PATIENT AMBULATION ASSESSMENT USING DEPTH CAMERAS

by

Zihang You

APPROVED BY SUPERVISORY COMMITTEE:

____
Dr. Mehrdad Nourani, Chair

____
Dr. Lakshman S. Tamil

____
Dr. Terrell R. Bennett

Dedicated to my parents, teachers, colleagues and friends.

Thank you for always supporting me.

PATIENT AMBULATION ASSESSMENT USING DEPTH CAMERAS

by

ZIHANG YOU, BS

THESIS

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2020

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Professor Mehrdad Nourani for his excellent guidance, caring and for providing me an outstanding atmosphere throughout my research. I am very grateful for the opportunity to work in his research lab.

I would like to thank my committee members, Dr. Lakshman S. Tamil and Dr. Terrell R. Bennett for attending my master's thesis defense and their comments and suggestions.

I am pleased to acknowledge my lab colleagues Alec M. Steele, Genchang Peng, Munawara Saiyara Munia and Hafez Eslami Manoochehri, for their invaluable support and assistance.

I also want to thank all the volunteers who participated in the data collection of my projects.

Finally, in particular, I would like to express my gratitude to my parents for their loving considerations and great confidence in me all through these years. Their love was a great help for me to finish this thesis successfully.

November 2020

PATIENT AMBULATION ASSESSMENT USING DEPTH CAMERAS

Zihang You, MS
The University of Texas at Dallas, 2020

Supervising Professor: Dr. Mehrdad Nourani, Chair

In the field of medical care, effective assessment of human activities is essential for understanding the patient's physical state in general and postoperative recovery, in particular. Detecting human joints by using depth cameras and computer vision methods is an important means of assessing human postures and activities. This thesis explores methods of evaluating human activity from three angles by using the second generation of Microsoft Kinect camera.

The first part focuses on quantifying human activity using three metrics: average speed, distance traveled, and postures. In particular, by tracking the human head, its position relative to the camera can be determined with reasonable accuracy. By setting thresholds for different postures, the posture of the human body can also be determined. Accuracy of at least 87% was achieved for distance traveled and average velocity measurements. For posture detection, accuracy of at least 80% was achieved.

The second part demonstrates a subject identity recognition method by measuring the height of the targeted human body and the distances among their joints. The distances between adjacent joints and height of a subject’s head are used to create a vector of eight features for an individual to use for identification. Using a modified KNN, full and partial feature sets were used to identify subjects. The classification results were promising, and the mean accuracy for all subjects reached 95.3%. In the third part, we proposed a method for posture detection based on tracking part of human joints. To differentiate static postures and dynamic movements, a hierarchical classifier is used. By analyzing the relative positions of the tracked joints, key features can be extracted as the basis for static posture classification. In addition, two indicators of speed and acceleration can be used to identify dynamic postures. In the specific classification stage, we used the Support Vector Machine (SVM) method. The performance of SVM shows that the average accuracy of the entire hierarchical classifier is 97.86%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Human Posture Recognition

Human body posture recognition has become an attractive and topical research topic because of its wide application and broad prospects. In general, posture recognition is a process in which sensors receive human biological information and environmental information, and then computers and mathematical algorithms process this information to interpret the human body's posture. Since posture recognition involves the human body and environmental factors, this technique has been applied in many fields, such as healthcare [1, 2, 3], athletics [4], human-computer interaction [5, 6], video surveillance [7, 8], etc.

1.1.1 Common Posture Recognition Methods

Posture recognition methods that are commonly used can be divided into two main categories: wearable and contactless methods. For wearable posture recognition, wearable devices [9, 10] or portable devices [11, 12] such as wristbands and smartphones are commonly used devices. These devices usually include motion sensors such as accelerometers and gyroscopes [11, 12]. After these sensors collect biological information of human body, the information is preprocessed and transmitted to the microcontroller, and the microcontroller will process it and obtain the results. Posture recognition with wearable devices is often used in health detection and sports rehabilitation, because accelerometers and gyroscopes are able to describe the state of motion more accurately than computer vision methods. Another advantage of this method is that these devices and the people wearing them are not limited by space, and these sensors can collect data almost anywhere. For contactless (also called transparent) methods, the more traditional means are through computer vision technology. The first computer vision method which is commonly used is using 3D modeling [13], and a similar technique is extracting features in 2D images to simulate 3D models [14, 15, 16]. Another solution is to extract the skeleton and joint information of the human body by analyzing the human body images, and then perform posture recognition through virtual skeleton and joints [17, 18, 19, 20]. Compared with the previous method, the algorithm based on human skeleton requires less data and calculation, and it is easier for researchers to focus on the body parts they are interested in. However, computer vision-based methods always require the participation of cameras. In the field of video surveillance, people will not have too many concerns. But in the field of personal healthcare, the participation of RGB cameras will make people worry about their privacy. Another relatively novel contactless posture recognition method is indoor positioning technology. Commonly used indoor positioning methods are based on near field communication, including wireless local area networks (WLAN) [21], radio frequency identification (RFID) [10, 22] and ultra-wideband [23]. The commonality of these techniques requires that corresponding electronic tags should be installed on the body or clothes of the observed subject. These tags help mark the coordinates of joints or key parts. After obtaining body information (mainly joint information), posture recognition can be performed in a similar manner to computer vision analysis of skeletons. This method protects people's privacy effectively, but the disadvantage lies in cost and space limitations.

1.1.2 Machine Learning in Posture Recognition

The entire process of posture recognition can be divided into four steps: (i) data collection, (ii) preprocessing, (iii) feature extraction and (iv) classification. Machine learning is mainly used to effectively classify data after feature extraction. There are three common methods used in posture recognition: k-nearest neighbors (KNN), neural networks and support vector machine (SVM).

- K-nearest neighbors algorithm is a non-parametric statistical method for classification and regression [24]. Training samples are usually vectors within a multi-dimensional feature space, and each vector has a label. In the training phase, only the feature vector and the class label of the training sample need to be stored. In the classification stage, the custom constant *k* is defined by user. Each unlabeled vector is classified by assigning the label which appears the most frequently among the *k* training samples closest to the query point. KNN is a very simple model, and only a few parameters need to be tuned.
- Neural networks consist of a collection of artificial neurons, which simulates the network of neurons in human brains. There are different types of neuron models that are commonly studied, including the perceptron, the sigmoid neuron and rectified linear units. A neural network is typically a directed graph consisting of a collection of neurons (the nodes in the graph), directed edges (each with an associated weight), and a collection of fixed binary inputs. In posture recognition, neural networks are often used because they have strong flexibility and are suitable for multiple input and multiple output systems with multiple input network structures [25].
- Support vector machine (SVM) is a popular machine learning method. It is widely used in both linear and nonlinear classification. In nonlinear classification, SVM can use kernel

functions which mapping inputs into high-dimensional feature spaces implicitly. In a high-dimensional space, all categories to be classified can be solved using linear classification. Since SVM can reduce the number of misclassifications, has high performance rate and good generalized ability, it is commonly used in posture classification.

1.2 Introduction and Applications of Depth Camera

As mentioned above, it is an effective method for posture recognition by extracting human skeleton information and joint position information. One of the main approaches to obtain this information is to use depth cameras.

1.2.1 Background of Depth Camera

Depth camera is a specific application of range imaging technology. Range imaging refers to the image produced by using the distance between a specific point (the location of some kind of sensor) and related points in the scene. The pixel values of the image correspond to the distance. If the sensor used to generate the distance image has been calibrated, the pixel value can be directly converted into the corresponding physical unit (e.g. feet). The principles that depth cameras use vary, including stereo triangulation, sheet of light triangulation, structure light, time-of-flight, etc. This part mainly discusses two imaging principles commonly used in posture recognition: structure light and time-of-flight.

- Structured light is a group of system structures composed of a projector and a camera. After the projector projects specific light onto the surface of the object and the background, the light is collected by the camera. The position and depth of the objects are calculated according to the change of the light signal caused by the objects, and then the entire three-dimensional space is restored [26]. The first-generation Microsoft Kinect for XBOX 360 and Windows 7 (Kinect V1) is equipped with this technology. In the application of posture recognition, Kinect V1 can recognize the coordinate data of 20 key points of the human body. Therefore, for basic postures, Kinect V1 can effectively complete recognition and classification tasks with the help of machine learning algorithms [17, 27].

- Time-of-flight (ToF) refers to the measurement of the time it takes for an object, particle or wave to travel a certain distance in a medium. According to the information obtained in this period, it can be used to measure velocity or the length of a path. Time-of-flight has various applications in many subjects, including electronics, mass spectrometry, and physics. Specifically, for ToF cameras, each pixel in the depth image they produced corresponds to the distance from each point in the scene to the camera [28].

1.2.2 Microsoft Kinect V2

The second generation of Microsoft Kinect (Kinect V2) is an upgraded version of Kinect V1, launched for XBOX One. Kinect V2 is equipped with ToF technology [29]. Figure 1.1 shows a Kinect V2 and its structure. According to Figure 1, Kinect V2 has three main parts: an RGB color camera, a depth camera and a microphone array.

- RGB Color Camera: The color camera has a resolution of 1920×1080, and the video captured can be displayed with the same resolution on monitor.
- Depth Camera: The depth camera includes 2 parts: the infrared projector and the infrared camera. The infrared projector actively projects the infrared spectrum, and when the spectrum illuminates the object, it will reflect. The infrared camera receives and analyzes the reflected infrared spectrum, and then creates a depth image within the field of view.

- Microphone Array: The microphone array of Kinect V2 contains 4 microphones. This allows it to collect sounds within 180 degrees and determine the direction of the sound source. The sound source can be pointed at 5-degree increments [30].

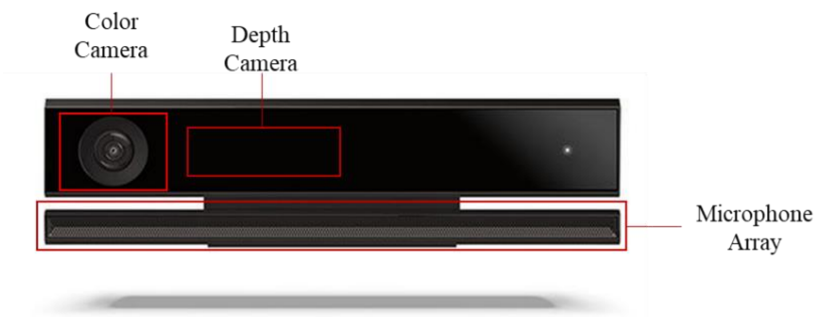

Figure 1.1: Microsoft Kinect V2 and its structure

Table 1.1 [31] compares specifications of Kinect V1 and Kinect V2. The resolution of the two cameras on Kinect V2 is much higher than the previous generation, which results in more image information and fundamentally improves accuracy. Equipped with time-of-flight technology, the depth camera can clearly identify human skeletons even in dark environments. A larger field of view and measurement range can also support recognition of more postures. Up to 25 identifiable joints allow researchers to have more choices in feature extraction. Figure 1.2 demonstrates the top view of the visual range of Kinect V2. It can be seen that although the visible distance of Kinect V2 is from 0.5 m to 4.5 m, the recognition effect reaches the best only when the recognition object is in the blue fan-shaped area, which is called “sweet spot” [30]. Kinect V2 can output three different video streams from two cameras: color stream, infrared stream and depth stream. The three video streams can be viewed separately on the output monitor, or can be viewed superimposed. In all the work involved in this thesis, the depth stream is mainly used. The color stream is used for monitoring and window operations during data collection. For the coordinate data output by Kinect V2, these data are based on its own coordinate system. The coordinate system of Kinect V2 is slightly different from the traditional Cartesian coordinate system. It does not conform to the right-hand rule, but conforms to the left-hand rule. In addition, the depth axis (Z-axis) of the coordinate system of Kinect V2 does not have a negative axis. Figure 1.3 shows the Kinect V2 coordinate system.

Table 1.1: Specifications of Kinect V1 and Kinect V2

Specification		**Kinect V1**	**Kinect V2**
Color Camera	Resolution	640×480	1920×1080
	Refresh Rate	30fps	30fps
Depth Camera	Resolution	320×240	512×424
	Refresh Rate	30fps	30fps
Depth Camera Technology		Structured Light	Time-of-flight
Range		1.2~3.5m	0.5~4.5m
Number of Joints Per Person		20	25
Number of Bodies Tracking Simultaneously		2	6
Angle of View	Horizontal	62°	70°
	Vertical	48.6°	60°

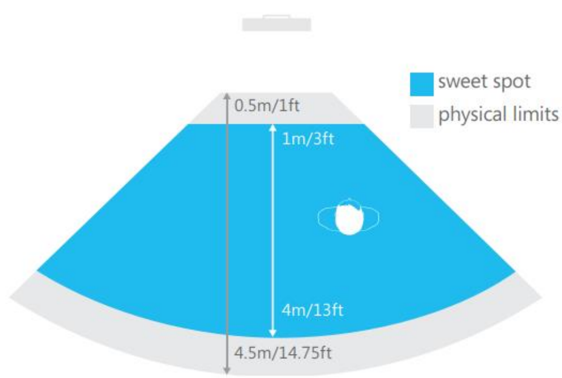

Figure 1.2: The field of view (top view) of Kinect V2

1.3 Contribution and Thesis Organization

Our work mainly focuses on using depth cameras for activity assessment of patients, which includes posture classification, activity quantification and subject identification. Chapter 2 introduces a method of posture classification by using simple thresholds under the condition of detecting the minimum number of joints, and quantifying the ambulation of a patient by detecting the movement of the patient's head.

Chapter 3 proposes a non-traditional computer vision-based approach for subject identification. By obtaining the human body joint information and the distance between the joints output by the depth camera, a vector consisting of multiple features can be constructed. On this basis, we used a modified KNN algorithm to achieve satifactory identification results.

Chapter 4 mainly discusses the use of information of key joints of part of the human body for posture recognition. It is a computer vision solution leverages image processing. By using a hierarchical classifier, the data were greatly streamlined by the depth camera. The program automatically constructs a human skeletal structure in order to calculate velocities and accelerations of specific joints. This classification process is augmented with support vector machine (SVM) so that ten positions can be identified. The methodology is validated by data from multiple volunteers.

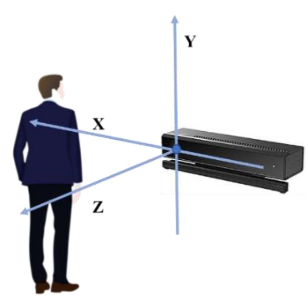

Figure 1.3: The Kinect V2 coordinate system

CHAPTER 2

QUANTIFYING HUMAN ACTIVITY USING HEAD TRACKING

Acknowledgement: The main part of this chapter has been reported in this paper: A. M. Steele, Z. You, M. Nourani, M. M. Bopp, T. S. Taylor and D. H. Sullivan, "Quantifying Human Activity Using Head Tracking," *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, USA, 2019, pp. 1247-1249. The data collection program, posture classification and part of test protocols were done by Zihang You.

2.1 Prior Works

A growing number of applications from athletics to healthcare rely on tracking human activity. Although GPS performs very well in outdoor positioning, its indoor performance is poor. Indoor positioning through near field communication is a possible method, but its cost is relatively high. With the growing number of cameras in public spaces, an obvious solution is to utilize image processing to quantify human activity indoors [32]. Researchers have already successfully determined specific postures using a computer-vision based solution paired with machine learning [32,33]. It is promising that optical sensors with the required components for depth mapping when determining position in 3D space [34]. Posture has also been identified using state-vector machines, bipolar neural networks, and naïve Bayes algorithms [17, 35] by identifying different body limbs. K-means clustering algorithms [36] have been used for connecting groups of postures and specific human activities. These studies have demonstrated the merit of the method of using only optical sensors to complete posture recognition and other human activities.

2.2 General Methodology

A Microsoft Kinect V2 was used to determine the location of a person's head during testing. As outlined in Table 1.1, the refresh rate is up to 30Hz. However, a sample rate of 2Hz is used in this research to prevent error propagation and drift. Only a few joints were used in posture recognition and ambulation quantification. However, in our work, only one joint (the head) was used.

Within the field of view of the camera, 23 positions were chosen to complete the analysis of posture recognition. The distribution of these points is shown in Figure 2.1, and the postures for recognized are indicated in Figure 2.2. The points are separated into three zones: close, mid-range, and far. The closest area to the camera is a "dead zone". At each position, three postures (standing, sitting and lying) were evaluated from three different perspectives: facing toward and away from the camera and facing sideways. Therefore, for one experimental subject, a total of 207 different sets of measurement can be collected. y is the 3D coordinates of the head were matched with estimates of the distance the head, h_p would be from the ground based on the individual's height, h_c is the height of a person's head while seated, and h_b is the height of the bed. p is used to indicate each classified posture. A straightforward thresholding method was applied using a universal threshold, t_h:

$$p = \begin{cases} standing, & \text{if } |y - h_p| \le t_h \\ sitting, & \text{if } |y - h_c| \le t_h \\ lying, & \text{if } |y - h_b| \le t_h \\ unknown, & \text{otherwise} \end{cases} \tag{2.1}$$

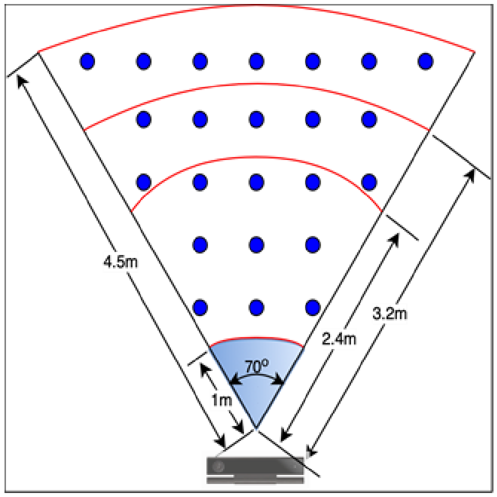

Figure 2.1: All test points relative to the Kinect V2 within its field-of-view

Figure 2.2: Example of all camera angles and postures. On the top all three camera angles are demonstrated standing. From left to right, the subject is facing the camera, facing sideways, and facing away from the camera. At the bottom left, the subject is sitting and lying down at the bottom right.

Within the camera's field of view, three paths with different distances from 5.9 m to 24 m were set up. Thirty-eight different trials in four types were performed using the same volunteer to determine system accuracy to track movement.

- Accuracy Test: The accuracy test is designed to test the system accuracy. The volunteer walks clockwise along the rectangular path shown in Figure 2.3(a). The timekeeper needs to record the elapsed time when the volunteer starts and finishes walking, and when the volunteer reaches each corner of the rectangular path.

- Wheelchair Test: The wheelchair test is designed to help with differentiating between walking and moving in a wheelchair. Being in a wheelchair is considered as sitting. Figure 2.3(b) shows the path of this test. The path is a straight line, with a wheelchair is placed at the starting point. For the first half, the volunteer needs to sit in the wheelchair and moves forward on their own for a round trip. After returning to the starting point, the volunteer will stop, then stand up and walk along the path for another round. The timekeeper needs to record the time that the volunteer gets in and out of the wheelchair, and when the volunteers finishes walking.

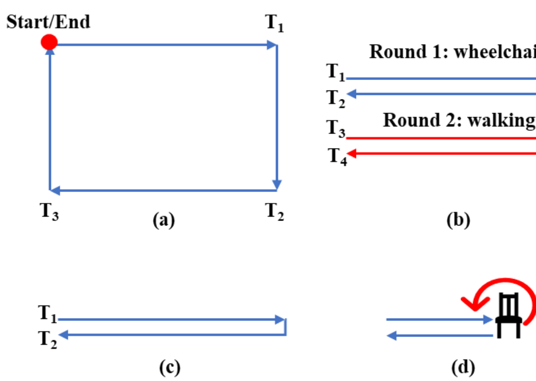

Figure 2.3: General schematic for four tests

- Stability Test: This test is designed to alter the stability of the volunteers by having them hold a heavy object. The volunteer needs to walk along a straight path holding a heavy object for a round trip (Figure 2.3(c)). The timekeeper needs to record the time when starting and ending. The purpose of setting the weight is to observe whether the weight will affect the volunteer's gaits.

- Sitting/Stopping Test: This test is designed to specifically target when a person is sitting versus standing still. The volunteer needs to walk towards a chair, stop for 30 seconds 1-2 feet from the chair, then sit in the chair for 30 seconds, then get up and walk around close to the chair for another 30 seconds, and finally return to the starting point (Figure 2.3(d)).

The distance traveled during each test was determined using the Euclidean distance, *d*, based on Cartesian coordinate pairs in 3D coordinate system, as in Equation (2.2):

$$d = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{2.2}$$

The average velocity was then calculated by dividing the derived distance by the time difference between when the first and last coordinates from the trial data were obtained.

2.3 Results and Validation

Table 2.1 shows the recognition results of the three postures. It can be seen that simple posture recognition performed by tracking only the head has a very good performance, and the error in classifying posture is also acceptable. Since Kinect V2 was developed for the gaming market, it performed better when the person was facing the camera, and the posture recognition was performed worse when the person was facing away from the camera.

Table 2.1: Posture detection results

Walk Characteristic [# of Trials]	**Range (m)**	**Error (%)**			
		Standing	***Sitting***	***Lying***	***Average***
Close [81]	[1,2.4]	33.33[a]	11.11	40.74[a]	28.40
Mid-Range [63]	(2.4,3.2]	19.05	14.29	4.76	12.70
Far [63]	(3.2,4.5]	0	0	0	0
Front [69]	[1.0,4.5]	0	0	17.39	5.80
Side [69]	[1.0,4.5]	4.35	0	13.04	5.80
Away [69]	[1.0,4.5]	18.84	8.70	17.39	14.98
Average	**-**	**18.84**	**8.70**	**17.39**	**14.98**

Table 2.2 shows the activity tracking results. On the whole, Kinect V2 is stable and satisfactory for head tracking. For different walking methods and walking distances, the distance and speed errors show that the tracking error is within an acceptable range. This proves that it is an effective way to use Kinect V2 to track human activity. Error was calculated for each trial based on a comparison of ground truth data, r_g, (i.e. coordinates, distance, timing), which was recorded manually, and the results of analysis of the corresponding system tracking data, r_e. Error was calculated for each trial using:

$$error(\%) = (r_e - r_g) \times 100/r_g \tag{2.3}$$

Table 2.2: Distance and velocity results

Walk Characteristic [# of Trials]	**Mean Distance (m)**	**Error (%)**	
		Distance	***Velocity***
Normal Walk [13]	12.40	14.75	10.01
Slow Walk [13]	12.40	16.84	10.97
Fast Walk [12]	13.50	6.35	10.00
Short Walk [15]	5.90	11.24	11.74
Medium Walk [13]	12.00	15.34	9.89
Long Walk [10]	24.00	11.88	8.91
Average	**12.43**	**12.82**	**10.34**

2.4 Summary

This work gives a solution based on computer vision that leverages image processing using a combination of infrared and visible cameras to accurately and non-invasively determine the position of specific body parts in 3D space. Through identifying a human skeletal structure and isolating the head, distance travelled and average velocity can be obtained. Additionally, the three different general postures are identified: standing, sitting, and lying down. By tracking the coordinates of a person’s head, acceptable results were achieved.

CHAPTER 3

SUBJECT IDENTIFICATION BY USING A DEPTH CAMERA

Acknowledgement: The main part of this chapter has been reported in this paper: A. M. Steele, Z. You, M. Nourani, M. M. Bopp, T. S. Taylor and D. H. Sullivan, "Subject Identification Using a Depth Camera for Patient Ambulation Monitoring," *42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada, 2020, pp. 5745-5748. All parts of this paper except for KNN algorithm and data analysis were done by Zihang You.

3.1 Prior Works

Based on Chapter 2, an improvement of the work is to automatically quantify and record the ambulation of humans in healthcare and rehabilitation settings. Subject identification facilitates the usability of our system in a healthcare environment. With the ability to identify subjects of interest, our system can simultaneously and automatically record data related to multiple individuals, without storing any data other than relevant medical information. Subject identification is a relatively mature technology, which is widely used in many industries and fields. More generally, autonomous driving, production automation, and human-computer interaction benefit from some sort of object recognition. Nowadays, using regular color cameras and thermal imaging cameras is a mainstream method of subject identification [37, 38]. However, large-scale deployment of color cameras to identify objects will inevitably involve personal privacy issues. Conversely, biometric data [39] which does not immediately imply subject identity is much less invasive and can be equally or more effective than RGB image-based approaches. Fingerprint recognition is currently one of the most widely used and most stable identification methods, but it relies heavily on the interaction between the sensor and the identified object. It cannot realize automatic identification. Another example, electroencephalograms (EEG), provides personally identifiable information about subjects [40, 41]. A less invasive but unpopular method, using Doppler radar technology, has also been successfully applied to subject identification in the past by analyzing the differences in breathing among different individuals [42]. The use of infrared depth images for identification is also non-invasive and protects personal privacy better comparing with RGB cameras. Moreover, researchers have successfully proved that this method is effective when applied on the whole body or only on the face [43, 44]. The joint recognition method derived from the depth image is applied to establish gait recognition to identify personal identity, such as in inebriation [45, 46]. At a more subdivided level, relative joint positions are used in smart home for the recognition of daily behavior, such as cooking [47].

3.2 General Methodology

Microsoft Kinect V2 was also used to track and locate human body joints in this work. Although the camera is able to track up to 25 synovial joints, 8 of these joints which located on upper body and arms were applied. According to the description in the Kinect V2 software development kit library, the head and manubrium (top of sternum) are referred as “joints”. Therefore, the 8 different joints in 3D space which applied in our algorithm are the head, manubrium (top of sternum), shoulders, elbows, and wrists. A sample frequency of 2Hz was used during data collection.

The test area is similar to the area used in Section 2.2. Within the field of view of the camera, 20 positions were chosen in order to collect sufficient data to construct a personalized signature. Each point is evenly distributed in the range of 1 m to 4.5 m from the camera. The specific distribution of these points is shown in Figure 3.1.

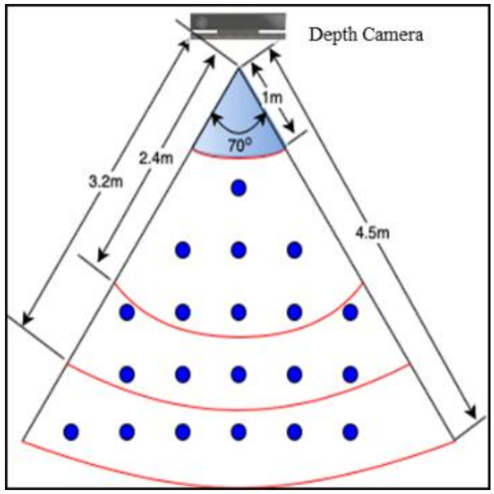

Figure 3.1: Distribution of all points used for testing. The closest area to the camera is a "dead zone".

Multiple volunteers, ages 23 to 59, performed the same two sets of actions for data collection. These two sets of actions are standing still and sitting in a wheelchair. The specific action demonstration is shown in Figure 3.2. During data collection, each volunteer needs to move between these 20 different points, slowly rotates 360 degrees at each point to represent all possible angles of human bodies relative to the camera. In the process of data collection, it is inevitable that some joints cannot be identified because of being blocked. For example, joints on the right arm are invisible when facing right (2^{nd} posture in 1^{st} row of Figure 3.2). However, four or more joints were visible in the majority of cases.

Figure 3.2: Examples of the different test cases used for subject identification. The first row shows a volunteer at various positions while standing. The same positions are demonstrated with the volunteer in a wheelchair in the second row.

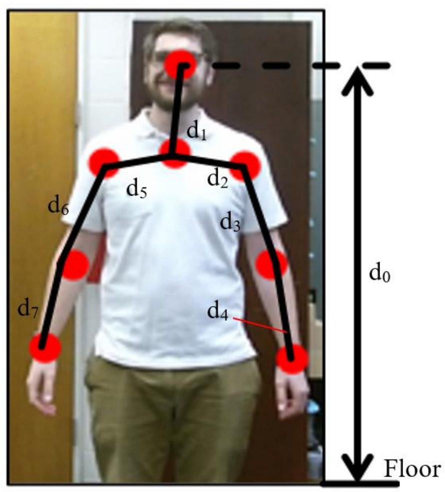

Figure 3.3: All features used for subject identification

The distance d_k (k from 1 to 7) between two adjacent joints a and b was determined using the Euclidean distance formula as followed and head height was determined with by adding the height of the head to the distance of the camera from the ground, eliminating the need to redesign classifiers if a camera changes position. Figure 3.3 shows seven distances d_1 to d_7 between every two adjacent joints and the height d_0 of a person.

$$d_k = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \tag{3.1}$$

For the data collected from each volunteer, we set up a classifier with three categories. The three classes were the same for each person – standing, sitting, and not subject. A modified K-nearest neighbor (KNN) algorithm was selected. In preprocessing part, each sample S is converted into a vector of N features: $S = [s_1, s_2, \ldots, s_N]$. For the ways to calculate the distance in KNN algorithm, the Manhattan distance (the taxicab metric) was chosen. Manhattan distance can increase the emphasis on each individual feature. Therefore, the Manhattan distance d_i between the training vector X, where $X = [x_1, x_2, \ldots, x_N]$, selected from the data set and the sample vector S can be expressed by the following equation:

$$d_i = \sum_{n=1}^{N} |s_i - x_i| \tag{3.2}$$

In the introduction to the KNN algorithm in Subsection 1.1.2, the constant K is self-defined. In this work, the value of the custom constant K is 9 which is determined empirically. Based on the class of the nearest validation data to the test vector, the class of the test vector is predicted. In our application with three classes, when K neighbors are selected, the final classification is based on the highest sum of the inverse square of the distances from each point associated with each of the three classes, c. This can be seen in the following equation, where M is the number of nearest neighbors associated with a given class:

$$max\,\{\sum_{i=1}^{M} \frac{1}{(d_i)^2}\} \tag{3.3}$$

3.3 Results and Validation

A confusion matrix was used to evaluate the performance of the KNN classifier. Accuracy ((TP+TN)/(TP+FP+TN+FN)), sensitivity (TP/(TP+FN)), specificity (TN/(TN+FP)), and F1 score (2TP/(2TP+FP+FN)) were determined for each subject, with respect to each class. TP, FP, TN and FN are number of true positive, false positive, true negative and false negative in classification, respectively. These metrics were based on the sum of all confusion matrices for each subject.

The final confusion matrix was used to determine the metrics mentioned previously. The means of these metrics for each class are presented in Table 3.1. Overall, the classifier is promising when identifying individual subjects. Surprisingly, the best accuracy is achieved when subjects sit. We believe this happens because we collected more data of subjects while sitting due to our data collection protocols (slow rotation in wheelchair compared to standing). The sensitivity of the standing class suggests our algorithm tends to be more conservative when classifying a subject as standing. Specificity of the "not subject" class also supports this argument, further supporting our argument that the larger sitting dataset leads to better results. In this work, we collected approximately 8,000 samples from several subjects. We consider a sample to be data collected from a single frame of the infrared depth camera used. We have provided the confusion matrix used to determine this information in Table 3.2.

Table 3.1: Mean classifier performance for all subjects

Class	**Mean Accuracy (%)**	**Mean Sensitivity (%)**	**Mean Specificity (%)**	**Mean F1 (%)**
Sub. Standing	94.3	76.2	97.4	78.9
Sub. Sitting	98.7	93.4	99.3	93.9
Not Subject	93.0	96.2	83.4	95.4
Mean	**95.3**	**88.6**	**93.4**	**89.4**

When collecting data, joints that were used for feature extraction were commonly obstructed from the view of the depth sensor, often times by a subject's own body. Instead of ignoring this case, we adapted our KNN algorithm based on each test vector. We tested our classifier when five or more features could be extracted. By using the same dataset for all classifications, computation time is only marginally affected. When a test vector only contained a subset of the full feature set, the unobtainable features were ignored in the validation data. However, only validation data with the features present in the test vector were used for each classification. Table 3.3 presents the results of this analysis.

Table 3.2: Confusion matrix for all subjects

		Total Samples	Predicted Class (%)		
			Sub. Standing	**Sub. Sitting**	**Not Subject**
True Class	**Sub. Standing**	330	74.1	0	25.9
	Sub. Sitting	430	0	93.8	6.2
	Not Subject	1122	2.8	0.8	96.4

Table 3.3: Classifier performance by features present

Features (#)	**Total Samples**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 (%)**
4	6250	91.4	78.6	87.8	79.7
5	5672	91.5	78.1	87.6	79.3
6	4400	91.4	77.7	87.3	78.9
7	2429	90.6	74.7	85.21	75.7
8	1882	95.3	88.6	93.4	89.4

These results indicate that having all eight features improves the results considerably; however, having four or more features still achieves reasonable accuracy. This indicates that our chosen feature set has a strong correlation with the classes being determined. For four to six features present, all metrics are very similar. It is important to note that the limited feature sets can contain any combination of features as long as the correct number of features are present, suggesting the significance of each feature is similar. We do believe that head height is the most significant, but more than 95% of collected samples include this feature. Additionally, the physical structure of some subjects (e.g. the forearms) may hold greater significance when both can be observed when compared to other features.

3.4 Summary

In this chapter, we have proposed a subject identification method based on computer vision. By leveraging image processing of an infrared depth map, we non-invasively determine the relative position of specific joints and the height of subjects. By observing the rigid distances between multiple joints above the waist and height of subjects, a unique signature is created. This signature is leveraged to identify a subject based solely on the information provided by the infrared depth map. Additionally, we address potentially incomplete data without sacrificing significant performance.

CHAPTER 4

PATIENT AMBULATION ASSESSMENT USING DEPTH CAMERA

Acknowledgement: The main part of this chapter will soon be submitted: "Patient Ambulation Assessment Using Depth Camera," by Zihang You, Alec M. Steele and Mehrdad Nourani, to be submitted to BiCOB 2021. The main part of the whole work was done by Zihang You.

4.1 Overview

From athletics, to healthcare, a growing number of assessments rely on tracking human activity. Applications such as physical fitness, gait analysis, and personal management can benefit from activity monitoring. Particularly, in healthcare, clinical studies have shown that early and progressive rehabilitation activities effectively reduce physical discomfort and decline in physical function [48]. On the other hand, continuous monitoring requires a large number of nursing staff. In the past decade, there has been a shortage of nursing staff in the United States [49]. According to a research reported by Lisa M. Haddad et al. [50], the number of people over 65 in the United States has reached the highest in history. In 2019, this number was 71 million. According to the US Bureau of Labor Statics, the entire nursing industry in the United States needs approximately 11 million nurses in the next few years. Therefore, advances in technology allowed various categories of electronic devices to be used to assist doctors and nurses to monitor patient's posture and movements in real time.

In Chapter 2, we used a simple threshold method to distinguish three different postures by only using the head coordinates. Meanwhile, by continuously recording the patient's head coordinates, we have successfully quantified the patient's movement trajectory within the camera's field of view, including calculating the patient's movement distance and the average speed of movement over a period of time. In Chapter 2, through 8 selected joints and key points of the upper body of the human body which were identified and located by the infrared camera, we extracted eight features including the height of human body, and realized non-invasive identity recognition by using machine learning methods. For our previous work, a very important limitation is that the camera must be calibrated before use. In addition, the thresholds mentioned in Chapter 2 could only recognize three postures (standing, sitting and lying), which does not meet the needs. In order to automatically quantify and record the ambulation of humans in healthcare and rehabilitation with higher efficiency and accuracy, an improved method will be discussed in this chapter.

4.1.1 Prior Work

Using a camera with computer vision technology to classify and recognize human posture is a common method. A more straightforward way to perform classification and recognition is to directly analyze the images collected by the RGB camera for processing. In O. P. Popoola and K. Wang's review [51], the technology based on the color intelligent monitoring system to identify abnormal postures has been widely used for crime warning in public places and elderly healthcare in smart homes. The work of H. Foroughi [52] and A. H. Nasution et al. [53] proved that the use of gesture recognition technology in smart homes can effectively reduce the risk of death due to falls at home for the elderly with high accuracy. However, real-time video surveillance will raise privacy problems. Therefore, researchers began to use depth camera images instead of RGB images to identify and classify human postures. Among various depth cameras, the Kinect series cameras released by Microsoft Corporation are widely used in posture recognition based on depth images. The work of Z. Xiao et al. [27] used the first-generation of Kinect to restore the 2D depth image to a 3D model, which was very helpful for posture recognition in 3D space. As a typical classification scenario, machine learning is a common method used in posture recognition. The work of D. Xu et al. [54] optimized the depth data collected by Kinect by using support vector machine, and successfully reconstructed a much more accurate 3D human body model to achieve better gesture recognition results. By applying artificial neural networks (ANN), M. D. Štrbac and D. B. Popović have successfully increased the recognition rate of gripping to over 85% [55]. In the work of A. Nandy and P. Chakraborty [56], they used the naive Bayes classifier to achieve exciting results in gait recognition. Going further, the research of A. Amini et al. [57] has proved that depth cameras also have certain applications in fall detection. With the assistance of other sensors, fall detection for the elderly could be detected [58].

4.1.2 Key Contributions

We propose an improved solution based on non-traditional computer vision which leverages image processing using infrared camera only to determine the position of specific body joints accurately and non-invasively in 3D space. By using a hierarchical classifier, we greatly streamlined data collected by the depth camera. By programming Kinect V2, it constructs a human skeletal structure in order to calculate velocities and accelerations of specific joints. This classification process is augmented with support vector machine (SVM) so that ten positions can be identified: standing, standing facing left/right, sitting, lying down, lying sideways, rolling or scooting in a wheelchair, and falling. We also compared performance of SVM algorithm and the threshold-based method.

The methodology shown in this part has been validated by tests performed by multiple participants.

4.2 Methodology

The aim of this work is to use depth cameras to collect depth image information on the basis of the work done in Chapter 2, so as to realize contactless real-time recognition and classification of human body postures without calibrating the cameras. The basis of this project is Kinect V2 equipped with time-of-flight camera. The flow chart of realizing human posture recognition is shown in Figure 4.1.

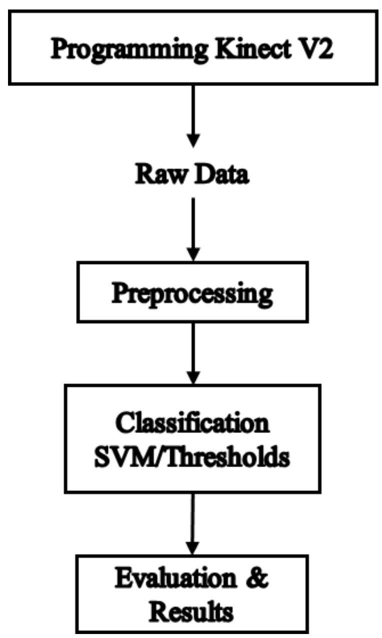

Figure 4.1: Schematic model

In the process of posture recognition, the first step is to program Kinect V2 to obtain the required joint coordinates. In the classification process, in addition to the three-dimensional coordinates of each joint, the distance between adjacent joints, the speed and acceleration of some joints when moving are also features required for classification. The value of these features can be calculated along with the three-dimensional coordinates of the joints when programming Kinect V2. All the above data are combined in the raw data. In the preprocessing stage, the data that meet the requirements are extracted from the raw data and put into the single classifiers and the hierarchical classifier. In this work, a hierarchical classifier with predetermined features is used for classification. For the hierarchical classifier, we use both support vector machine and threshold-based method to do classification and compare their performance. Finally, the classification performance of each decision in the hierarchical classifier, and the overall performance of the hierarchical classifier are evaluated and the final posture recognition results are obtained.

4.2.1 Joint Tracking Method

Although Kinect V2 is capable of tracking 25 synovial joints for one person, 7 joints at most were used for this work. According to the description in the library of Kinect V2 software development kit, the head is referred as a joint. The 7 joints that our algorithm depends on are: the head, 2 hips (pelvis), 2 knees and 2 wrists. The process as followed of identifying and tracking these joints is based on the work by J. Shotton et al. [59].

1) The time-of-flight depth sensor on Kinect V2 will create a depth map by using a projected infrared grid. After sending a modulated infrared signal to the object being observed, the phase shift between the signal being sent and the reflected received is used to determine the depth from the camera to the object in the 3D space [60]. This process will finally result in a grayscale image with the resolution of 512×424 (Table 1.1).
2) For each pixel in the grayscale image, its coordinates in the image are bound to the depth of the specific position in the corresponding 3D space. The pixels of different color depths in the grayscale image reflect the depth of these pixels from the camera. Accordingly, the depth map of the human body can be isolated from the environment.

3) In order to determine the position of the joints on the human body depth map, the human body depth map needs to be divided into multiple parts and numbered. All pixels need to be mapped into these body parts. The boundaries of adjacent body parts are the approximate positions of the joints. The classification process is completed by constructing a random forest with three-layer decision trees. The result is that all pixels have features related to body parts. This step can also be referred as body parts recognition.
4) It is still rough to judge body joints based on the body part features carried by each pixel. Therefore, combinations of pixels are pooled together to generate a more reliable insight into part of the body. A local mode-fitting method based on mean shift [61] with a weighted Gaussian kernel is applied. In this way, the intersection of adjacent bones can be determined as a joint.
5) The depth values of the multiple pixels corresponding to the determined joints from the last step are combined with the coordinate values in grayscale image, and the coordinate values of the joints in the 3D space are determined through conversion.

After selecting and successfully tracking the required human joints, the data collection work can be started.

4.2.2 Data Collection Protocol

This part includes three sub-parts: the development of the data collection program, the data collection area, and the protocols of data collection.

There are certain hardware and software requirements for programming the Kinect V2. The computer used for programming must be equipped with Windows 8/8.1 or newer operating system. The integrated development environment of the program is designated as Microsoft Visual Studio

2013 or newer version. At the hardware level, since the resolution of the two cameras equipped by Kinect V2 is nearly double that of the previous generation, the transmission of video signals must meet the USB 3.0 standard. This standard requires the transmission rate of the interface of the to reach 500MB per second. Kinect V2 supports two programming languages: C++ and C#. In this work, C# is chosen for programming. The C# project type to which the program belongs is a Windows Presentation Foundation (WPF) project. The core namespaces (e.g. *System.Windows*) are imported by default in the WPF project. For developing Kinect V2, except for the core namespaces, the only namespace which is necessary is *Microsoft.Kinect*. In addition, other customized namespaces can be searched and added in the references of the solution explorer.

The logic of the entire program includes the following steps: a) initializing Kinect V2 to have the access to the sensors and to read camera streams, b) reading the streams, c) getting the information of the joints needed, and d) collecting the coordinates and outputting them. Data collected will be formatted in a .csv format table and stored in the terminal.

a) The initialization of Kinect V2 has two steps. The first step is to express *KinectSensor* class and *MultiSourceFrameReader* class in the form of fields, and then activate the sensors.
b) Compared to the three video streams mentioned in Subsection 1.2.2, at the data level, Kinect also provides so-called "body stream" in addition to these three video streams. Through the body stream, Kinect V2 is allowed to transmit the recognized human joint coordinate values to the terminal in real time. When the depth camera detects someone in the field of view, the camera will automatically count the number of people in the field of view. Then the camera starts to output body stream.
c) After obtaining the body stream, the required joint information can be selected in the body

stream and the coordinates of these joints can be obtained. When a human body is tracked by the depth camera, Kinect V2 will continuously output relevant coordinate data. Because the resolution of the depth camera and the color camera are different (Table 1.1), when the depth stream image and the color stream image are superimposed, the joint icons drawn in the color stream cannot be projected correctly. Therefore, Microsoft has added a tool called *CoordinateMapper* in the software development kit. This tool can map joints with three-dimensional coordinates to 2D color video stream. In this way, the color video stream can be used intuitively to observe the tracking of each joint.

d) According to Table 1.1, the refresh rate of Kinect V2 is 30 fps. In other words, for each joint under observation, 30 sets of coordinate values are output to the terminal every second. In C#, the *ArrayList* can perfectly meet the requirements of adding elements in real time. Unlike normal arrays, in an *ArrayList*, items can be added and removed at specified positions by using indexes, and the *ArrayList* will automatically resize it. Because the amount of data brought by 30 sets of coordinate values per second is very large, the remainder method is used for sampling. Taking 6 Hz sampling as an example, in the *ArrayList*, data whose index is divided by 5 and remaining 1 will be finally output to the data set.

A WPF project includes two main parts: a C# program document (.cs file) and a window design document (.xaml file). Figure 4.2 shows the designed window for tracking static postures as an example.

The upper right corner is the time and dates synchronized with the terminal, which can be used as a timer when the human body moves. The right column of buttons is used to record static postures. When starting to record a posture, click the button to leave a mark in the output table. When finishing recording the posture, click the "End" button at the bottom to leave a mark indicating the end in the table. The multiple blue labels at the bottom of the window are used to display the current posture and some joint coordinates in real time.

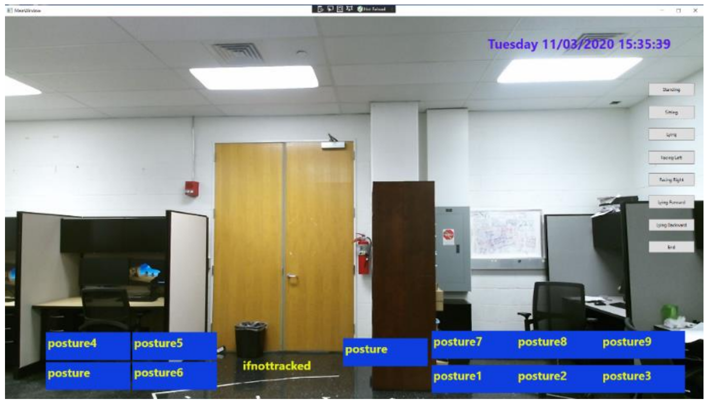

Figure 4.2: The window used for collecting static posture data

The test area is similar to the work in Chapter 3. We have followed the test area and 20 evenly spaced points in Figure 3.1. However, unlike the previous work, the test area used this time is divided into three sub-areas, which are divided by two green straight lines in Figure 4.3. These three sub-areas will be used for fall recognition. Besides, the red rectangular position in the center of the fan-shaped area will be used to place a standard medical bed. This medical bed will be used to identify three types of bed rest positions. Moreover, the two straight intersecting dashed lines in the field of view are used to indicate the paths during rolling and scooting.

During the formal data collection experiment phase, we collected data from several volunteers aged 24 to 59. Under University of Texas at Dallas IRB approval, each volunteer took the same multiple sets of actions in our laboratory. The postures which need to be distinguished are classified into three major categories: static, dynamic and fall. Figure 4.4 shows all the posture examples. Table 4.1 shows the methods that are used to collect data for each kind of posture.

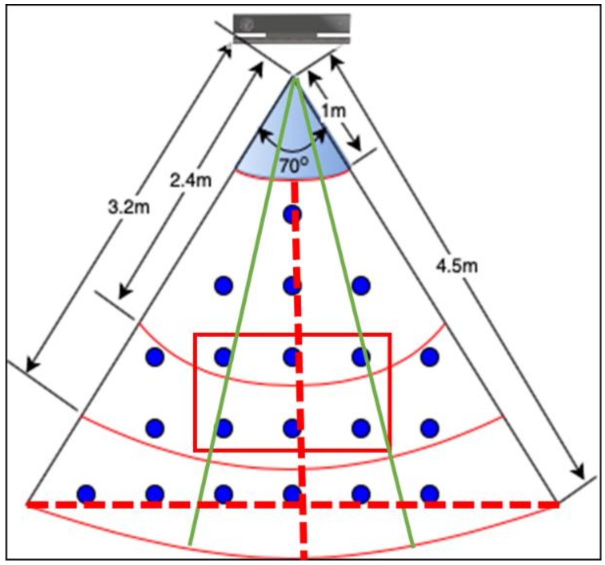

Figure 4.3: The test area for data collection

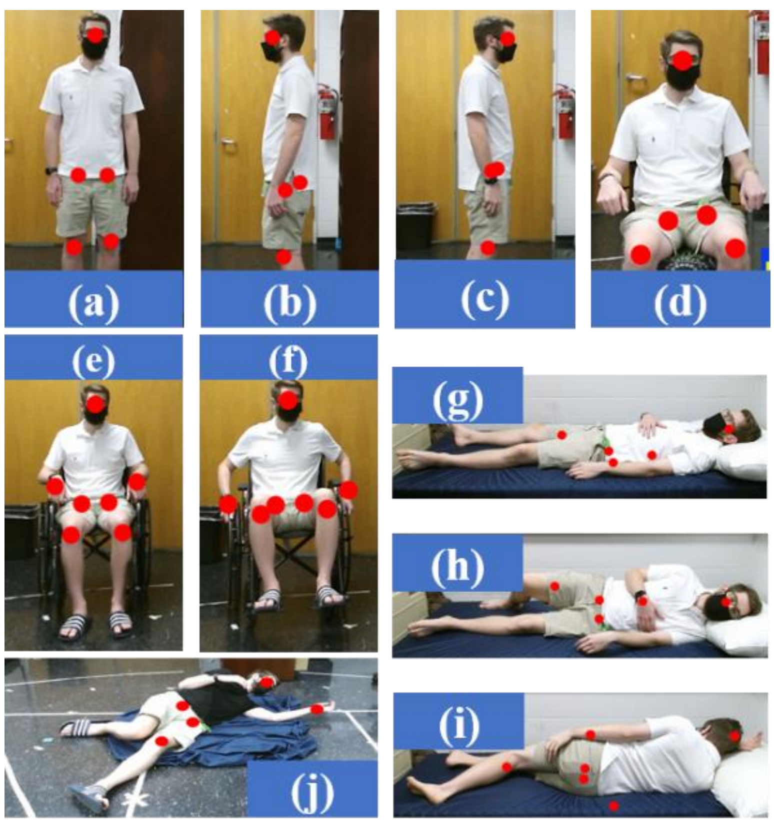

Figure 4.4: 10 different example postures, the red dots are indicated as joints

Table 4.1: Protocols for data collection

Postures	Measurement Protocol
(a) Standing	Turning 360° at each point
(b) Standing Facing Left	Standing still at each point for 15 seconds
(c) Standing Facing Right	Standing still at each point for 15 seconds
(d) Sitting	Turning 360° at each point
(e) Scooting	Scooting 5 round trips along the horizontal and vertical line respectively
(f) Rolling	Rolling 5 round trips along the horizontal and vertical line respectively
(g) Lying Flat	Lying flat for 60 seconds
(h) Lying Facing Forward	Lying for 30 seconds for each posture
(i) Lying Facing Backward	Lying for 30 seconds for each posture
(j) Fall	Making postures after falling under hypothetical situations randomly in each area (Figure 3.3) and holding still for 30 seconds, then standing for another 30 seconds

4.2.3 Hierarchical Classifier

To differentiate static postures and dynamic movements, different features were used, and all the data are classified by a hierarchical classifier with seven branches, as shown in Figure 4.5.

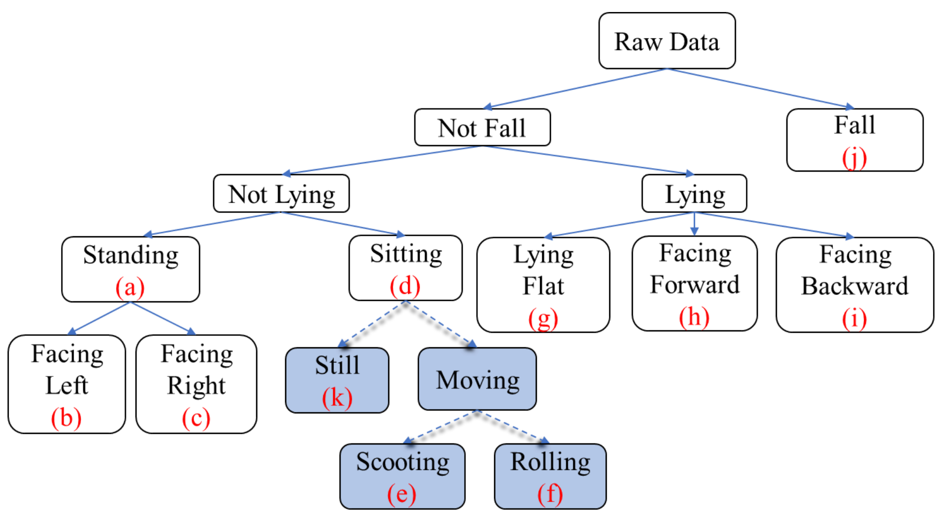

Figure 4.5: The hierarchical classifier used for classification

The advantage of using a hierarchical classifier is that when making decisions on branches, the most representative features can be used to classify based on the result of previous classification.

The two shaded branches with dashed lines are designed to detect dynamic postures. As in Table 4.1, when collecting the data of standing and sitting at various angles, the participants needed to turn 360 degrees at each point. This method would cause both velocity and acceleration to be inevitably included in the data of standing and sitting. Furthermore, the collected data, especially the data of standing and sitting, will interfere the classification results in the shaded branches. Therefore, when performing data analysis, the classification results and performance of dynamic postures will be analyzed separately.

4.2.4 Threshold-based Method

According to our work in Chapter 2, it is feasible to recognize simple postures by using thresholds. Based on the 7 joints tracked in this work mentioned in Subsection 4.2.1, we propose the following basic classification scheme. This threshold-based will be used to classify all postures except for fall detection.

a) Distinguishing Lying

When a human body is lying flat, the head, hips, and knees are approximately in the same horizontal plane. Therefore, when a human body is lying down, the coordinate difference between the head and hips on the Y-axis is significantly smaller than when standing and sitting. According to the coordinates recorded by the depth camera, the distance between head and hips on Y-axis is determined by the Y coordinates of these 3 positions: y_{head}, $y_{hipleft}$ and $y_{hipright}$. The classification of lying posture p is expressed by as below, where d_1 is determined empirically with the value of $d_1 = 0.25m$ in our experimentation but not sensitive in general.

$$p = \begin{cases} lying, & \text{if } \left|y_{head} - y_{hipleft}\right| \leq d_1 \text{ and } \left|y_{head} - y_{hipright}\right| \leq d_1 \\ not\ lying, & \text{otherwise} \end{cases} \tag{4.1}$$

b) Distinguishing Standing and Sitting

When a human body is in a normal standing state, the head, hips and knees are approximately in the same vertical plane. For sitting, the head and hips are approximately in the same vertical plane, while the hips and knees are in the same horizontal plane. Therefore, the sitting posture can be distinguished by detecting the distance change of knees and the hip on Y-axis. d_2 is also determined empirically with the value of $d_2 = 0.25m$ in our experimentation but not sensitive.

$$p = \begin{cases} standing, & \text{if } \left|y_{hipleft} - y_{kneeleft}\right| \geq d_2 \text{ and } \left|y_{hipright} - y_{kneeright}\right| \geq d_2 \\ sitting, & \text{otherwise} \end{cases} \tag{4.2}$$

c) Distinguishing Lying Forward/Backward/Flat

It is considered that the bed is placed horizontally in front of the camera, and it is assumed that the human body is lying with the head toward the right side of the field of view of the camera. Based on the above assumptions, a vector $\boldsymbol{m_{hip}}$ from left hip to right hip is constructed. When lying sideways ((h) and (i) in Figure 4.4), the vector $\boldsymbol{m_{hip}}$ will obviously point to the positive direction (lying forward) of Y-axis and negative direction (lying backward) of Y-axis. For lying flat ((g) in Figure 4.4), $\boldsymbol{m_{hip}}$ will point to the positive direction of Z-axis. The expression for determining the postures by thresholds is given below. The threshold d_3 and d_4 are determined empirically with the value of *-0.12m* and *0.12m* respectively but not sensitive. According to Equation (4.4), only Y coordinates are used in determining lying postures in threshold method. But in the SVM algorithm, all three features from $\boldsymbol{m_{hip}}$ will be used.

$$\boldsymbol{m}_{hip} = \begin{bmatrix} x_{hipright} - x_{hipleft} \\ y_{hipright} - y_{hipleft} \\ z_{hipright} - z_{hipleft} \end{bmatrix} \tag{4.3}$$

$$p = \begin{cases} Forward, & \text{if } y_{hipright} - y_{hipleft} \leq d_3 \\ Backward, & \text{if } y_{hipright} - y_{hipleft} \geq d_4 \\ Flat, & \text{if } d_3 < y_{hipright} - y_{hipleft} < d_4 \end{cases} \tag{4.4}$$

d) Distinguishing Standing Facing Left/Right

The idea of detecting standing sideways is similar to the idea of detecting lying sideways, because the posture of standing can be approximated as the result of a 90-degree rotation of the lying posture. The same vector $\boldsymbol{m}_{hip}$ used in c) is applied in this classification to show the positional relationship of the two hips. The expression for determining the postures by thresholds is given below, where d_3 and d_4 have the same value with Equation (4.4).

$$p = \begin{cases} Left, & \text{if } z_{hipright} - z_{hipleft} \leq d_3 \\ Right, & \text{if } z_{hipright} - z_{hipleft} \geq d_4 \end{cases} \tag{4.5}$$

e) Distinguishing Sitting Still and Moving

In Section 2.1, we have already quantified the movement of the human body with only collecting head coordinates. Therefore, when distinguishing sitting still and moving in a wheelchair, only the head coordinates need to be collected and the average speed of head v_{head} in a short period of time needs to be calculated. Through the three-dimensional coordinates of the i_{th} frame, the adjacent $i\text{-}1_{th}$ frame, and the time Δt between these two frames, the average head speed can be determined. When the average speed of head is larger than the threshold speed, it can be considered that the human body is in a sitting and moving in a wheelchair. The threshold speed v_{th} is set as 0.15m/s in the experiment.

$$v_{head} = \sqrt{\frac{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}{\Delta t}} \tag{4.6}$$

$$p = \begin{cases} Sitting\ Still, & \text{if } v_{head} > v_{th} \\ Moving, & \text{otherwise} \end{cases} \tag{4.7}$$

f) Distinguishing Rolling and Scooting

The definition of scooting is that a patient sits in a wheelchair and moves with his own feet on the ground. The patient's arm should place on the arm of the wheelchair statically. The idea is to detect the relative speed of the head and wrists. Under ideal circumstances, the relative speed between the head and wrist should be 0. By recording the coordinate data of the sampling frame, the average value of speed of the head and wrist in time period Δt can be calculated the same way as the average speed of head. Then, the relative speed would be:

$$v_{relative} = v_{head} - v_{wrist} \tag{4.8}$$

Under the condition that the sitting posture is detected, when the relative speed of the head and wrists is less than the threshold, and the speed for all of them is not zero, we conclude the movement as scooting.

The definition of rolling is that a patient sits in a wheelchair and uses his arms to turn the wheels to move, with his feet placing on the pedals. Unlike Scooting, the arms are doing a rotational motion when the patient is rolling. This results in more difficulties in calculating the relative speed. Therefore, we consider calculating the relative acceleration of the head and wrists. It is assumed that the moving speed of rolling is uniform. Then, the acceleration of the head and wrists can be calculated by the method of successive differences. By recording 7 consecutive frames, 7 coordinates of a joint can be obtained. Then 6 segments of displacement d_1~d_6 (Figure 4.6) in the order of time can be obtained. With knowing the sample rate f, the acceleration a can be deduced from Equations (4.9) and (4.10). The acceleration threshold a_{th} is determined empirically between 9 m/s^2 to 10m/s^2 in our work.

$$d_4 - d_1 = 3a\frac{1}{f^2}, d_5 - d_2 = 3a\frac{1}{f^2}, d_6 - d_3 = 3a\frac{1}{f^2} \tag{4.9}$$

$$a = \frac{(d_4 + d_5 + d_6) - (d_1 + d_2 + d_3)}{9 \times \frac{1}{f^2}} \tag{4.10}$$

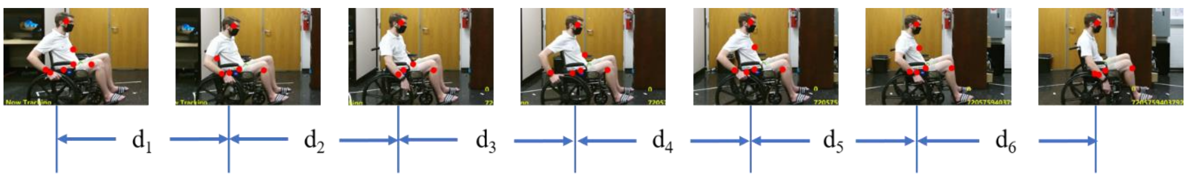

Figure 4.6: Frames and relating distances

The relative acceleration would be the subtraction of head's and wrists' acceleration (4.11).

$$a_{relative} = a_{head} - a_{wrist} \tag{4.11}$$

$$p = \begin{cases} Scooting, & \text{if } v_{relative} < v_{th} \text{ and } a_{relative} < a_{th} \\ Rolling, & \text{otherwise} \end{cases} \tag{4.12}$$

4.2.5 Support Vector Machine (SVM) Classifier

SVM has been generally introduced in Subsection 1.1.2. This part mainly considers two parameters inside the SVM: the kernels and the parameter C. In SVM, the commonly used kernels are as follows:

- Radial basis function (RBF) kernel is widely used in different kinds of kernel learning algorithm. The definition of RBF is as followed, where x and x' are two feature vectors of some input space, and σ is a free parameter [62]. According to (4.13), the value of $K(x, x')$ is depended on the squared Euclidean distance between x and x' with a given σ. RBF kernel is able to map the vectors in the low dimensional space into high dimensional space effectively, in order to have a higher accuracy of linear classification.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma}\right) \tag{4.13}$$

- Linear kernel is a degenerate version of RBF kernel. If an RBF kernel is properly tuned, the linear kernel cannot be more accurate than the RBF kernel [63].
- Polynomial kernel is another typical kernel used to map vectors in the low dimensional space into high dimensional space. For a degree-*d* polynomial, the definition of polynomial kernel is as followed, where x and x' are two feature vectors of some input space, and c is a free parameter.

$$K(x, x') = (x^T x' + c)^d \tag{4.14}$$

The C parameter is a penalty parameter in SVM. It is the weight that adjusts the preference of two indicators (margin of the decision function and classification accuracy) in the optimization, which reflects the tolerance to error. Larger C value means less error could be tolerated, and overfitting happens. On the contrary, smaller may lead to underfitting [64].

In this work, SVM will be used to classify all postures. In fall detection, we focus on the situation after the human body falls on the ground. The three-dimensional coordinates of the 7 joints, a total of 21 features, will all be used as the features of SVM algorithm.

4.3 Experimental Results

4.3.1 Parameter Selection

To determine the parameters in SVM classifier, three kernels mentioned in Subsection 4.2.4 and seven C parameters ranging from 1 to 10^6 were prepared for selection. Figure 4.7 demonstrates the performance of different kernels and parameter C for each branch in the hierarchical classifier with using 10-fold cross validation. It can be seen that in most cases, the performance of RBF kernel is better than or equal to the performance of the other two kernel. For parameter C, in addition to distinguishing standing facing left/right, a larger C value will bring better but not significant performance to the model. Table 4.2 lists the final parameters that used in the evaluation of the classifiers.

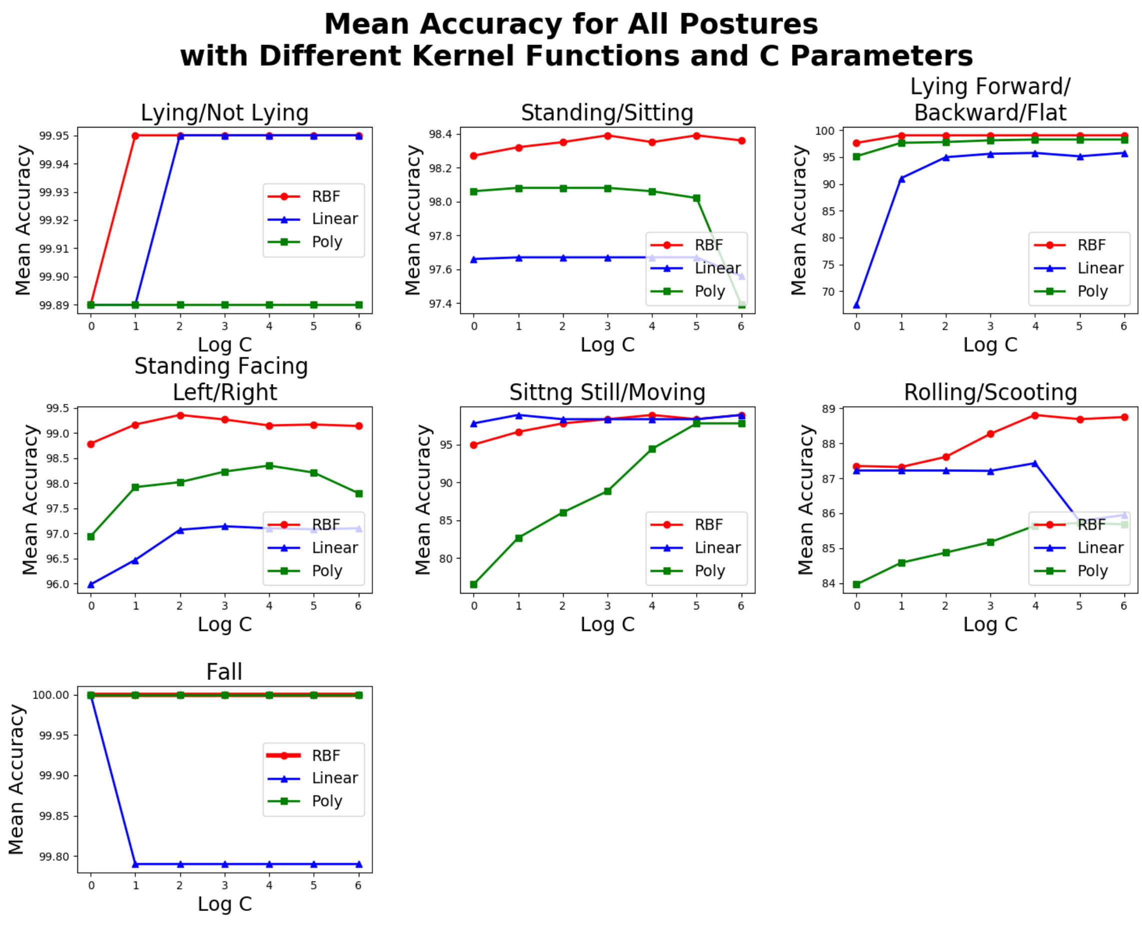

Figure 4.7: Mean accuracy for all postures with different kernel functions and C parameters

Table 4.2: Parameters used in classifications

Classification	Kernel	C Parameter
Lying/Not Lying	RBF Kernel	10
Standing/Sitting		1000
Lying Flat/Forward/Backward		10000
Standing Facing Left/Right		100
Sitting Still/Moving		10000
Rolling/Scooting		10000
Fall/Not Fall		10

4.3.2 Performance of Each Decision

The way of evaluating the performance of each decision in the hierarchical classifier is roughly similar to the way of using a confusion matrix in Section 3.3. The four metrics: accuracy, sensitivity, specificity and F1 score were determined for each subject, with respect to each class. The mean values of F1 score and accuracy of these metrics for each branch in the hierarchical classifier are presented in Table 4.3.

Table 4.3: Mean classifier performance for all subjects

Classification	Mean Accuracy (%)	Mean Sensitivity (%)	Mean Specificity (%)	Mean F1 (%)
Lying/Not Lying	99.97	99.86	99.87	99.90
Standing/Sitting	98.42	99.69	97.36	98.05
Lying Flat/Forward/Backward	99.06	98.94	98.73	99.46
Standing Facing Left/Right	99.47	99.48	99.47	99.49
Sitting Still/Moving	98.88	100.00	96.67	99.17
Rolling/Scooting	89.05	85.21	92.34	88.77
Fall/Not Fall	100.00	100.00	100.00	100.00

In general, the classification of various postures is very promising. The best accuracy is achieved in fall detection. Considering that this part is the beginning of the hierarchical classifier, it can ensure better performance in the lower branches. This result also shows that the joints used for classification of other postures are also very effective in fall detection. The classification performance of rolling and scooting is satisfactory, but not as good as the classification of static actions. On the one hand, when a volunteer is rolling at a slower speed, the speed and acceleration of the wrists are accordingly slowed down. This will cause the acceleration difference between the wrist and the head to become smaller, leading to confusion with scooting. Kinect occasionally drifts during joint recognition. An intuitive reaction is that the joint coordinates will jitter, which will cause a small speed and acceleration of the head and wrists. This will result in scooting data to be incorrectly classified into rolling.

4.3.3 Performance of The Hierarchical Classifier

a) Threshold-based Method

Table 4.4 shows the four performance metrics for evaluation. In the static posture test, with more than 45,000 sets of data in total, more than 43,000 sets of data were correctly classified (96.04%), and about 1,800 sets of data were incorrectly classified. The threshold-based method has perfect performance in distinguishing lying and not lying. However, large error exists in the test of lying sideways. All categories of facing away from the camera are classified as facing flat. We think it may be caused by the following two reasons: 1) the set threshold cannot clearly distinguish a person facing away from camera, and 2) the camera may have difficulty in distinguishing the front and back of a person. In the dynamic posture test, the classification results are disappointing. Almost three-quarters of scooting data were incorrectly classified as rolling. We consider that it is because the moving speed of different participants was different, and it was difficult to use one threshold to differentiate.

Table 4.4: Performance metrics for all postures (threshold-based)

Class	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1 Score (%)
Standing	98.95	98.36	99.52	98.94
Facing Left	99.67	98.28	99.79	97.91
Facing Right	99.70	98.88	99.77	98.03
Sitting	98.02	96.84	98.80	96.86
Lying Forward	99.96	100.00	99.96	99.10
Lying Backward	97.97	0.00	99.78	0.00
Lying Flat	97.81	100.00	97.75	70.89
Still	95.18	92.34	95.51	79.77
Scooting	63.90	26.96	93.34	39.85
Rolling	62.70	86.93	42.58	67.89
Mean	**91.39**	**79.78**	**92.68**	**74.88**

b) SVM Algorithm

According to Table 4.5 and 4.6, the overall classification by using SVM is much better. In the static posture test, more than 44,000 sets of data were correctly classified (98.30%), and less than 1,000 sets of data (1.69%) were incorrectly classified. The best-performing classifier is fall detection. This outstanding performance proves that our idea of using key joints to recognize fall detection was correct. Only a few standing postures were mistakenly considered as sitting. This part of the data came from only a few volunteers, and the average height of these volunteers is shorter than other volunteers. Therefore, we think this classifier is more friendly to people with larger skeletons, but it needs to be optimized for people with smaller skeletons (e.g. children).

Table 4.5: Confusion matrixes for SVM

		Predicted Class	
	Posture	**Fall**	**Not Fall**
True Class	**Fall**	423	0
	Not Fall	0	381

		Predicted Class	
	Posture	**Not Lying**	**Lying**
True Class	**Not Lying**	35472	16
	Lying	2	3030

Table 4.5, continued

		Predicted Class	
	Posture	**Standing**	**Sitting**
True Class	**Standing**	22274	496
	Sitting	55	12648

		Predicted Class	
	Posture	**Facing Left**	**Facing Right**
True Class	**Facing Left**	3600	57
	Facing Right	28	3454

		Predicted Class		
	Posture	**Lying Forward**	**Lying Backward**	**Lying Flat**
True Class	**Lying Forward**	983	0	3
	Lying Backward	1	823	2
	Lying Flat	51	13	1154

		Predicted Class	
	Posture	**Moving**	**Still**
True Class	**Moving**	1806	129
	Still	2	220

		Predicted Class	
	Posture	**Scooting**	**Rolling**
True Class	**Scooting**	769	59
	Rolling	60	918

In the dynamic posture test, the overall classification performance of SVM is better, but some scooting data is incorrectly classified into the class of sitting still. This part of the data shows that when scooting at a very slow speed, the head speed cannot be used as the only reference feature. In general, SVM algorithm is better than the method of thresholds in classifying multiple postures.

Table 4.6: Performance metrics for all postures (SVM)

Class	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**
Fall	100.00	100.00	100.00	100.00
Standing	98.83	97.88	99.76	98.81
Facing Left	99.75	98.36	99.87	98.43
Facing Right	99.76	99.20	99.81	98.42
Sitting	98.66	99.44	98.53	97.80
Lying Forward	99.88	99.70	99.89	97.28
Lying Backward	99.96	99.52	99.97	98.92
Lying Flat	99.82	94.67	99.96	96.45
Still	88.37	99.10	93.29	76.92
Scooting	88.42	80.36	94.84	86.02
Rolling	94.44	93.77	95.00	93.87
Mean	**97.08**	**96.51**	**98.26**	**94.79**

4.4 Summary

In this work, we optimized and expanded the accuracy and diversity by using Kinect V2 compared to our first work in Chapter 2. By collecting several key joint coordinates of the human body, we have realized the classification and recognition of human body postures by using a hierarchical classifier. By reducing unnecessary features, we used fewer features to achieve high accuracy. For the extracted features, we put them in threshold-based method and SVM algorithm to classify the data respectively. In the tuning of SVM, we tried three kernels and seven different C values. The results show that the RBF kernel is the best, and the value of C parameter does not have a great influence on the accuracy. In the comparison of the two algorithms, the results show that the performance of SVM is better than the performance of the threshold-based method.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

The main focus of this thesis was to study the application of the depth camera Kinect V2 in human activity assessment. In our first work of using the depth camera for posture recognition, we used both thresholds and machine learning methods to perform simpler and more detailed recognition and classification, respectively. By using the thresholds, we achieved basic posture recognition with only one joint (head). The results of tracking head coordinates show that by using Kinect V2, human activities can be well quantified.

In our second work for subject identification, we found that the human joint information recognized by the depth camera can effectively distinguish identities. The distances between joints, in addition to an individual's height, create a unique signature which can be used to successfully identify a subject.

In our third work, we proposed a method of using multiple human joints to recognize postures and compared the performance of SVM algorithm and the threshold method. The results showed that the SVM algorithm has obvious advantages in classifying the postures compared with the threshold method. Overall, we have proved that it is effective to evaluate patients' ambulation by using depth camera.

There are two things that need to be improved in our third work. First, when designing protocols for collecting static posture data, it did not take into account that the two features of speed and acceleration that would affect dynamic posture classification. Therefore, the method of collecting static posture data needs to be optimized. Second, we need to improve the adaptability of the data collecting program to people with different skeleton sizes, so that the classification accuracy of people with smaller skeletons can be improved.

The performance of Kinect V2 is overall excellent in our work. However, as a device released in 2013, its hardware parameters and recognition efficiency are no longer as good as new devices released in recent years. Because the field of view covered by a single camera is limited, it is one of the important tasks in the future to expand the range of posture recognition by combining multiple cameras. Another point of concern is the posture recognition of the specified subject. Although Kinect V2 can recognize multiple human joints, it can hardly recognize the specified subject in the case of multiple people in the field of view.

In a conclusion, depth cameras can be effective in the field of posture recognition and have advantages compared to other methods. It is transparent and more effective in clinical environment where wearing even small devices for a long time is not easy or practical for many patients. The human body joint data collected by the depth cameras can truly reflect the state and activity of the human body in three-dimensional space. The recognition algorithm design and getting rid of the dependence on the monitoring pictures play a role in protecting personal privacy. Therefore, the exploration of this thesis can benefit the healthcare field and patient monitoring.

REFERENCES

[1] S. Ostadabbas, M. Baran Pouyan, M. Nourani and N. Kehtarnavaz, "In-bed posture classification and limb identification," *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, Lausanne, 2014, pp. 133-136.

[2] M. Heydarzadeh, M. Nourani and S. Ostadabbas, "In-bed posture classification using deep autoencoders," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 2016, pp. 3839-3842.

[3] S. Liu and S. Ostadabbas, "A Vision-Based System for In-Bed Posture Tracking," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, 2017, pp. 1373-1382.

[4] K. Kaewplee, N. Khamsemanan and C. Nattee, "A rule-based approach for improving Kinect Skeletal Tracking system with an application on standard Muay Thai maneuvers," *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, Kitakyushu, 2014, pp. 281-285.

[5] J. Zeng, Y. Sun and F. Wang, "A Natural Hand Gesture System for Intelligent Human-Computer Interaction and Medical Assistance," *2012 Third Global Congress on Intelligent Systems*, Wuhan, 2012, pp. 382-385.

[6] Xiaoming Yin, Ming Xie, "Finger identification and hand posture recognition for human–robot interaction, " *Image and Vision Computing*, Volume 25, Issue 8, 2007, Pages 1291-1300, ISSN 0262-8856.

[7] S. Gong, Y. Wang, M. Zhang and C. Wang, "Design of remote elderly health monitoring system based on MEMS sensors," *2017 IEEE International Conference on Information and Automation (ICIA)*, Macau, 2017, pp. 494-498.

[8] D. Brulin, Y. Benezeth and E. Courtial, "Posture Recognition Based on Fuzzy Logic for Home Monitoring of the Elderly," in IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 5, pp. 974-982, Sept. 2012.

[9] B. Schiele, "2007's Wearable Computing Advances," in *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 82-84, Jan.-March 2008.

[10] A. M. Steele, M. Nourani, M. M. Bopp, T. S. Taylor and D. H. Sullivan, "Patient Activity Monitoring Based on Real-Time Location Data," *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, USA, 2019, pp. 1244-1246.

[11] R. Samiei-Zonouz, H. Memarzadeh-Tehran and R. Rahmani, "Smartphone-centric human posture monitoring system," *2014 IEEE Canada International Humanitarian Technology Conference - (IHTC)*, Montreal, QC, 2014, pp. 1-4.

[12] M. I. Idris, A. Zabidi, I. M. Yassin and M. S. A. M. Ali, "Human posture recognition using android smartphone and artificial neural network," *2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC)*, Shah Alam, 2015, pp. 120-124.

[13] M. E. Amine Elforaici, I. Chaaraoui, W. Bouachir, Y. Ouakrim and N. Mezghani, "Posture Recognition Using an RGB-D Camera: Exploring 3D Body Modeling and Deep Learning Approaches," *2018 IEEE Life Sciences Conference (LSC)*, Montreal, QC, 2018, pp. 69-72.

[14] Bernard Boulay, François Bremond, Monique Thonnat, "Human Posture Recognition in Video Sequence," *IEEE International Workshop on VS-PETS, Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003, Nice, France.

[15] Huang JB., Yang MH. (2010) "Estimating Human Pose from Occluded Images. " In: Zha H., Taniguchi R., Maybank S. (eds) Computer Vision – ACCV 2009. ACCV 2009. Lecture Notes in Computer Science, vol 5994. Springer, Berlin, Heidelberg.

[16] C. Hong, J. Yu, D. Tao and M. Wang, "Image-Based Three-Dimensional Human Pose Recovery by Multiview Locality-Sensitive Sparse Retrieval," in *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3742-3751, June 2015.

[17] Kan Chen and Qiong Wang, "Human posture recognition based on skeleton data," *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, Nanjing, 2015, pp. 618-622.

[18] Abbondanza P., Giancola S., Sala R., Tarabini M. "Accuracy of the Microsoft Kinect System in the Identification of the Body Posture, " *Proceedings of the 6th International Conference on Wireless Mobile Communication and Healthcare;* Milan, Italy. 14–16 November 2016; pp. 289–296.

[19] V. Sombandith, A. Walairacht and S. Walairacht, "Recognition of Lao sentence sign language using kinect sensor," *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, 2017, pp. 656-659.

[20] S. R. Tripathy, K. Chakravarty, A. Sinha, D. Chatterjee and S. K. Saha, "Constrained Kalman filter for improving Kinect based measurements," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017, pp. 1-4.

[21] X. Tian, W. Li, Y. Yang, Z. Zhang and X. Wang, "Optimization of Fingerprints Reporting Strategy for WLAN Indoor Localization," in *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 390-403, 1 Feb. 2018.

[22] F. Antoniazzi, G. Paolini, L. Roffia, D. Masotti, A. Costanzo and T. S. Cinotti, "A web of things approach for indoor position monitoring of elderly and impaired people," *2017 21st Conference of Open Innovations Association (FRUCT)*, Helsinki, 2017, pp. 51-56.

[23] Huang, Xiaoping et al. "A Posture Recognition Method Based on Indoor Positioning Technology." *Sensors (Basel, Switzerland)* vol. 19,6 1464. 26 Mar. 2019.

[24] Altman, N. S. "An introduction to kernel and nearest-neighbor nonparametric regression. " The American Statistician. 1992, **46** (3): 175–185.

[25] Hexi Li and Qilin Sun, "The recognition of moving human body posture based on combined neural network," IEEE Conference Anthology, China, 2013, pp. 1-5, doi: 10.1109/ANTHOLOGY.2013.6784858.

[26] Morgan, Willard D.; Lester, Henry M. (October 1954). *Stereo Realist Manual*. and 14 contributors. New York: Morgan & Lester. OCLC 789470.

[27] Z. Xiao, F. Mengyin, Y. Yi and L. Ningyi, "3D Human Postures Recognition Using Kinect," *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Nanchang, Jiangxi, 2012, pp. 344-347.

[28] Miles Hansard, Seungkyu Lee, Ouk Choi, Radu Horaud. "Time of Flight Cameras: Principles, Methods, and Applications." Springer, pp.95, 2012, SpringerBriefs in Computer Science, ISBN 978-1-4471-4658-2. ff10.1007/978-1-4471-4658-2ff. ffhal-00725654

[29] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. "Depth mapping using projected patterns, " 2012.

[30] Microsoft Corp., "Kinect for Windows | Human Interface Guidelines v2.0," 2014, https://download.microsoft.com/download/6/7/6/676611B4-1982-47A4-A42E-4CF84E1095A8/KinectHIG.2.0.pdf

[31] Cai, Z., Han, J., Liu, L. *et al.* "RGB-D datasets using microsoft kinect or similar sensors: a survey." *Multimed Tools Appl* **76,** 4313–4355 (2017).

[32] T. Visutarrom, P. Mongkolnam and J. H. Chan, "Postural classification using Kinect," *2014 International Computer Science and Engineering Conference (ICSEC)*, Khon Kaen, 2014, pp. 403-408.

[33] T. Le, M. Nguyen and T. Nguyen, "Human posture recognition using human skeleton provided by Kinect," *2013 International Conference on Computing, Management and Telecommunications (ComManTel)*, Ho Chi Minh City, 2013, pp. 340-345.

[34] E. Lachat, H. Macher, T. Landes and P. Grussenmeyer, “Assessment and calibration of a RGB-D camera (Kinect v2 sensor) towards a potential use for close-range 3D modeling,” *Remote Sensing*, vol. 7, pp. 13070-13097, 1, Oct. 2015.

[35] Z. Zhang, Y. Liu, A. Li and M. Wang, "A novel method for user-defined human posture recognition using Kinect," *2014 7th International Congress on Image and Signal Processing*, Dalian, 2014, pp. 736-740.

[36] S. Gaglio, G. L. Re and M. Morana, "Human Activity Recognition Process Using 3-D Posture Data," in *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 586-597, Oct. 2015.

[37] A. L. Parise, “Human identification by walk recognition using relevance vector machine,” *2017 International Conference on IoT and Application (ICIOT)*, Nagapattinam, 2017, pp. 1-4.

[38] G. Koukiou, G. Panagopolulos and V. Anastassopoulos, “Drunk person identification using thermal infrared images,” *2009 16th International Conference on Digital Signal Processing*, Santorini-Hellas, 2009, pp. 1-4.

[39] A. K. Jain, A. Ross and S. Prabhakar, "An introduction to biometric recognition," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, Jan. 2004.

[40] G. De Vico Fallani, G. Vecchiato, J. Toppi, L. Astolfi and G. Babiloni, “Subject identification through standard EEG signals during resting states,” *33rd Annual International Conference of the IEEE EMBS*, Boston, Massachusetts, USA, Aug. 30 – Sept. 3, 2011.

[41] J. Birjandtalab, M. B. Pouyan and M. Nourani, “An unsupervised subject identification technique using EEG signals,” *2016 38th Annual Conference of the IEEE Engineering in Medicine and Biological Society (EMBC)*, Orlando, FL, 2016, pp. 816-819.

[42] A. Rahman, V. M. Lubecke, O. B. Lubecke, J. H. Prins and T. Sakamoto, “Doppler radar techniques for accurate respiration characterization and subject identification,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 8, No. 2, June 2018.

[43] V. O. Andersson and R. M. Araujo, “Full Body Person Identification Using the Kinect Sensor,” *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, Limassol, 2014, pp. 627-633.

[44] P. Das, A. K. Sadhu, A. Konar, A. Lekova and A. K. Nagar, “Type 2 fuzzy induced identification using Kinect sensor,” *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Istanbul, 2015, pp. 1-8.

[45] A. Sinha and K. Chakravarty, “Pose Based Person Identification Using Kinect,” *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, 2013, pp. 497-503.

[46] M. G. Bhargavas, K. Harshavardhan, G. C. Mohan, A. N. Sharma and C. Prathap, “Human identification using gait recognition,” *International Conference on Communication and Signal Processing*, Apr. 6-8, 2017, India, pp. 1510-1513.

[47] T. Mizumoto, A. Fornaser, H. Suwa, K. Yasumoto and M. De Cecco, “Kinect-based micro-behavior sensing system for learning the smart assistance with human subjects inside their homes,” *2018 Workshop on Metrology for Industry 4.0 and IoT*, Brescia, 2018, pp. 1-6.

[48] L. Pashikani and D. Von Ah, "Impact of early mobilization protocol on the medical-surgical inpatient population: an integrated review of literature", *Clinical Nurse Specilist*, vol. 26, pp. 87-94, April 2012.

[49] Fox R.L. & Abrahamson K. (2009) A critical examination of the U.S. nursing shortages: contributing factors, public policy implications. *Nursing Forum* **44** (4), 235–244.

[50] Haddad LM, Annamaraju P, Toney-Butler TJ. Nursing Shortage. [Updated 2020 Jul 25]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2020 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK493175/

[51] O. P. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition—A Review," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865-878, Nov. 2012.

[52] H. Foroughi, B. S. Aski and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," *2008 11th International Conference on Computer and Information Technology*, Khulna, 2008, pp. 219-224.

[53] A. H. Nasution and S. Emmanuel, "Intelligent Video Surveillance for Monitoring Elderly in Home Environments," *2007 IEEE 9th Workshop on Multimedia Signal Processing*, Crete, 2007, pp. 203-206.

[54] D. Xu, X. Xiao, X. Wang and J. Wang, "Human action recognition based on Kinect and PSO-SVM by representing 3D skeletons as points in lie group," *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, 2016, pp. 568-573.

[55] M. D. Štrbac and D. B. Popović, "Computer vision with Microsoft Kinect for control of functional electrical stimulation: ANN classification of the grasping intentions," *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, Belgrade, 2014, pp. 153-156.

[56] A. Nandy and P. Chakraborty, "A new paradigm of human gait analysis with Kinect," *2015 Eighth International Conference on Contemporary Computing (IC3)*, Noida, 2015, pp. 443-448.

[57] A. Amini, K. Banitsas and J. Cosmas, "A comparison between heuristic and machine learning techniques in fall detection using Kinect v2," *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Benevento, 2016, pp. 1-6.

[58] J. Barabas, T. Bednar and M. Vychlopen, "Kinect-Based Platform for Movement Monitoring and Fall-Detection of Elderly People," *2019 12th International Conference on Measurement*, Smolenice, Slovakia, 2019, pp. 199-202.

[59] J. Shotton *et al*., "Real-time human pose recognition in parts from single depth images," *CVPR 2011*, Providence, RI, 2011, pp. 1297-1304.

[60] M. Hansard, S. Lee, O. Choi and R. Horaud, “Time of flight cameras: principals, methods, and applications,” Springer, SpringerBriefs in Computer Science, 2012, pp. 95

[61] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.

[62] Bernhard Schölkopf, Koji Tsuda, Jean-Philippe Vert, "A Primer on Kernel Methods", *Kernel Methods in Computational Biology*, 2004.

[63] Keerthi SS, Lin CJ. "Asymptotic behaviors of support vector machines with Gaussian kernel, " *Neural Comput.* 2003 Jul;15(7):1667-89.

[64] RBF SVM parameters, https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

BIOGRAPHICAL SKETCH

Zihang You was born in Shifang, Sichuan Province, China. He completed his Bachelor of Engineering at University of Electronics Science and Technology of China in 2018. He also completed his Bachelor of Science, Honor of the Second Class (Division I), from the University of Glasgow in 2018. During his undergraduate studies, he was awarded the Renmin Scholarship and UESTC Special Scholarship for several times. He also won the Excellent Graduation Thesis of Glasgow College in 2018. He was admitted to the master's program in Electrical Engineering at The University of Texas at Dallas in August 2018. He joined the Predictive Analysis and Technologies Laboratory in 2019. He was awarded the Master's Summer Research Fellowship Program in 2020.

CURRICULUM VITAE

Zihang You

Department of Electrical and Computer Engineering, The University of Texas at Dallas
800 W. Campbell Rd., Richardson, TX 75080-3021, USA
Email: zihang.you@utdallas.edu | Phone: (469) 468-1930

EDUCATION

THE UNIVERSITY OF TEXAS AT DALLAS (UT Dallas)

Master of Science in Electrical Engineering, expected in Dec. 2020

Major: Electrical Engineering

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA(UESTC)

UNIVERSITY OF GLASGOW, Glasgow, UK (Joint Program with UESTC)

Bachelor of Science, Honors of the Second Class (Division I)

Major: Electronics and Electrical Engineering

PUBLICATIONS

Alec M. Steele[1], Zihang You[1], Mehrdad Nourani[1], Melinda M. Bopp[2], Tanya S. Taylor[2], Dennis H. Sullivan[2], "**Subject Identification Using a Depth Camera for Patient Ambulation Monitoring**", 2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Montreal, Quebec, Canada, 2020, pp. 5745-5748

Zihang You, Alec M. Steele, Mehrdad Nourani, Melinda M. Bopp, Tanya S. Taylor, Dennis H. Sullivan, "**Quantifying Human Activity Using Head Tracking**", 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 2019, pp. 1247-1249

Zihang You[1], Adnan Zahid[2], Hadi Heidari[2], Muhammad Imran[2], Qammer H. Abbasi[2], **"A Compact Wearable System for Detection of Plantar Pressure for Diabetic Foot Prevention,"** 2018 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Chengdu, Sichuan, China, 2018, pp. 64-67

HONORS

Master's Summer Research Fellowship Program, 2020

Excellent Graduation Thesis of Glasgow College, 20/229, 2018 UESTC

COMPUTER SKILLS

Programming Languages: C/C++, C#, Python, MATLAB, VHDL, Verilog

Software: Microsoft Visual Studio, MATLAB, PyCharm, Cadence, Synopsys, Multisim

RESEARCH INTERESTS

Embedded System

Computer Vision

Wearable Devices

Machine Learning