Bolas, and A. Rizzo, "Interactive game-based rehabilitation using the microsoft kinect," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, 2012, pp. 171–172.

[58] E. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas, "FAAST: the flexible action and articulated skeleton toolkit," in *IEEE Virtual Reality*, Singapore, Mar. 2011, p. 245246. [Online]. Available: http://ict.usc.edu/pubs/FAAST-%20The%20Flexible%20Action%20and%20Articulated%20Skeleton%20Toolkit.pdf

[59] M.-C. Huang, W. Xu, Y. Su, B. Lange, C.-Y. Chang, and M. Sarrafzadeh, "Smartglove

for upper extremities rehabilitative gaming assessment," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012, pp. 20:1–20:4. [Online]. Available: http://doi.acm.org/10.1145/2413097.2413122

[60] M. Abdur Rahman, A. M. Qamar, M. A. Ahmed, M. Ataur Rahman, and S. Basalamah, "Multimedia interactive therapy environment for children having physical disabilities," in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, ser. ICMR '13. New York, NY, USA: ACM, 2013, pp. 313–314. [Online]. Available: http://doi.acm.org/10.1145/2461466.2461522

[61] J. Chapinal Cervantes, F. L. G. Vela, and P. P. Rodrĺguez, "Natural interaction techniques using kinect," in *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*, ser. INTERACCION '12. New York, NY, USA: ACM, 2012, pp. 14:1–14:2. [Online]. Available: http://doi.acm.org/10.1145/2379636.2379650

[62] S. Saini, D. Rambli, S. Sulaiman, M. Zakaria, and S. Shukri, "A low-cost game framework for a home-based stroke rehabilitation system," in *Computer Information Science (ICCIS), 2012 International Conference on*, vol. 1, 2012, pp. 55–60.

[63] M. Gotsis, V. Lympouridis, D. Turpin, A. Tasse, I. Poulos, D. Tucker, M. Swider, A. Thin, and M. Jordan-Marsh, "Mixed reality game prototypes for upper body exercise and rehabilitation," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, 2012, pp. 181–182.

[64] L. Pedro and G. de Paula Caurin, "Kinect evaluation for human body movement analysis," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012, pp. 1856–1861.

[65] A. Bo, M. Hayashibe, and P. Poignet, "Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 3479–3483.

[66] A. Bigdelou, T. Benz, L. Schwarz, and N. Navab, "Simultaneous categorical and spatio-temporal 3d gestures using kinect," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, 2012, pp. 53–60.

[67] L. Gallo, A. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data: Experiencing the kinect," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, 2011, pp. 1–6.

[68] G. Mastorakis and D. Makris, "Fall detection system using kinects infrared sensor," *Journal of Real-Time Image Processing*, pp. 1–12, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11554-012-0246-9

[69] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, ser. ICOST'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 121–128. [Online]. Available: http://dl.acm.org/citation.cfm?id=2026187.2026206

[70] Z.-P. Bian, L.-P. Chau, and N. Magnenat-Thalmann, "Fall detection based on skeleton extraction," in *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, ser. VRCAI '12. New York, NY, USA: ACM, 2012, pp. 91–94. [Online]. Available: http://doi.acm.org/10.1145/2407516.2407544

[71] Z. Zhang, W. Liu, V. Metsis, and V. Athitsos, "A viewpoint-independent statistical method for fall detection." in *ICPR*. IEEE, 2012, pp. 3626–3630. [Online]. Available: http://dblp.uni-trier.de/db/conf/icpr/icpr2012.html#ZhangLMA12

[72] E. Stone and M. Skubic, "Passive, in-home gait measurement using an inexpensive depth camera: Initial results," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, 2012, pp. 183–186.

[73] G. Parra-Dominguez, B. Taati, and A. Mihailidis, "3d human motion analysis to detect abnormal events on stairs," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, 2012, pp. 97–103.

[74] B. Ni, N. C. Dat, and P. Moulin, "Rgbd-camera based get-up event detection for hospital fall prevention," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 1405–1408.

[75] R. El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, 2012, pp. 1280–1288.

[76] C. Hoilund, V. Kruger, and T. Moeslund, "Evaluation of human body tracking system for gesture-based programming of industrial robots," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, 2012, pp. 477–480.

[77] V. V. Nguyen and J.-H. Lee, "Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot," in *System Integration (SII), 2012 IEEE/SICE International Symposium on*, 2012, pp. 93–98.

[78] D. Xu, Y.-L. Chen, C. Lin, X. Kong, and X. Wu, "Real-time dynamic gesture recognition system based on depth perception for robot navigation," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, 2012, pp. 689–694.

[79] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao, "Design and implementation of human-robot interactive demonstration system based on kinect," in *Control and Decision Conference (CCDC), 2012 24th Chinese*, 2012, pp. 971–975.

[80] J. Boyd, A. Godbout, and C. Thornton, "In situ motion capture of speed skating: Escaping the treadmill," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, 2012, pp. 460–467.

[81] E. Machida, M. Cao, T. Murao, and H. Hashimoto, "Human motion tracking of mobile robot with kinect 3d sensor," in *SICE Annual Conference (SICE), 2012 Proceedings of*, 2012, pp. 2207–2211.

[82] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Mechatronics (ICM), 2011 IEEE International Conference on*, 2011, pp. 358–362.

[83] F. Wang, C. Tang, Y. Ou, and Y. Xu, "A real-time human imitation system," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, 2012, pp. 3692–3697.

[84] F. Zuher and R. Romero, "Recognition of human motions for imitation and control of a humanoid robot," in *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, 2012, pp. 190–195.

[85] M. Popa, A. Koc, L. Rothkrantz, C. Shan, and P. Wiggers, "Kinect sensing of shopping related actions," in *Constructing Ambient Intelligence: AmI 2011 Workshops*, undefined, K. Van Laerhoven, and J. Gelissen, Eds., Amsterdam, Netherlands, 11 2011.

[86] L. Wang, R. Villamil, S. Samarasekera, and R. Kumar, "Magic mirror: A virtual handbag shopping system," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 19–24.

[87] G. Galatas, G. Potamianos, and F. Makedon, "Audio-visual speech recognition using depth information from the kinect in noisy video conditions," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012, pp. 2:1–2:4. [Online]. Available: http://doi.acm.org/10.1145/2413097.2413100

[88] M. d. S. Anjo, E. B. Pizzolato, and S. Feuerstack, "A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect," in *Proceedings of the*

11th Brazilian Symposium on Human Factors in Computing Systems, ser. IHC '12. Porto Alegre, Brazil, Brazil: Brazilian Computer Society, 2012, pp. 259–268. [Online]. Available: http://dl.acm.org/citation.cfm?id=2393536.2393574

[89] C. Martin, D. Burkert, K. Choi, N. Wieczorek, P. McGregor, R. Herrmann, and P. Beling, "A real-time ergonomic monitoring system using the microsoft kinect," in *Systems and Information Design Symposium (SIEDS), 2012 IEEE*, 2012, pp. 50–55.

[90] T. Dutta, "Evaluation of the kinect sensor for 3-d kinematic measurement in the workplace," *Applied Ergonomics*, vol. 43, no. 4, pp. 645 – 649, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003687011001529

[91] A. Chatterjee, S. Jain, and V. M. Govindu, "A pipeline for building 3d models using depth cameras," in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, ser. ICVGIP '12. New York, NY, USA: ACM, 2012, pp. 38:1–38:8. [Online]. Available: http://doi.acm.org/10.1145/2425333.2425371

[92] S. Farag, W. Abdelrahman, D. C. Creighton, and S. Nahavandi, "Extracting 3d mesh skeletons using antipodal points locations," in *UKSim*, 2013, pp. 135–139.

[93] L. Norrie and R. Murray-Smith, "Virtual sensors: rapid prototyping of ubiquitous interaction with a mobile phone and a kinect," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '11. New York, NY, USA: ACM, 2011, pp. 25–28. [Online]. Available: http://doi.acm.org/10.1145/2037373.2037378

[94] D. G. Rodrigues, E. Grenader, F. d. S. Nos, M. d. S. Dall'Agnol, T. E. Hansen, and N. Weibel, "Motiondraw: a tool for enhancing art and performance using kinect," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: ACM, 2013, pp. 1197–1202. [Online]. Available: http://doi.acm.org/10.1145/2468356.2468570

[95] K. Funes Mora and J. Odobez, "Gaze estimation from multimodal kinect data," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 25–30.

[96] M. B. Winkler, M. Hver, A. Hadjako, and M. Mhlhuser, "Automatic camera control for tracking a presenter during a talk," in *2012 IEEE International Symposium on Multimedia*, December 2012.

[97] M. N. K. Boulos, B. J. Blanchard, C. Walker, J. Montero, A. Tripathy, and R. Gutierrez-Osuna, "Web gis in practice x: a microsoft kinect natural user- google earth," *International Journal of Health Geographics*, vol. 10, no. 45, pp. 2566 – 2570, 2011. [Online]. Available: http://www.ij-healthgeographics.com/content/10/1/45

[98] D. Liebling and M. R. Morris, "Kinected browser: depth camera interaction for the web," in *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ser. ITS '12. New York, NY, USA: ACM, 2012, pp. 105–108. [Online]. Available: http://doi.acm.org/10.1145/2396636.2396652

[99] "Primesense,ltd," http://www.primesense.com, 2014.

[100] N. Pittman, A. Forin, A. Criminisi, J. Shotton, and A. Mahram, "Image segmentation using hardware forest classifiers," in *Microsoft Research on-line publication*, 2013.

[101] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman *et al.*, "Efficient human pose estimation from single depth images," in *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013, pp. 175–192.

[102] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Underst.*, vol. 115, no. 2, pp. 224–241, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2010.10.002

[103] S. Apewokin, B. Valentine, D. Forsthoefel, L. Wills, S. Wills, and A. Gentile, "Embedded real-time surveillance using multimodal mean background modeling," in *Embedded Computer Vision*, ser. Advances in Pattern Recognition, B. Kisaanin, S. Bhattacharyya, and S. Chai, Eds. Springer London, 2009, pp. 163–175. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-304-0_8

[104] D. Brooks, Y. ping Chen, and A. Howard, "Simulation versus embodied agents: Does either induce better human adherence to physical therapy exercise?" in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012, pp. 1715–1720.

[105] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[106] M. Rohith and C. Kambhamettu, "Augmenting monocular motion estimation using intermittent 3d models from depth sensors," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 473–476.

[107] J. Fayad, C. Russell, and L. Agapito, "Automated articulated structure and 3d shape recovery from point correspondences," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 431–438.

[108] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 32–36. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2004.747

[109] Carnegie mellon university motion capture databases. [Online]. Available: http://http://mocap.cs.cmu.edu/

[110] P. Schäfer, *Towards Time Series Classification without Human Preprocessing*. Cham: Springer International Publishing, 2014, pp. 228–242. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08979-9_18

[111] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2007.70711

[112] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.

[113] J. Blackburn and E. Ribeiro, "Human motion recognition using isomap and dynamic time warping," in *Human motion–understanding, modeling, capture and animation*. Springer, 2007, pp. 285–298.

[114] K. Adistambha, C. Ritz, and I. Burnett, "Motion classification using dynamic time warping," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, 2008, pp. 622–627.

[115] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning," *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, vol. 5, no. 6, p. 12, 2011.

[116] A. Criminisi and J. Shotton, Eds., *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013.

[117] V. Vapnik, *The nature of statistical learning theory*. Springer Science Business Media.

[118] E. Ceseracciu, Z. Sawacha, and C. Cobelli, "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept," *PloS one*, vol. 9, no. 3, p. e87640, 2014.

[119] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *Int. J. Comput. Vision*, vol. 87, no. 1-2, pp. 156–169, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0284-3

[120] A. Mansur, Y. Makihara, and Y. Yagi, "Inverse dynamics for action recognition," *Cybernetics, IEEE Transactions on*, vol. 43, no. 4, pp. 1226–1236, 2013.

[121] L. Zhang, J.-C. Hsieh, and J. Wang, "A kinect-based golf swing classification system using hmm and neuro-fuzzy," in *Computer Science and Information Processing (CSIP), 2012 International Conference on*, 2012, pp. 1163–1166.

[122] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images." in *ICRA*. IEEE, 2012, pp. 842–849. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2012.html#SungPSS12

[123] C. Myers, L. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 6, pp. 623–635, 1980.

[124] C. Waithayanon and C. Aporntewan, "A motion classifier for microsoft kinect," in *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, 2011, pp. 727–731.

[125] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 415–422.

[126] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 188:1–188:12, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366207

[127] R. C. B. Madeo, C. A. M. Lima, and S. M. Peres, "Gesture unit segmentation using support vector machines: segmenting gestures from rest positions," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 46–52. [Online]. Available: http://doi.acm.org/10.1145/2480362.2480373

[128] O. Patsadu, C. Nukoolkit, and B. Watanapa, "Human gesture recognition using kinect camera," in *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, 2012, pp. 28–32.

[129] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[130] Y. M. Lui, "A least squares regression framework on manifolds and its application to gesture recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 13–18.

[131] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[132] S.-Y. Lin, C.-K. Shie, S.-C. Chen, M.-S. Lee, and Y.-P. Hung, "Human action recognition using action trait code," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 3456–3459.

[133] R. Sivalingam, G. Somasundaram, V. Bhatawadekar, V. Morellas, and N. Papanikolopoulos, "Sparse representation of point trajectories for action classification," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 3601–3606.

[134] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, "Exemplar-based human action pose correction and tagging," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1784–1791.

[135] H. P. Shum, E. S. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *Cybernetics, IEEE Transactions on*, vol. 43, no. 5, pp. 1357–1369, 2013.

[136] S. Asteriadis, A. Chatzitofis, D. Zarpalas, D. S. Alexiadis, and P. Daras, "Estimating human motion from multiple kinect sensors," in *Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*. ACM, 2013, p. 3.

[137] E. Velloso, A. Bulling, and H. Gellersen, "Motionma: motion modelling and analysis by demonstration," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 1309–1318.

[138] C.-J. Su, "Personal rehabilitation exercise assistant with kinect and dynamic time warping," *International Journal of Information and Education Technology*, pp. 448–454, 2013.

[139] Š. Obdržálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population," in *Engineering in medicine and biology society (EMBC), 2012 annual international conference of the IEEE*. IEEE, 2012, pp. 1188–1193.

[140] K. Gould and M. Shah, "The trajectory primal sketch: A multi-scale scheme for representing motion characteristics," in *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*. IEEE, 1989, pp. 79–85.

[141] M. K. Leung and Y.-H. Yang, "Human body motion segmentation in a complex scene," *Pattern recognition*, vol. 20, no. 1, pp. 55–64, 1987.

[142] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2847–2854.

[143] M. C. Domingo, "An overview of the internet of things for people with disabilities," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 584–596, 2012.

[144] I. M. Pires, N. M. Garcia, N. Pombo, and F. Flórez-Revuelta, "From data acquisition to data fusion: a comprehensive review and a roadmap for the identification of activities of daily living using mobile devices," *Sensors*, vol. 16, no. 2, p. 184, 2016.

[145] A. Grünerbl, A. Muaremi, V. Osmani, G. Bahle, S. Oehler, G. Tröster, O. Mayora, C. Haring, and P. Lukowicz, "Smartphone-based recognition of states and state changes in bipolar disorder patients," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 140–148, 2015.

[146] M. Altini, R. Vullers, C. Van Hoof, M. van Dort, and O. Amft, "Self-calibration of walking speed estimations using smartphone sensors," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE, 2014, pp. 10–18.

[147] W.-J. Yi, O. Sarkar, S. Mathavan, and J. Saniie, "Wearable sensor data fusion for remote health assessment and fall detection," in *Electro/Information Technology (EIT), 2014 IEEE International Conference on*. IEEE, 2014, pp. 303–307.

[148] A. Dimitrievski, E. Zdravevski, P. Lameski, and V. Trajkovik, "Towards application of non-invasive environmental sensors for risks and activity detection," in *Intelligent Computer Communication and Processing (ICCP), 2016 IEEE 12th International Conference on*. IEEE, 2016, pp. 27–33.

[149] M. Heinz, J. Cho, N. Kelly, P. Martin, J. Wong, W. Franke, W.-H. Hsieh, and J. Blaser, "The potential of three computer-based communication activities for supporting older adult independent living," *Information*, vol. 7, no. 2, 2016.

[150] S. I. Lee, M. Y. Ozsecen, L. Della Toffola, J.-F. Daneault, A. Puiatti, S. Patel, and P. Bonato, "Activity detection in uncontrolled free-living conditions using a single accelerometer," in *Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 1–6.

[151] D. Hagedorn and E. Holm, "Effects of traditional physical training and visual computer feedback training in frail elderly patients. a randomized intervention study," *European journal of physical and rehabilitation medicine*, vol. 46, no. 2, p. 159168, June 2010. [Online]. Available: http://europepmc.org/abstract/MED/20485221

[152] J. Parker, S. Mawson, G. Mountain, N. Nasr, and H. Zheng, "Stroke patients utilisation of extrinsic feedback from computer-based technology in the home: a multiple case study realistic evaluation," *BMC medical informatics and decision making*, vol. 14, no. 1, p. 46, 2014.

[153] M. D. Popovi, M. D. Kosti, S. Z. Rodi, and L. M. Konstantinovi, "Feedback-mediated upper extremities exercise: Increasing patient motivation in poststroke rehabilitation,."

[154] P. De Filippi and S. McCarthy, "Cloud computing: Centralization and data sovereignty," *European Journal of Law and Technology.*, vol. 3, no. 2, Oct. 2012. [Online]. Available: https://ssrn.com/abstract=2167372

[155] W. Zhao, "A concise tutorial on human motion tracking and recognition with microsoft kinect," *Science China Information Sciences*, vol. 59, no. 9, p. 93101, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11432-016-5604-y

[156] W. Zhao, D. D. Espy, M. A. Reinthal, and H. Feng, "A feasibility study of using a single kinect sensor for rehabilitation exercises monitoring: A rule based approach," in *Computational Intelligence in Healthcare and e-health (CICARE), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–8.

[157] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Systems*, vol. 20, no. 1, pp. 81–99, 2014.

[158] R. M. Araujo, G. Graña, and V. Andersson, "Towards skeleton biometric identification using the microsoft kinect sensor," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 21–26.

[159] W. Zhao, D. D. Espy, M. A. Reinthal, and H. Feng, "A feasibility study of using a single kinect sensor for rehabilitation exercises monitoring: A rule based approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference*, 2012, pp. 1784–1791.

[160] TED-Wiley, "Statistics: Visualizing data," http://www.media.wiley.com/assets/7083/48/TLGIntroEssay.pdf, Media Wiley, Tech. Rep.

[161] Centerline-Digial. The importance of data visualization. [Online]. Available: www.slideshare.net/Centerline_Digital/the-importance-of-data-visualization

[162] K. Roghwell. Visualize this: the benefits of data visualization. [Online]. Available: http://www.producersweb.com/r/pwebmc/d/contentFocus/?pcID=c6291072b1727db9b2d219921a733abf&pn=2

APPENDIX

APPENDIX A

Publications

- Roanna Lun, Wenbing Zhao, "A Survey of applications and human motion recognition with Microsoft Kinect," [8] *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 05, p. 1555008, 2015.

- Wenbing Zhao,Roanna Lun, Connor Gordon, A. B. M. Fofana, D. D. Espy, M. A. Reinthal,B. Ekelman,G. D. Goodman,J. E. Niederriter and X. Luoand, "A Human-Centered Activity Tracking System: Toward a Healthier Workplace," [9] in *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, June 2017, pp.1784 - 1791.

- Roanna Lun, Wenbing Zhao, "Kinect Applications in Healthcare," [10] in *Encyclopedia of Information Science and Technology, Fourth Edition. IGI Global*, March 2018.

- Roanna Lun, Connor Gordon and Wenbing Zhao, "Tracking the activities of daily lives: an integrated approach," [11] in *Proceedings of Future Technologies Conference (FTC)*, January 2017.

- Wenbing Zhao, Roanna Lun, Connor Gordon, Abou-Bakar M. Fofana, Deborah D. Espy, Ann Reinthal, Beth Ekelman, Glenn D. Goodman, Joan E. Niederriter, Chaomin Luo and Xiong

Luo, "LiftingDoneRight: A Privacy-Aware Human Motion Tracking System for Healthcare Professionals" [12] in *International Journal of Handheld Computing Research (IJHCR)*, March 2016.

- Wenbing Zhao, Roanna Lun and Connor Gordon [13], "The design and implementation of a Kinect-based framework for selective human activity tracking," in *Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 2016.

- Roanna Lun and Wenbing Zhao, "A Survey of Using Microsoft Kinect in Healthcare," [14] in *Encyclopedia of Information Science and Technology, Third Edition. IGI Global*, March 2015

- Wenbing Zhao, Roanna Lun,Deborah D Espy and M Ann Reinthal,"Rule based real-time motion assessment for rehabilitation exercises," [15] in *Proceedings of Computational Intelligence in Healthcare and e-health (CICARE), 2014 IEEE Symposium on* December 2014.

- Roanna Lun, Connor Gordon and Wenbing Zhao, "The design and implementation of a Kinect-based framework for selective human activity," [13] in *Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 2016.

- Wenbing Zhao, Roanna Lun,Deborah D Espy and M Ann Reinthal,"Realtime Motion Assessment For Rehabilitation Exercises: Integration Of Kinematic Modeling With Fuzzy Inference," [16] in *Journal of Artificial Intelligence and Soft Computing Research*, 2014.

- Wenbing Zhao, Hai Feng and Roanna Lun, "A Kinect-based rehabilitation exercise monitoring and guidance system," [17] in *Proceedings of 2014 IEEE 5th International Conference on Software Engineering and Service Science*, October 2014.