MICROSOFT KINECT BASED REAL-TIME SEGMENTATION AND RECOGNITION FOR HUMAN ACTIVITY LEARNING

by

Sanobar Kadiwal, B.E.

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE,

in Computer Engineering

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

DECEMBER, 2017

MICROSOFT KINECT BASED REAL-TIME SEGMENTATION AND RECOGNITION FOR HUMAN ACTIVITY LEARNING

by

Sanobar Kadiwal

APPROVED BY

_____

Jiang Lu, Ph.D., Committee Chair

_____

Hakduran Koc, Ph.D., Committee Member

_____

Kewei Sha, Ph.D., Committee Member

APPROVED/RECEIVED BY THE COLLEGE OF SCIENCE AND ENGINEERING

_____

Said Bettayeb, Ph.D., Interim Associate Dean

_____

Ju H. Kim, Ph.D., Interim Dean

Dedication

For my parents.

ACKNOWLEDGEMENTS

Initially, I would like to thank all those who supported me indirectly or directly and those who have encouraged and guided me in all the ways throughout my masters.

In addition, I would like to express my deep gratitude to all my committee members for helping me out in this thesis. My sincerest gratitude to my mentor and committee chair Dr. Jiang Lu, who had offered me assistance during my research and who had accepted my thesis research and approach and encouraged me to achieve this goal. The motivation and guidance he has given me made me a better student and raised my knowledge on the machine learning and the implementation on this research. I would also like to thank Dr. Hakduran Koc and Dr. Kewei Sha for being my committee member. I would like to express my special thanks to Dr. Kewei Sha for sharing his suggestions and ideas as further approaches in every step.

I would like to especially thank my parents and my siblings for believing in me and motivate me to earn this degree. I would like to thank them for helping me build my hope and for their financial and emotional support.

Lastly, I would like to thank my friends Nitish Kelagote, Dhruvin Gajjar and Alif Momin for sharing their ideas for improvement of this research and for their constant support and motivation to achieve this goal.

ABSTRACT

MICROSOFT KINECT BASED REAL-TIME SEGMENTATION AND RECOGNITION FOR HUMAN ACTIVITY LEARNING

Sanobar Kadiwal, M.S.

The University of Houston-Clear Lake, 2017

Thesis Chair: Jiang Lu, Ph.D.

Lower body pain and injury have become common in this technical world, especially in elderly people. It is quite difficult to recover from these injuries leading to problems in performing daily routine activities like walking, running, sitting etc. Although there are many activity recognition models present today, there has been relatively little multiple activities recognition study of lower limbs. Most of the previous researchers focused on single activity recognition using various machine learning algorithms. Researchers have evolved with the learning of gait using different methods and techniques for upper and lower body using the sensors and different camera systems. This research has two main sections, one is for segmenting the motion and another is recognizing those movements. In this research, multiple activities were performed by the patients in a random manner without stopping and these activities were recognized in different groups stating the performed activity if the part of the multiple activities is walking or running or leg raising activity. The first goal of this dissertation is to plot the human gaits as a skeleton using MATLAB with a camera sensor second goal is to segment those derived gaits using the on-line

aligned cluster analysis and dynamic time alignment kernel method and the last goal is to recognize the segmented gaits using the support vector machine algorithm. This is done by tracking and learning the person's lower limb data points and finding the exact action performed by a dynamic time alignment Kernel method for segmentation and comparison of different algorithms like Support Vector Machine, K-Nearest neighbors for recognition. The experimental results collected in this research show that the Support Vector Machine performs higher recognition accuracy.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1:
INTRODUCTION

1.1 Background

Technology nowadays has become more and more advanced and efficient in terms of improving quality of life of the people. They have become useful in many aspects to coordinates meeting and appointments and for online bank payments or bill payments from home, checking weathers, live updates can be seen with just a click. In health care, doctors and patients communication can be seen in online and live video communication applications without going to hospitals. One such application known as telehealth is widely accepted, where it actually collects all the data from the body (physiological signals) or body motion to improve health care system and to educate (i.e. provide medical-related suggestions and evaluations) the people connected directly or indirectly to health systems using the telecommunication technology. It delivers the quality education to the public via various technologies since the patients and doctors are at distant places, where it helps in seeking advice, educating and treating the patient from far away distance. This can be done by using various types of sensors, camera systems or live videos and mobile health to assist in different applications.

Human-computer interaction (HCI) is in wide demand as it studies the human activities using the computer technology to enhance the human interaction while processing the human to machine and machine to human interface to complete a particular task. HCI depends on the human senses example motion, vision, touch, sensor etc.

The various examples of human-computer interaction are the technological games, humanoid robot, and biometrics. It has some applications in the field of E-

learning i.e. learning online and gaining academic knowledge. It is also useful for healthcare and medical issues i.e. learning the information of the patients from the derived result. Nowadays, telehealth care took a lot of attention of most of the researchers with various applications previously using telephone and facsimile machines and now wirelessly using 4G technologies and using cellphones. It can also use different machine learning algorithm to acknowledge advancement of the scheme design for human-computer interaction program. While conventional method requires the patients to go rehabilitation centers for physical training. Expert therapists or attendants need to be around the patients to guide them and also to note and correct the progress of the patient, but there are insufficient human healthcare professionals to manage the completely elderly associate in the population so telehealth care is considered to be significant. Its function is to monitor the patient, store its data and to communicate with other technology in order to deliver healthcare information from distance using sensors and/or cameras along with some wireless technologies thus helps to improve the quality of life and public health using the newfangled technologies virtually. Telehealthcare combines the information communication technology and the biometric sensor gadgets to address disease management and facilitates longitudinal health monitoring status, which will be very helpful in improving the telemedicine services and can be used to develop the therapies and treatments. It helps the patient with the rehabilitation. It can also help monitor the patient at every single stage while allowing long-distance clinician to contact and to take care of the patient.

Human-Computer Interaction is the study of human activities using the computer technology to enhance the human interaction with the latest technologies. The two main tasks are human activity capture and recognition.

In this research, visual-based HCI is studied where the surrounding nature can be captured using different cameras. This research shows the intuitive and low-cost

system for tracking and recognizing of different human activities. Researchers in the past have studied about the usage of the Microsoft Kinect in the telehealth field. However, in recent years, it has become important to learn about the segmentation of different activities and labeling those activities real-time in computer vision applications. This study aims at improving the technology to assist the patients to boost their potential movement.

Our system uses low-cost Microsoft Kinect camera sensors and personal computers to gather the lower-body movement data and provides analysis of the performed activity. Here, we are using the Kinect RGB –D sensors to track the human body to study the lower limb movements. In this thesis, we will first track the lower limb followed by the estimation of skeleton formation and detecting the skeleton. These skeletal movements are video tracked based on temporal segmentation and the actions and gestures are performed which is then segmented while recognizing the activity posture based on movements.

1.2 Challenges

Recognizing and analyzing daily routine human activities in an intelligent and low-cost manner (*e.g.*, vision-based) for elderly as well as injured people has become significant as more and more people are starting to live alone and to provide them with appropriate health and medical services is a major task. Video-based human activity recognition has been an active research topic in computer vision over the last few years [1]. However, the fundamental limitation of the sensing device (*i.e.*, color camera) restricts previous techniques to only help in detecting lateral motions with 2D images. The human bodies can be modeled as a three-dimensional data set. In the 3-D viewing, the information loss in the depth channel could cause serious deterioration of the data representation. As a result, the feature representation of the different activities

turns into a difficult task. However, the newer researches are based on Microsoft Kinect's depth data and RGB data which are beneficial for capturing the real-time motion data especially the depth data along with color images having the depth accuracy for approximately one centimeter. These data produced will be three-dimensional motion data of the person. In such a way, the three-dimensional data of the color images with depth information can help to build a much more accurate human motion model.

The human activity data is captured by Microsoft Kinect which produces the color image and depth image. Using the output from Kinect and with the basic light brightness in the surrounding, the data from the surrounding is considered as a noise and is removed. Moreover, the 3D human skeleton model is built from the output of Kinect. With the help of the 3D skeleton model, we can track (segment) different activities and recognize them.

However, the challenges of the research are:

1. Sensing and detecting the body movement is the crucial. As using the Kinect hardware for motion is an easy task but, to extract the important features and form the 3D skeleton model from Kinect is quite challenging.
2. Several types of activities may be performed in one individual movement. It is hard to segment different activities in a single phase, especially, real-time segmentation. Real-time segmentation helps to locate the two unique and dissimilar activities. It is quite difficult to allocate different activities. Many offline algorithms are already in research [2]. However, the algorithm used online is a challenge to learn.
3. To recognize different activities is still a challenge. Real-time recognition is important and practical in real applications. Many recognition algorithms is studied such as HMM (hidden Markov Model) which are used for clustering, but the accuracy of few algorithms might be

unacceptable for recognizing the gestures. So, the Support Vector Machine algorithm will be used to get the better accuracy.

1.3 Related Work

An image from the depth sensors are captured for various applications but for capturing two dimensional data along with another inertial sensors and thus recognizing the human action [3] [4]. Another application is learning and analyzing the human gait using Pyroelectric infrared (PIR) sensors, thus, estimating the pace of the human activity and learning gesture on treadmill [5] [6]. Removing the background as the noise followed by feature extraction and learning the joint angle as feature [7] [8]. An application of statistical pattern recognition technique to the classification of activity is described in the paper [9]where pattern recognition techniques extended to identify altered techniques. In this paper feature vectors were extracted from the activities recorded based on average duration, number and intensities as well as frequency and propagation characteristics Pattern training and classification were performed by the machine learner technique, Bayes decision rule. In other experiment, [10] Reflective pads are attached to the joints of a person and his or her movements are filmed against a black background to capture grey level images, based on spatial and temporal uniformity, to recognize the sequential presentation of people's walking, running or jumping etc. image's frequency domain analysis where low-level motion information isolates and track likely locations of activity and low-level structural features are used to classify the detected activities. Some research paper uses multiple sensors to recognize the pattern and activities using a network of connected objects like iPhone, an Apple Watch and an Apple TV remote containing an accelerometer, gyroscope, and microphonic data and the data is recognized using Deep Neural network [11]. Another neural network approached experiment is

performed, which design a recognition model of forearm movements using a single wrist-worn accelerometer sensor.

Nowadays, study of activity recognition has become a significant part to improve quality of life, some of the techniques uses Microsoft Kinect RGB Depth sensors, which provides high information, included in [12] provides real-time processing with the less consumption of power and by combining three different machine learning techniques i.e. K-means clustering, support vector machine and Hidden Markov model detects the 3D posture via assumption of skeleton. The authors used the Kinect activity recognition dataset and CAD 60 to perform the experiments. Another proposed method [11], uses the deep convolutional neural networks (ConvNet) and weighted hierarchical depth motion maps (WHDMM) to recognize the activity from depth sensor and encodes the spatiotemporal pattern to spatial motion structure in 2D with the minimum number of training samples and the action performed are captured and converted to image classification (pseudo-color images), also viewing the maximum angles viewpoint. With enough number of samples and spatially and temporally action localizing vision based method is used in [13] and finding the most robust method for classification and action recognition considering few challenges such as view invariance and occlusion. Another offline classifier method for the hand posture is proposed with the robustness of touchless display and the multiple sensors to improve the accuracy from all the viewpoints and improving the pose estimation accuracy with the maximum of 120 pose estimation per second [14] while reducing the pose estimation error. Using computer vision technique hand movements are tracked in mid-air with the help of dual sensors, leap motion sensor, to combat the occlusion issues.

Some techniques based on various types of wearable sensors are approached; one of these is mentioned in the [15], where a natural, practical, comfortable and

accurate technique is proposed to capture gestures and sensing its activity using the accelerometer and surface electromyography.

The gestures are captured and sensed using these sensors through the vibration and gravity and interacted with the mobile system. The recognized gesture must be segmented and identified if there are multiple actions performed in single movement i.e. the different motion are set apart through segmentation. Jonathan Feng-Shun Lin, Michelle Karg, and Dana Kuli´c in [16] proposed online, offline and semi-online method, based on template training and observation segmentation, to segment the motion primitives. Farzad Siyahjani and Saeid Motiian deal with high dimensionality of captured data [17] by reproducing kernel Hilbert spaces and mapping data into these space and state space models resulting in accurate estimation analysis, recognition, and detection. Another paper by Jonathan Feng-Shun Lin [2] proposes an online segmentation and identification of movement segments from continuous time-series data, based on velocity features and probabilistic modeling of every single motion while handling high dimensional data. Sometimes already temporally (spatially) segmented data can cause some inaccuracy to solve this, Imran Mumtaz proposed another online segmentation and classification in [18] which simultaneously performs temporal segmentation as well as classification with the online incoming stream of human motion.

Although lots of research has been done for segmentation and recognition of human activities, there are still couples of limitations in this research area:

1. Offline. The above-mentioned papers on Kinect based system has several limitations as the real-time segmentation is possible with the fact that they can only be segmented as an offline method. Whereas, the necessity of the online-based method is expanding.
2. Accuracy. Some of the approaches have upgraded the modern technology with the increasing accuracy rate and with the two-dimensional view. But,

still, the accuracy rate must be increased to some acceptable level along with high dimension for improvement of telehealth vision i.e. 3-dimensional views, which makes it easy to study the features of the human body. Accuracy is also affected by the vision, which is impacted by background and lamination. As some of the telehealth technology uses the wearable sensors, which makes few patients uncomfortable to wear and some may not be willing to wear the sensors, which will be another issue with the sensor usage.

1.4 Proposed Work

Two main tasks in thesis: (a) online segmentation; (b) real-time recognition.

a. The online segmentation includes the use of K-mean kernel algorithm and DTAK (dynamic time alignment kernel) [19] along with the alignment cluster analysis to cluster a single data of multiple activities into multiple segments. This algorithm is online. The dynamic time alignment kernel algorithm along with the alignment cluster analysis will make the algorithm online giving the advantage of segmenting the data online. The data is segmented into parts and each segmented part is moved forward to check and backward for error correction giving the maximum accuracy. Using the algorithm, the forward and backward steps are performed multiple times, leading to segmented data, giving more accuracy [20].
b. The segmented data are recognized correctly and accurately. The Support Vector Machine algorithm used will train the data. After training a feature set will be stored and used for testing. The training phase is done offline and the testing phase of SVM is achieved real-time. As soon as one

segmented activity is detected, it will pass to real-time SVM recognition to find which activity it is.

1.5 Organization

The thesis is organized as following structure. In Chapter 2, we will talk about the hardware components used and system setup of the experiment is defined along with the system diagram. Our experiment is divided into two main categories online recognition, which leverages the kernel algorithm and dynamic time alignment kernel. It will be explained in Chapter 3 and Real-time recognition including the support Vector machine and its algorithm with the architecture will be explained in Chapter 4 in details. Chapter 5 provides the experimental results and dissection. The conclusion and future work are shown in chapter 6.

CHAPTER 2:

SYSTEM SETUP

2.1 Hardware Components

Activity is recognized based on the information provided by the Kinect. Kinect (Figure 2.1) is the Microsoft device, used mostly for gaming purpose, [21] allowing the user to interact with the device without any mediator. However, along with the gaming application, researchers planned to use Kinect for multiple useful purposes such as manipulating medical images through gestures. It has various medical applications such as it produces the high-quality 3D [22] scans, which can helps to recover the stroke patients at low cost as well as real-time sign language translation. It translated the sign language [23], written or spoken the language by reading the gestures of the person and interpreting it. Utilizing body gestures, Kinect can break into a PC's security framework. Using Kinect, the normal screen surface can be turned to the interactive touchscreen. This research uses Kinect to detect the human body and its gestures and based on the gestures and activities, the Kinect will segment the different activities at one time and recognize the various activities. Microsoft Kinect is used to capture the information using both the cameras in the system i.e. the Infrared (IR) depth sensor and the RGB camera. The capability of Kinect's video recording and image capture is image and video is captured as a color image using the RGB camera with the resolution of 1280 x 1024 [24]. The depth map image and video uses the color white to far blue and the frame rate with which the output video is captured depending on resolution is approximately 9Hz to 30 Hz. The sensor has a higher range of accuracy, usually 4 mm depending on the depth distance. Its angular field of view is 57˚ horizontally and 43˚ vertically, while the camera sensor can be tilted upwards or downwards up to 27˚.The Kinect's ranging limit

distance is 1.2 m to 3.5m practically i.e. 3.9ft to 11.5 ft. It contains the Infrared emitter, which emits the infrared light beam and the reflected IR beam is read by the infrared depth sensor in terms of depth information which will be helpful in finding the distance between the sensor and the object, leading to the depth image as well as extracting high level of information from the data captured using sensors. The other camera system can view the body skeleton but the video processing requires many resources and has the disadvantage of background obstacle, which makes the Kinect a powerful tool for the general camera. Installing Kinect requires windows operating system with the media feature pack for Kinect for Windows Runtime. The computer requires 32-bit (x86) or 64-bit (x64) processors, Dual-core, 2.66-GHz or faster processor, USB 2.0 bus dedicated to the Kinect, 2 GB of RAM, Graphics card that supports DirectX 9.0c. The computer also requires other software to install i.e. the Microsoft visual studio and .Net framework. This research uses Kinect to capture the patient's motion especially the lower limb body part to recognize his ability to move the legs accurately.

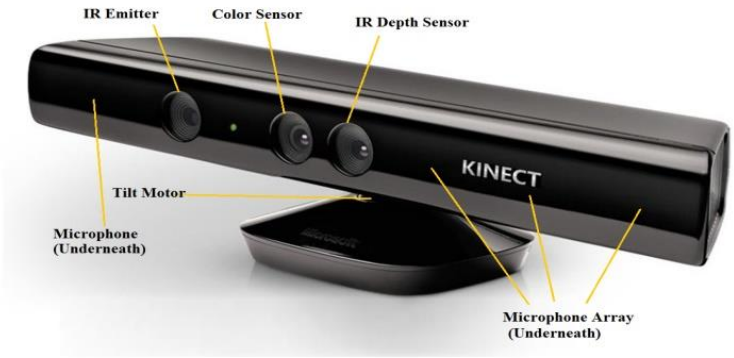
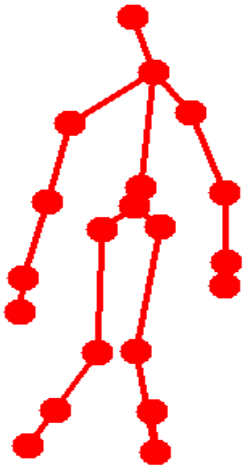


Figure 2.1: Kinect Image

The experimental setup includes MATLAB, as it is preferable to work on the processing files. The experimental process needs the Kinect connected to the computer with the camera sensors facing towards the person. The person from

approximate 1.2 m to 3.5m from the Kinect will perform some random gestural movement, which is then captured by Kinect camera sensor as a video. The image from Kinect is extracted to MATLAB, which uses the data from Kinect as the input (figure 2.2a).

(a) Depth Image



(b) Skeletal Image

Figure 2.2: Images from Kinect shown in MATLAB (a) Depth Image; (b) Skeletal Image

This research focuses on the lower limb part of the body to segment and recognizes the similar movements online. The knee movements are recognized for some temporal moment. This is done considering only a few skeletal points i.e. from

hipbone to toe. Each leg sample in skeletal view consists of four points as shown in figure 2.2b. For the particular amount of time, the skeletal video-like sequence is captured by Kinect and fed to MATLAB, which in turn captures the gesture motion for that specific time. There includes types of motion including walk, run, squat, front leg raise, etc. are required to analyze the gait balance. The result of the MATLAB will show the accurate gesture lower limb recognition.

2.2 Software Components

The generated image from the software will have several motions. These motions will be continuous and random in nature, as the patients are allowed to perform certain motion exercises to improve the balance and mobility. Considering three stages in patients' condition early stage, mid and late stage, patients are asked to perform several minute exercises. For example, the patient is allowed to do few minute exercises like standing, walking at an early stage, followed by mid-stage performing 15 to 20 minutes fast walk and running with the increment in a number of steps they are performing. The Late stage might include performing the squat exercise, jumping and few harder steps with an increase in time and steps. Motion segmentation is a vital stage to obtain the slightest details. Many algorithms have been developed to estimate the gait parameter and segmenting the gait cycle [25]. The speed, with which person is performing an action, will not be the factor of concern as we can use the zero velocity crossing feature, which recognizes the time series features and also frequency filter for different velocity motions. In addition to this, dynamic time warping is the method, which measures the similarities between two-time series data taking into account its amplitude, thus matching the points according to both the amplitudes. Dynamic time alignment kernel (DTAK) is a kernel-based method, which uses the unsupervised learning technique to segment the data points.

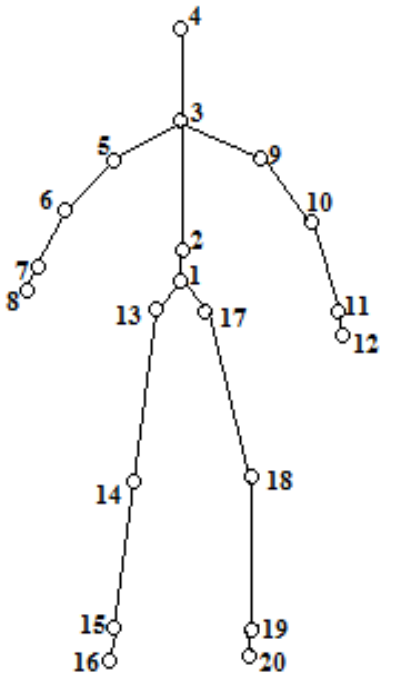
The skeletal position is as shown in table 2.1. In this research, the main consideration is lower limb which is joint points from 13 to 20 for both the gait movement. The joint being represented by

$$P = [pl, pr] \epsilon \Re^{m*2} \tag{2.1}$$

where pl and pr are joint position for left and right leg respectively. $pl = [q1, q2, \dots qm]_l \in \Re^{3m*2}$, where $m = 4$ and $[qi]_l = [xi, yi, zi]_l$ is the coordinate of the i[th] marker of left leg.

Table 2.1: Skeletal Position Connection Map

Joints	Number	Joints	Number
Hip center	1	Wrist right	11
Spine	2	Hand right	12
Shoulder center	3	Hip left	13
Head	4	Knee left	14
Shoulder left	5	Ankle left	15
Elbow left	6	Foot left	16
Wrist left	7	Hip right	17
Hand left	8	Knee right	18
Shoulder right	9	Ankle right	19
Elbow right	10	Foot right	20



The human body is represented as a skeleton, eliminating the surrounding environment, thus having an advantage of being least affected by the neighboring environment and low-light condition, as a set of kinematic joints using the Microsoft Kinect. Here, for the lower limb body, the skeleton only utilizes from the hip bone to the foot for both the limb movement. The movement of the body is calculated in the spatial model.

Multiple kinds of movements will be performed by the subject and the performed movement will be segmented and the similar movements are recognized. Example the subjects are performing the following actions of running, walking, jumping all at the same moment, this movement will be recognized as the unique from each other and will be segmented depending on which activity is performed followed by some other activity, here the activities performed will be varying in time, so temporal clustering will be done to segment the information signal temporally.

2.3 Feature Extraction

During the rehabilitation process, motion sequence is generated by the depth sensor contains multiple actions in one-time frame where each action lasts for a certain amount of time. This method uses the feature, which are extracted from the depth sensors data. There are many features used such as mean, standard deviation, entropy, variance, maximum value, mean absolute deviation and number of the mean crossing. So the total extracted features from the experiment will be 168 (2leg×4joints×3 dimensions×7 features).

2.4 System Diagram

The person will perform some activity in front of Kinect. This person may walk for 10 times, run for 15 times or squat 3 times for few seconds to few minutes. He may repeat the activities. This activity will be captured as a video, snapping continuously varying images which will be connected to each other spatially. This image will be in form of skeleton. The lower limb points are of most importance to study the movement. The online segmentation part will segment the similar activities separating the dissimilar motions. As seen in Figure 2.3, the green, red and blue

represent the three different of activities performed multiple times. This activity is still unknown if its walk, run or any other movement, which will be recognized in the activity recognition block.

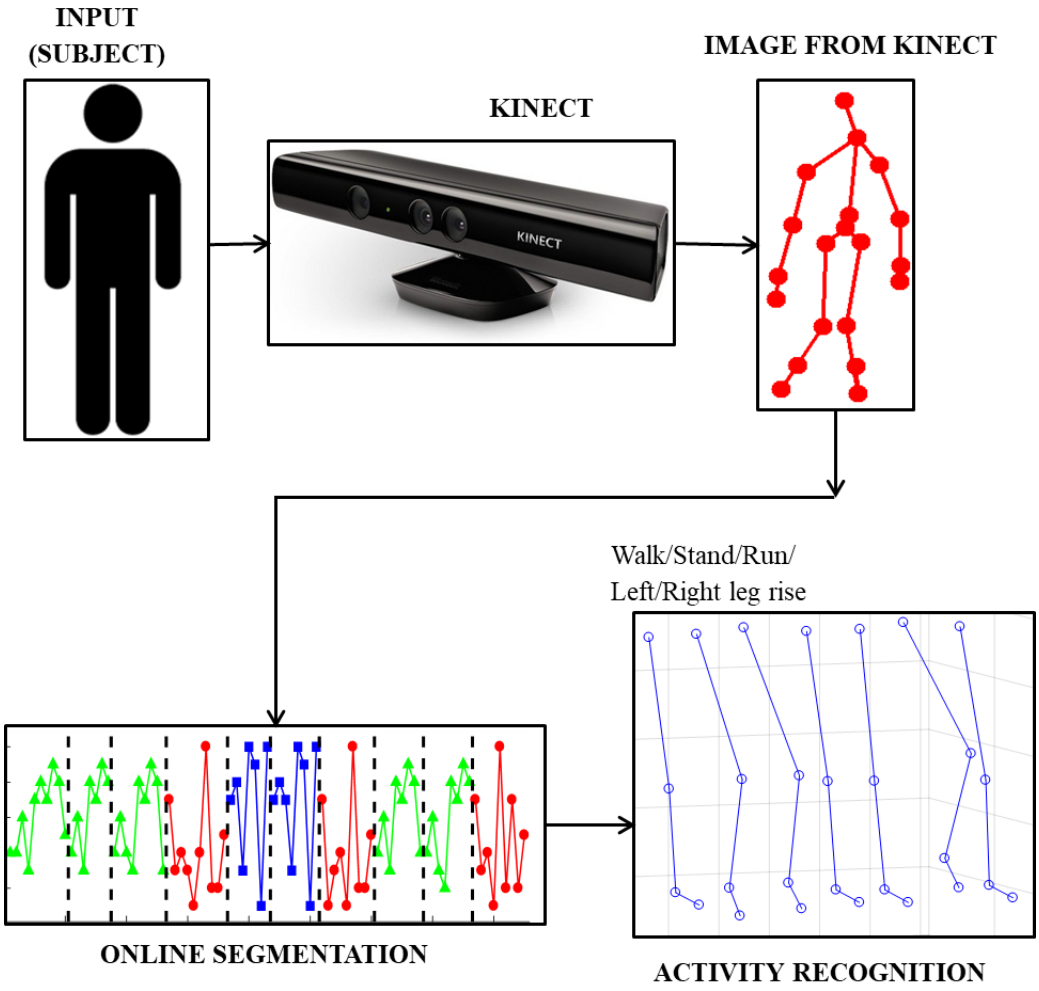


Figure 2.3: System Diagram

CHAPTER 3:
SEGMENTATION

3.1 Definition

The procedure of rehabilitation includes many physical exercises, but to detect the gestures and to segment, the motions have become an important factor to notice the improvement of the rehabilitation. Segmenting the data defines subdivision of the motion into accurate boundaries of the action performed. Segmentation of human motion sequence temporally will cut sequences into segments with most variation i.e. finding the starting and ending location of each motion [26]. Temporal segmentation is divided into two categories, offline or online. The temporal clustering of the human motion is performed offline generally, making it difficult to segment the actions in real-time application. The online segmentation detects and segments the motions, which depends on the time series data [27]. This online approach deals with the high dimensional data as well as the variation complexity. The online processing is necessary to provide immediate feedback to the therapist and the patient for the application in rehabilitation. Most of the gesture recognition requires the offline programming or offline training datasets to model the gait performance. These systems are not suitable for the gesture recognition, where the gestures are needed to be learned. It has a major disadvantage to waiting for the offline training data sets and programming stage. Apart from the above drawback, it also needs to be learned new datasets whenever new data points are collected and with a large number of training sets. It is also bounded in one dimension of the data, whereas the information of the human gait is usually high dimensional. So, the online system is used to segment the data points without having to learn a large amount of examples. This simplifies the overall process, as it resembles the instructions used for training people.

3.2 Methodology

The online method used in this system is the kernel based method which transforms the high dimensional raw represented data into the feature vector representation for temporal clustering. There are already some applications on the semi-supervised temporal clustering and segmentation based on velocity and acceleration estimation is also obtained in previous researches. This work deals with the aligned cluster analysis (ACA) and hierarchical ACA (HACA) and k-means algorithm.

Kernel method is the new class of pattern analysis algorithm. It integrates and merges the different type of data. It processes the dataset to kernel matrix and analyzes the data. Kernel algorithm includes Gaussian process, Fisher kernel, polynomial kernel, and Kernel k-mean [28]. The most common kernel algorithms used are linear kernel and Gaussian kernel. Here, we are using the exponential kernel method to form binary frame matrix [20] [25] [29].

3.2.1. Kernel k- means Algorithm

Standard k-means clustering algorithm has an expansion called as kernel k-mean algorithm that identifies nonlinearly separable clusters. Kernel k-means clustering is used to minimize the within-cluster variation by clustering the set of n samples into k disjointed cluster groups. It finds the partition of the data of the energy function given below:

$J_{km}(Z,G) = \sum_{c=1}^{k}\sum_{i=1}^{n} g_{ci}||x_i - z_c||^2 = ||X - ZG||_F^2 \qquad \text{s.t} \qquad G^T 1_k = 1_n \quad (3.1)$

when $x_i \in IR^d$ is vector a for i_{th} data point. Geometric centroid of class c is represented as $z_c \in IR^d$. If the sample x_i is part of cluster c then the binary indicator matrix $G \in \{0,1\}^{k*n}$ has$g_{ci} = 1$, else its zero. By minimizing the energy function, for each data points, initially calculating $g_i \in \{0,1\}^k$ to make one of the row as one and

others as zero. Step two includes calculation of the mean of each cluster by computing $Z=XG^T(GG^T)^{-1}$.

$$J_{km}(Z,G) = \sum_{c=1}^{k}\sum_{i=1}^{n} g_{ci}\left||\emptyset(x_i) - z_c|\right|^2 = \left||\emptyset(X) - ZG|\right|_F^2 \quad (3.2)$$

The squared distance between the centroid of class c and i_{th} sample is given by

$$dist_{\emptyset}^2(x_i, z_c) = k_{ii} - \frac{2}{n_c}\sum_{j=1}^{n} g_{cj}k_{ij} + \frac{1}{n_c^2}\sum_{j1,j2=1}^{n} g_{cj1}g_{cj2}k_{j1j2}, \quad (3.3)$$

where $n_c$is number of class c samples and $k_{ij} = \emptyset(x_i)^T\emptyset(x_j)$ is the Kernel function. Here, K-mean clustering has the drawback that it doesn't consider the temporal frame ordering. So the ACA is used combining the k-mean with the spectral clustering [20] [30].

3.2.2. Frame Kernel Matrix

Frame kernel matrix finds the similarity between the two frames, $x_i$and x_j, by kernel means function $\emptyset(x_i)^T\emptyset(x_j)$ [20]. Here, the exponential frame kernel matrix, $k_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$ is used to compute the frame kernel matrix. Another name for frame kernel matrix is recurrent matrix as mentioned in [31]. This matrix will decompose the same time series at two different temporal scales, to prevent such uncertainty,$n_{max}$is introduced for segment limitation.

3.2.3. Dynamic Time Alignment Kernel

Dynamic time alignment kernel (DTAK) is the type of the support vector machine (SVM) and an expansion of dynamic time warping. Dynamic time warping (DTW) finds the closest matches in the dataset, used in retrieval [32]. It doesn't satisfy the triangular inequality and it is not a positive definite. So, DTAK is used [33].

For the sequence, $X = [x_1, x_2 \ldots . x_{nx}] \in IR^{d*n_x}$ and $Y = [y_1, y_2 \ldots . y_n] \in \in IR^{d*n_y}$, DTAK finds the similarity. Cumulative kernel matrix is needed by DTAK. So, it is necessary to build U for short sequences. Now, the bottom right of Cumulative

kernel matrix compared with the addition of other sequence lengths to find the final value of DTAK. The DTW has a recursive function given as

$$\tau(x,y) = \frac{u_{n_x n_y}}{n_x + n_y}, u_{ij} = \max\{ \begin{matrix} u_{i-1,j} + k_{ij} \\ u_{i,j} + 2k_{ij} \\ u_{i,j-1} + k_{ij} \end{matrix} \quad (3.4)$$

Where $k_{ij} = \emptyset(x_i)^T \emptyset(y_j) = \exp(-\frac{||x_i - y_j||^2}{2\sigma^2})$ is frame Kernel. This matrix notation implies that the DTAK has never decreasing growth, when l is the numbers of steps to align X and Y, the two frame indices for x and y parameters are $p \in \{1: n_x\}^l$ and $q \in \{1: n_y\}^l$. Now the new normalized corresponding matrix, W is defined as

$$W = [w_{ij}]_{n_x + n_y} \in IR^{n_x + n_y} \quad (3.5)$$

Where $w_{ij} = \frac{1}{n_x + n_y}(p_c - p_{c-1} + q_c - q_{c-1})$ only if the values of p_c and q_c exist for some class c, else $w_{ij} = 0$ [34]. Thus, reducing the DTAK equation as $\tau(x,y) = tr(K^T W) = \varphi(X)^T \varphi(Y)$, where $\varphi(.)$indicates mapping of sequence into feature space [20]. In DTAK, kernel matrix needs regularization, since it may not be positive sometimes [35].

3.2.4. Aligned Cluster Analysis (ACA)

It is the combination of the kernel k-mean and spectral clustering. ACA segments the data stream into actions ACA divides the sequence into disjointed segments corresponding to one of the classes. Each segment Y_i has an indicator matrix $G \in (0,1)^{k*m}$ and $g_{ci} = 1$ when Y_i belongs to class c with the start and end positions $S_i$and $S_{i+1} - 1$ respectively. ACA mixed with K-means gives

$$J_{km}(G,s) = \sum_{c=1}^{k} \sum_{i=1}^{m} g_{ci} \left|\left|\varphi(X_{(S_i, S_{i+1})}) - z_c\right|\right|^2 = ||[\varphi(Y_1), \ldots \varphi(Y_m)] - ZG||_F^2 \quad (3.6)$$

and $G^T 1_k = 1_m$ and $S_i - S_{i+1} \in [1, n_{max}]$ and the distance square is

$$dist_{\varphi}^2(y_i, z_c) = \tau_{ii} - \frac{2}{m_c} \sum_{j=1}^{m} g_{cj} \tau_{ij} + \frac{1}{m_c^2} \sum_{j1,j2=1}^{m} g_{cj_1} g_{cj_2} \tau_{j_1 j_2}, \quad (3.7)$$

and number of class c segments $m_c = \sum_{j=1}^{m} g_{cj}$. Construction of matrix includes following, Starting from the upper left corner $p_{1,1} = p_{0,0} + 2\text{k}(1,1)$. The other matrix values are filled by increasing the $p_{i,j}$ subscript. DTAK value is obtained by dividing

the bottom-right by the sum of the sequence lengths. Considering the short sequences, $\hat{x} = [1,2,1,2]$ and $\tilde{x} = [1,1,2,2]$ and taking $\sigma = \infty$, and k(1,1) and k(2,2)=1 and k(1,2) and k(2,1)=0. Thus, applying the left corner equation $p_{1,1} = 2$ The value of DTAK is calculated by dividing bottom right $p_{4,4}$ by sum of sequence length, $k_{dtak}(\hat{x}, \tilde{x}) = \frac{7}{8}$. The main advantage over kernel k-mean is that it can have different sample size. The distance is robust to noise and is constant to scaling factors as the DTAK is used [20].

3.3 Algorithm Flowchart

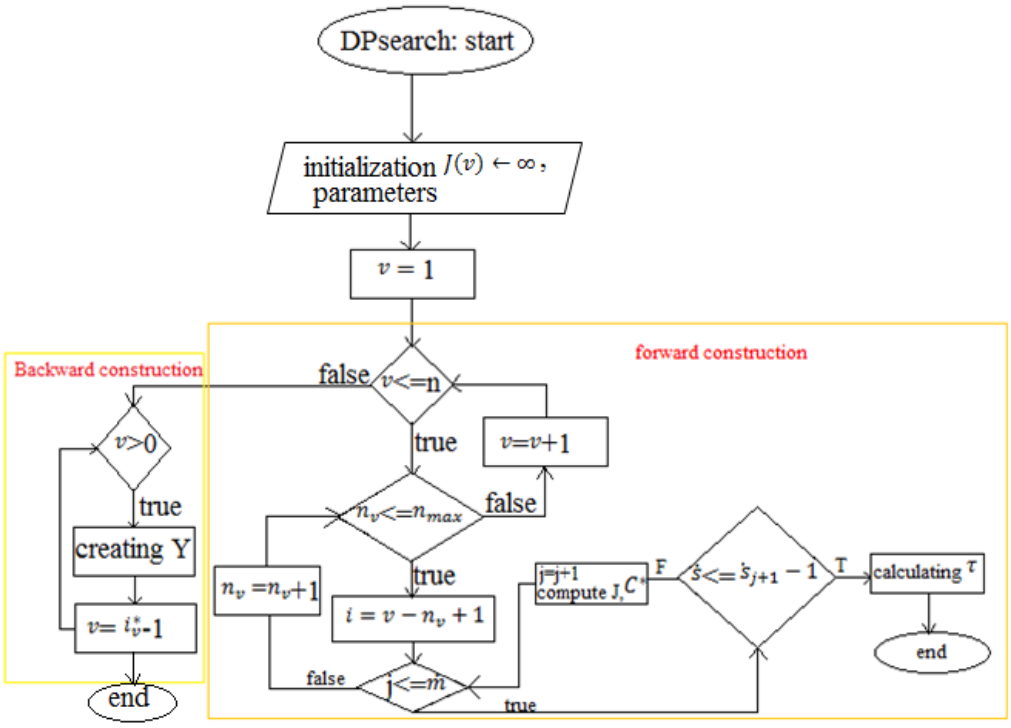


Figure 3.1: ACA Algorithm Flowchart

Applying the forward and backward step using ACA as shown in Figure 3.1; Forward step includes scanning from the beginning of the sequence to end i.e. from v = 1 to $v = n$, for every v the $J(v)$is computed which computes the DTAK between

segment and each of the segment of each classes. It stores the head position i_v^*, label g_v^* and $J(v)$, having the lowest error. Backward step includes tracing the sequence from the back i.e. from $v = n$ and cutting off the head segment, $S = i_v^*$ The Indication vector $g = g_v^*$ are recorded, further repeating the process on left of sequence [20].

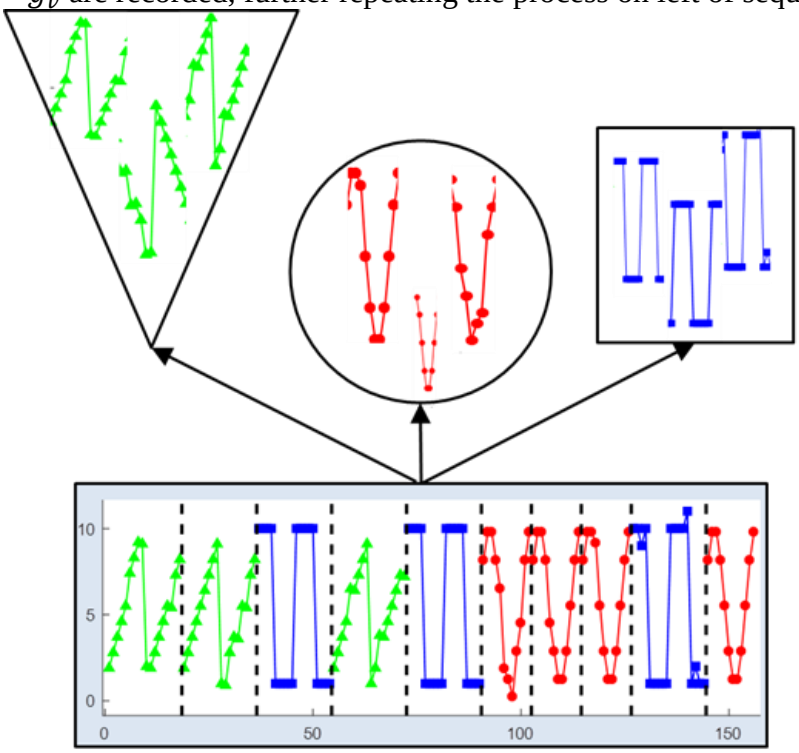


Figure 3.2: ACA the forward and backward tuning

In Figure 3.2, we have synthetic time series experiment, where there are three clusters i.e. three actions were performed randomly as a single level of temporal random noise. As can be seen, the three colors red, blue and green shows the three different groups. At start the numbers of frames are randomly initialized and so are the colors. For instance, considering Figure 3.3, the first two frames are color red and are belong to a single group but, the next frame is blue and forth frame is green, indicating the different groups. Fifth frame is again red, similar to first two frames, which belongs to the first two frames. Hence, the fist signal which segments from start till around 18 is same as the segment starting from around 29 to 39. Also, the noise level is inserted every 10th frame which is indicated as 0.10 in the figure 3.3.

Here, we are sampling 10 Gaussian distribution randomly, each time series with the frame length of 75. Figure 3.3 shows the demo of the toy data with the ground truth. It can be seen that the ACA accuracy is 96 percent.

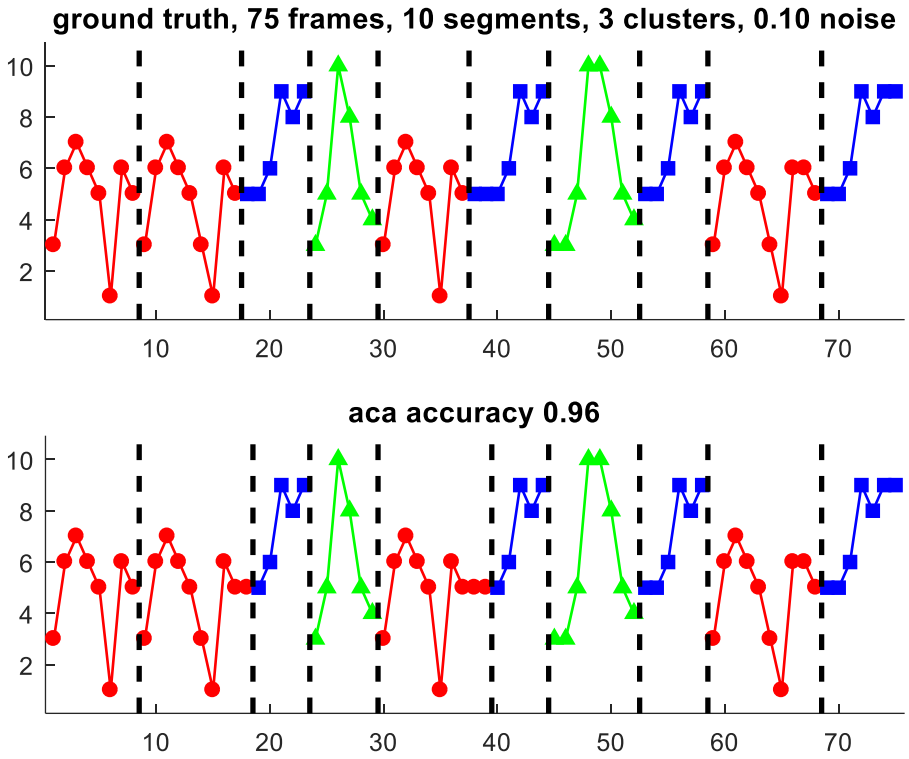


Figure 3.3: Synthetic Time Series Experiment using ACA Demo toy

CHAPTER 4:

REAL-TIME RECOGNITION

4.1 Definition

In the previous chapter, we have learned about the gesture segmentation, finding the start and the end of each motion. The next step is to find the type of motion and recognizing each motion is necessary, as it will help the patient know the variance of their motion, thus indicating if any improvement in their movement. Human activity is outlined as the temporal and spatial movement of various body postures. These postures may be repetitive. After capturing images from Kinect, the postures represented as a skeleton, will be segmented by kernel k-means algorithm, aligned cluster analysis and dynamic time alignment kernel. After which each segmented cluster need to be recognized, if it is walking or running or squatting from the four mentioned skeletal points.

Action recognition used several strategies such as using wearable sensors, SVM (support vector machine) technique and HMM (hidden Markov model) technique for feature extraction [36]. Wearable sensors might be limited to some environment, apart from the occlusion between the sensors and the person and change in viewpoint, execution speed and the camera motion can be the major barrier, while other might be the offline method. The other model using the HMM may not be suitable for the high dimensional data and with the low-level accuracy. As we are using the depth sensor and grey scale data, it can improve the performance of recognizing the complicated activities, while the single activity is a combination of multiple activities.

The multi-dimensional data produced must be highly accurate in order to recognize a single activity. So, in order to produce the high accuracy, Support vector machine (SVM) algorithm is used. SVM used is the multi-class classification

algorithm, which optimizes each parameter. Support Vector Machine classifier recognizes the human activities [37]. Features were computed to obtain the feature set. Different feature subsets were then evaluated based on the precision and recall scores. The purpose of this recognition is to make computers recognize people's gesture motion state by analyzing lower limb signals parameters. The interaction process between human and technology can be friendly, real and natural. There are many adopted signal recognition methods for gestural skeletal signal processing include Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Artificial Neural Network (ANN), Support Vector Machine (SVM) and combining these techniques. The first step is extracting motion characteristic parameters from gestural data sets then these extracted characteristic parameters are used for training to identify different states. Therefore, in order to accurately determine the person's state of motion, firstly it is important to effectively extract the video characteristic parameters of these states. The features that associated with extracted features are amplitude energy, duration of single movement, speed etc., and statistics.

Five significant statistical parameters which change significantly are counted, including maximum, minimum, mean, variance and standard deviation. The energy of the video signal changes drastically over time. The light energy is usually much smaller than the energy of actually motion video, the selection of the window function plays a decisive role in the representation of characteristics of this energy. Distinguishing between the actual skeleton and the skeleton formed from the noise is the main task. In this signal recognition system, energy is generally used as a feature of the three-dimensional parameters to represent the energy of this signal and its information.

4.2 SVM Multiclass Classification Algorithm

The SVM multi-class problem can be divided into a multiple SVM series that can be directly solved. Then obtain the final results of discrimination using the results of the series SVM problems. SVM is not widely used as there is a wide number of problems associated with it due to its computational and optimization issues. Support Vector machine can usually be used to compare exactly two classes, where it classifies by searching the hyperplane to separate those two classes' data points. Hyperplane decision is made by computing the distance of each classes plotting a margin and finding the maximum width of the two classes data points. Some of the boundary data points are selected as support vectors which are closest to the hyperplane. An SVM comes into existence when the data is signal or text and it has large-margin basically its work is to find out the decision boundary between the two far away classes it considers the vector space as the distance between the training samples and test samples are very large. SVM has multiple cases for two-class data sets that are linearly separable, and other data sets which are non-separable, multi-class problems, and non-linear models.

There are different types of model one of them is a linearly separable model. When there are exactly two classes, data from those two classes are linearly separable. This technique separates the two-class data sets by a decision boundary which is drawn at the center of the void between data items of the two classes. Comparing with the other algorithms, Naive Bayes algorithm chooses the best linear separator according to its criteria and on the other side perceptron algorithm searches for the linear separator. But the SVM in particular looks for multiple decision surfaces and chooses decision boundary where the data points are maximally away and the margin is calculated as the distance from the decision surface to the closest data usually the larger margin is made for no low certainty classification decisions giving

classification safety margin [38]. Only a few data points are considered and chosen as support vector which states the separator position. The figure (figure. 4.1) below shows the support vectors and classification of the two classes of motion stand and walk. Sometimes there are misclassifications if a slight error in measurement.

SVM algorithm chooses a decision hyperplane. This decision hyperplane is defined by two components, decision hyperplane normal vector $\vec{\omega}$ and an intercept 'b' and it is perpendicular to the hyperplane and this vector is referred as a weight vector. We specify the intercept value b for choosing the hyperplane perpendicular to the normal vector. As it is perpendicular to the normal vector, all points $\vec{x}$ on the hyperplane satisfy equation $\vec{\omega}^T\vec{x} = -b$. Now consider the training data sets $D = \{(\overrightarrow{x_i}, y_i)\}$, where each member is a pair of a point $\overrightarrow{x_i}$ and a class label y_i corresponding to it. For SVMs, the two data classes are always named as +1 and -1 and not 1's and 0's [39] [40] [41]. The linear classifier function is

$$f(\vec{x}) = sign(\vec{\omega}^T\vec{x} + b) \tag{4.1}$$

A value of -1 indicates one class, and a value of +1 indicates the other class. For a given data set and decision hyperplane, we define functional margin of the i^{th} example $\overrightarrow{x_i}$ with respect to a hyperplane$\langle\vec{\omega}, b\rangle$ as the quantity $y_i(\vec{\omega}^T\overrightarrow{x_i} + b)$. The functional margin of single data set with respect to a decision surface is twice the functional margin of any of the points in the data set with minimal functional margin. To increase the functional margin, there is a need of scaling up the values of $\vec{\omega}$ and b. But here lies the problem that if replacing $\vec{\omega}$ by $5\vec{\omega}$ and b by $5b$ then the functional margin $y_i(5\vec{\omega}^T\overrightarrow{x_i} + 5b)$ would be five times larger. Therefore, there should be constraint placement on the $\vec{\omega}$ vector size [39].

SVMs are actually two-class classifiers. And the conventional method for multi-classification depends if the classes are mutually exclusive and can be linearly classified, where it builds classifier with the training set as positive class classifier and

negative class classifier for all the classes and then testing the data sets if it includes in positive class or negative and in this case decision doesn't depends on other classifier.

But for multiclass classification, one might need to use one-of classification with linear classifiers which build classifier with the training set as a positive class classifier and negative class classifier for all the classes. Now, it applies each classifier separately by giving each class a maximal score, confidence value, and probability on the test data, which then helps in constructing a confusion matrix, which shows if there are wrong classes assigned. Confusion matrix improves the accuracy of the system.

Another technique is to generate one-versus-all method which classifies the test sets with the greatest margin. Whereas one-versus-one classifiers choose the class that is selected by the most classifiers and in this case training classifiers time actually decreases as the training data set for each classifier is much smaller.

SVM methods generally are applicable and solve the variety of input problems which includes one versus all methods for classification, one versus one method, etc.

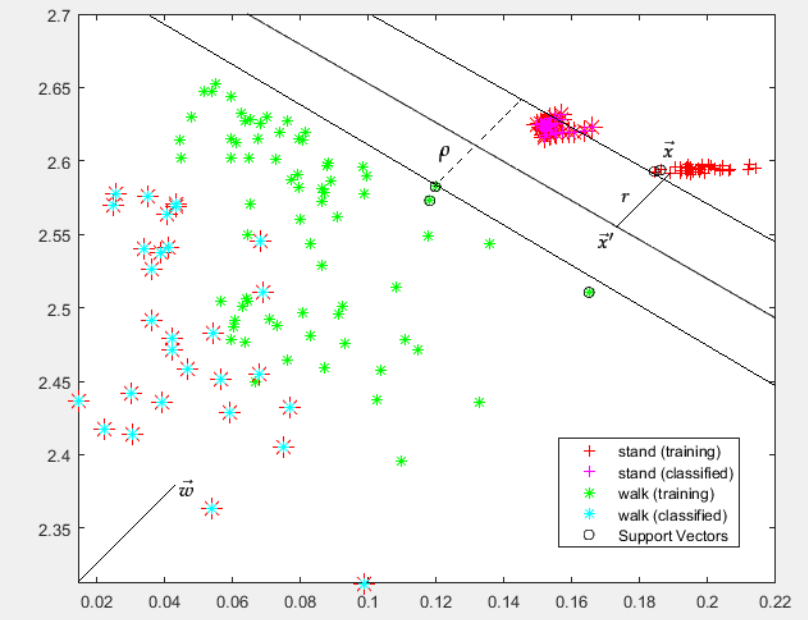


Figure 4.1: SVM Classifier Description Image

The shortest distance between a point and a hyperplane is perpendicular to the plane, and hence, parallel to $\vec{\omega}$. A unit vector in this direction is $\vec{\omega}/|\vec{\omega}|$. The dotted line in the diagram is then a translation of the vector $r\vec{\omega}/|\vec{\omega}|$. Labelling the point on the hyperplane closest to $\vec{x}$ as $\vec{x}'$. Then:

$$\vec{x}' = \vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|} \tag{4.2}$$

where multiplying by y just changes the sign for the two cases of $\vec{x}$ being on either side of the decision surface. Moreover,$\vec{x}'$ lies on the decision boundary and hence satisfies $\vec{\omega}^T\vec{x}' + b = 0$. Hence:

$$\vec{\omega}^T\left(\vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|}\right) + b = 0 \tag{4.3}$$

Solving r

$$r = y\frac{\vec{\omega}^T\vec{x}+b}{|\vec{\omega}|} \tag{4.4}$$

Again, the points closest to the separating hyperplane are support vectors. The classifier's geometric margin is the maximum width that is drawn separating the support vectors of the two classes. It is twice the minimum value over data points for r, or, equivalently, the maximal width of one of the fat separators. The geometric margin is clearly invariant to scaling of parameters: if we replace $\vec{\omega}$ by $5\vec{\omega}$ and b by $5b$, then the geometric margin is the same, because it is normalized by the length of $\vec{\omega}$. This means that imposing any scaling constraint on $\vec{\omega}$ will not affect the geometric margin. Using $|\vec{\omega}| = 1$ makes the geometric margin same as the functional margin [42].

Since scaling the functional margin solves for larger SVMs, choosing the functional margin of all data points of at least 1 and equal to 1 for at least one data vector. So, for all items in the data:

$$y_i(\vec{\omega}^T\vec{x}_i + b) \geq 1 \tag{4.5}$$

There exist support vectors for which the inequality is equality. Since each example's distance from the hyperplane is $r_i = y_i(\vec{\omega}^T\vec{x}_i + b)/|\vec{\omega}|$, the geometric margin is $\rho = 2/|\vec{\omega}|$. Desire is to maximize this geometric margin. That is, to

find $\vec{\omega}$ and b such that: Geometric margin $\rho = 2/|\vec{\omega}|$ is maximized for all $(\vec{x}_i, y_i) \in D$, $y_i(\vec{\omega}^T\vec{x}_i + b) \geq 1$. So, in short, minimizing $|\vec{\omega}|/2$ gives the final standard formulation of an SVM as a minimization problem. Find $\vec{\omega}$ and b such that $\frac{1}{2}\vec{\omega}^T\vec{\omega}$ is minimized and for all $\{(\vec{x}_i, y_i)\}$, $y_i(\vec{\omega}^T\vec{x}_i + b) \geq 1$, condition is satisfied.

Optimizing a quadratic function subject to linear constraints is a standard, well-known class of mathematical optimization problems. However, understanding the shape of the solution of an optimization problem involves constructing a dual problem where a Lagrange multiplier α_i is associated with each constraint $y_i(\vec{\omega}^T\vec{x}_i + b) \geq 1$ in the primal problem: finding $\alpha_1, \ldots \alpha_N$ such that $\sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$ is maximized and $\sum_i \alpha_i y_i = 0$, $\alpha_i \geq 0$ so that the solution is of the form $\vec{\omega} = \sum \alpha_i y_i \vec{x}_i$ and $b = y_k - \vec{\omega}^T\vec{x}_k$ for any $\vec{x}_k$so that $\alpha_k \neq 0$.

In the solution, most of the α_i are zero. Each non-zero α_i indicates that the corresponding $\vec{x}_i$ is a support vector. The classification function is then:

$$f(\vec{x}) = sign(\sum_i \alpha_i y_i \vec{x}_i^T \vec{x} + b) \quad (4.6)$$

Both the term to be maximized in the dual problem and the classifying function involves a dot product between pairs of points ($\vec{x}$ and $\vec{x}_i$ or $\vec{x}_i$ and $\vec{x}_j$), and that is the only way the data are used [43] [44].

To recap, starting with a training data set, the data set uniquely defines the best separating hyperplane and feeding the data through a quadratic optimization procedure to find this plane. Given a new point $\vec{x}$ to classify, the classification function $f(\vec{x})$ computes the projection of the point onto the hyperplane normal. The sign of this function determines the class to assign to the point. The value of $f(\vec{x})$ may also be transformed into a probability of classification; fitting a sigmoid to transform the values is standard. Also, since the margin is constant, if the model includes dimensions from various sources, careful rescaling of some dimensions may be required. So, to maximize the margin, it goes through two cases, hard margin and

then soft margin SVM [45]. There are two classifier stages; first is one vs all and second is one vs one classifier.

One-vs-All Classification

First, Building N different binary classifiers and For the i^{th} classifier, consider all the points in class I as positive example, and all the points not in class I as negative example. Let f_i be the i^{th} classifier. Classify with

$$f(x) = \arg max_i \, f_i(x) \tag{4.7}$$

One vs one classifier

Building $N(N-1)$ classifiers, one classifier to distinguish each pair of classes i and j. Let f_{ij} be the classifier where class i were positive examples and class j were negative. Note $f_{ji} = -f_{ij}$. Classify using $f(x) = \arg max_i \, (\sum_j f_{ij}(\mathrm{x}))$. One vs one method is memory efficient and is fast. If the time to build a classifier is super-linear in the number of data points, this method is a better choice.

4.3 Multiclass Error Correcting Output Code (ECOC) model for SVM

SVM is basically used to differentiate between two class samples. But the problem arrives when there are more than two classes. To combat this problem, error correcting output code model (ECOC) is specially made to separate more than three classes. Here the method of error-correcting output code is used where the task is to classify the daily motion activities into m=5 categories {walk, run, stand, left leg raising, right leg rising}. Firstly, a unique n-bit vector is assigned to each. Viewing the i^{th} bit vector as a unique coding for label i. For this reason (and others, which will soon become apparent), referring to the set of bit vectors as a code and denote it by C. The ith row of C can be written as C_i , and the value of the jth bit in this row as C_{ij} [46]. The second step in constructing an ECOC classifier is to build an individual binary classifier for each column of the code. The positive instances for classifier j are

documents with a label i for which C_{ij} = 1. The third classifier, for instance, has the responsibility of distinguishing between all the labels. Conventional method usually refers algorithm 1 to predicting the value of a single bit as a "plug-in classifier" (P_iC). A P_iC, then, is a predictor of whether a document belongs to some fixed subset of the classes. To summarize, training an ECOC classifier consists of learning a set $\Lambda = \{\lambda_1, \lambda_2, \ldots \lambda_n\}$ of independent binary classifiers [46]. With Λ in hand, one can hypothesize the correct class of an unlabeled document x as follows. Evaluate each independent classifier on x, generating a n-bit vector $\Lambda(x) = \{\lambda_1(x), \lambda_2(x), \ldots \lambda_n(x)\}$. Most likely, the generated bit vector $\Lambda(x)$ will not be a row of C, but it will certainly be closer to some rows than to others by calculating hamming distance. Categorizing the document x involves selecting $argmin_i\ \Delta(C_i, \Lambda(x))$. Decision boundaries for the plug-in classifiers corresponding to the code are given below in Table 4.1 [46].

Algorithm 1:

Training an ECOC document classifier

Input: Documents $\{x_1, x_2, \ldots x_D\}$; Labeling $\{y_1, y_2, \ldots y_D\}$ (with m distinct labels);

Desired code size $n \geq \log_2 m$

Output: m by n coding matrix C; n classifiers $\{\lambda_1, \lambda_2, \ldots \lambda_n\}$

1. Generate a m by n 0/1 coding matrix C

2. Do for $j \in [1, 2 \ldots n]$ – Construct two super classes, S_j and $\bar{S}_j$. S_j consists of all labels i for which $C_{ij} = 1$, and $\bar{S}_j$ is the complement set. – Construct a binary classifier λ_j to distinguish S_j from $\bar{S}_j$. Which C_i is closest to $\Lambda(x)$. (If more than one row of C are equidistant to $\Lambda(x)$, select one arbitrarily.)

To the extent that rows of C are well-spaced in Hamming distance, the classifier will be robust to a few errant PiCs [46] [47]. This is the idea behind error-correcting codes as well: to transmit a point in the m-dimensional cube reliably over a noisy channel, map it to one of a set of well separated "fixed points" in a higher-

dimensional cube; to recover the original point, find the closest fixed point to the point actually received and take its preimage in the original cube. In general in algorithm 2, $\lambda_j(x)$---may not be a 0/1 value, but a real valued probability, measuring the classifier's confidence that document x belongs in the j^{th} superclass. One can search for the nearest neighbor according to some Lp distance, rather than Hamming distance.

Algorithm 2:

Applying an ECOC document classifier

Input: Trained ECOC classifier: m by n coding matrix C and n classifiers$\{\lambda_1, \lambda_2, \ldots \lambda_n\}$ unlabeled document x

Output: Hypothesized label y for x:

1. Do for $j \in [1, 2 \ldots n]$ – Compute $\lambda_j(x)$---the confidence with which $PiC\ j$ believes $x \in S_j$.

2. Calculate $\Delta(\Lambda(x), C_i) = P_{nj} = 1|\lambda_j(x) - C_{ij}|$ for $i \in [1, 2 \ldots m]$.

3. Output $argmin_i \Delta(\Lambda(x), C_i)$, the plug-in classifiers output a probability, and computing the nearest neighbor according to L_1 distance [46] [47] [48].

Each class is assigned a unique binary string of length 5. The string is also called a code word. During training, one binary classifier is learned for each column. Thus 5 binary classifiers are trained in this way. To classify a new data point x, all 5 binary classifiers are evaluated to obtain a five-bit string. Finally, we chose the class whose code word is closest to x's output string as the predicted label [46] [47].

Some standard classification algorithms such as backpropagation are best suited to distinguishing between two outcomes. A natural way to combine such algorithms to predict from among k > 2 outcomes is to construct k independent predictors, assigning predictor i the task of deciding whether the i^{th} outcome obtains.

To build the classifier, construct m individual classifiers, where the positive examples for classifier λ_i are those documents with label i. To apply the classifier to an unlabeled document x, select $i^* = argmax_i \lambda_i(x)$ the label whose classifier produces the highest score which is called as the one versus rest strategy.

Table 4.1: 5-bit Error-Correcting Output Code cost matrix for five classes.

Class	Cost				
0	0	1	1	1	1
1	1	0	1	1	1
2	1	1	0	1	1
3	1	1	1	0	1
4	1	1	1	1	0

This method is a special case of ECOC classification where C is the m by m identity matrix in our case table 4.1. To see why one might expect ECOC classification to outperform a one-vs.-rest approach, considering the problem of learning single data for walking. Within the labeled set of examples used to train the individual one-vs.-rest classifiers, the only data points matching the walk data sets will be compared. So, λ^{walk} will learn a strong association between all the 24 data points of walk. Now giving data points of run activity same as the walking data points to the trained one-vs.-rest classifier. The value of λ^{run}(x) will likely be close to one after all. But the value of λ^{walk}(x) will be very close to one, and the system will misclassify the object as a walk. ECOC classification is less brittle than the one-vs.-rest approach: the distributed output representation means one errant subordinate classifier won't necessarily result in a misclassification. This is an obvious way to say that ECOC reduces variance of the individual classifiers. Many classification algorithms, including decision trees and neural networks have the capability to

directly perform multiway classification. Consider the classes as clouds in a large-dimensional feature space; a single classifier must learn all the decision boundaries simultaneously, whereas each data of an ECOC classifier only learns relatively small number of decision boundaries. Moreover, (assuming n is sufficiently large) an ECOC classifier learns each boundary many times, and it ignores if a few data places the input on the wrong side of some decision boundaries [47] [48].

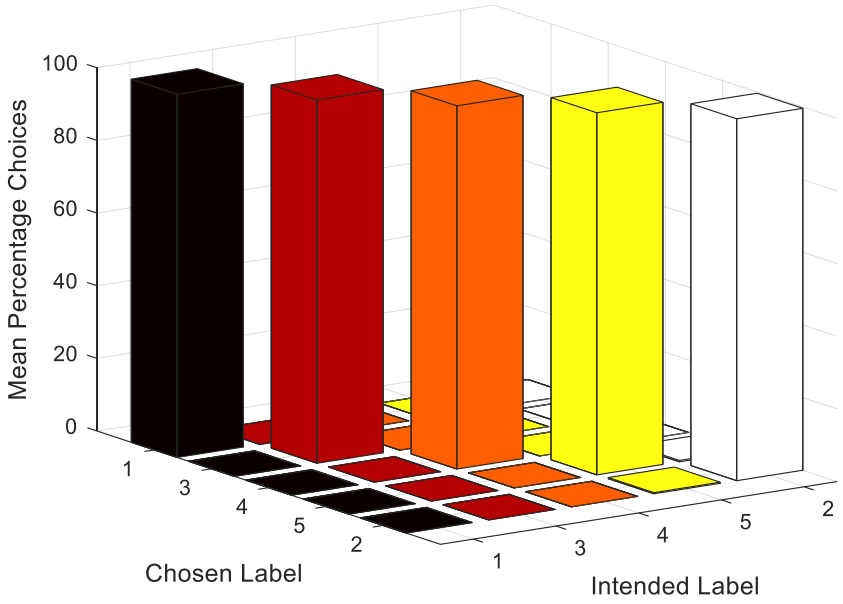


Figure 4.2: 3-Dimension Confusion matrix plot

Table 4.2: Confusion accuracy

	1	2	3	4	5
1	100	0	0	0	0
2	0	100	0	0	0
3	0	0	100	0	0
4	0	0	0	99.6932	0.37174
5	0	0	0	0.30674	99.6282

The samples for five activities; walking, running standing and left and right leg-raising are trained to test the SVM code in MATLAB and the same trained data are fed for testing. Here the number block 1 shows the activity 1 which is described as stand, 2 as walk, 3 as run 4 as left leg raise and 5 as right leg raise. These activities are compared by the corresponding activities. The result is simulated in three dimensional in figure 4.2, where it can be seen that the blocks are all 100 percent showing that the all the activities are 100 percent correctly recognized as these are the results which are given when the same numbers of trained data are fed to test data. Similarly, in two-dimensional in figure 4.3, the output classes from 1 to 5 activities are compared with the same and with others, here there are 1000 samples in all. For example, activity one corresponding to same activity and as seen the number of samples taken are 151(first block from left) and all the samples are from activity 1 giving the total of 15.1% correct classification and others blocks shows 0% showing that all the simulated samples are correctly recognized. The simulation result proves the accuracy of SVM code. Table 4.2 shows the confusion plot for the same code with the second trial.

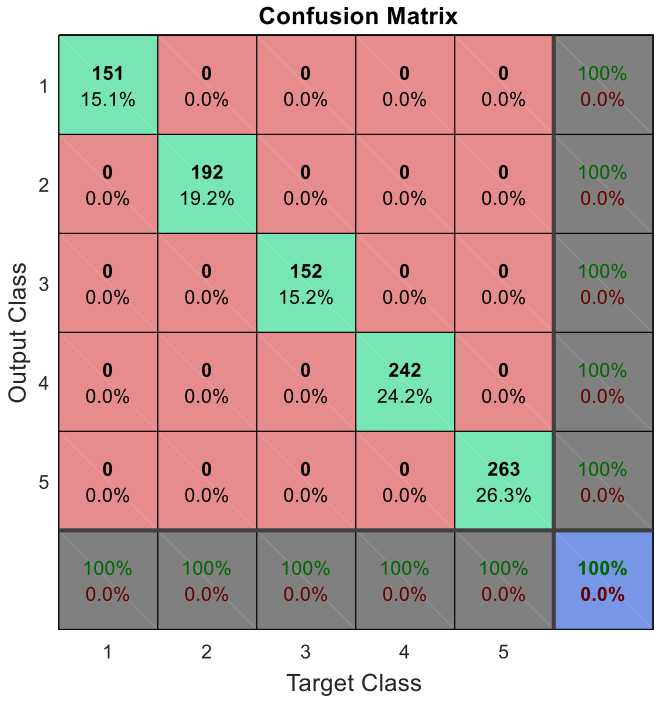


Figure 4.3: Confusion Plot

CHAPTER 5:

EXPERIMENTAL RESULTS

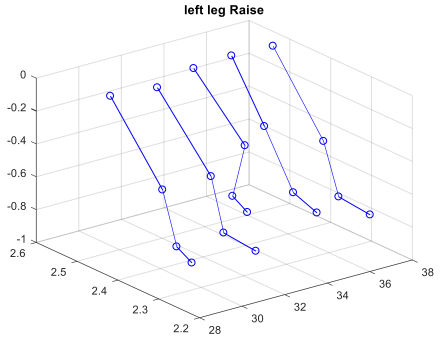
5.1 3D Modeling

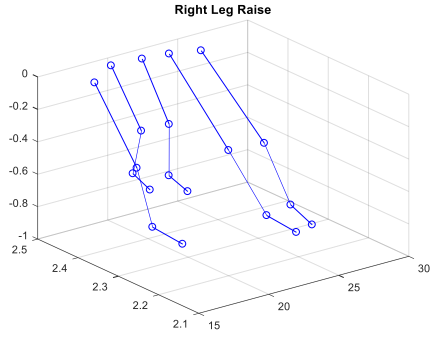
5.1.1 Experimental Setup

In this experiment, the human subject standing in front of Microsoft Kinect makes the lower limb movement. These five movements were standing, walking, leg side-raising and running. The movements are done approximately 5 foot away from Microsoft Kinect and facing the Kinect Camera sensors. The motions are video recorded by the Kinect with the help of MATLAB for around 80 seconds to 120 seconds. These videos and data points are recorded and saved in MATLAB for the two subjects.

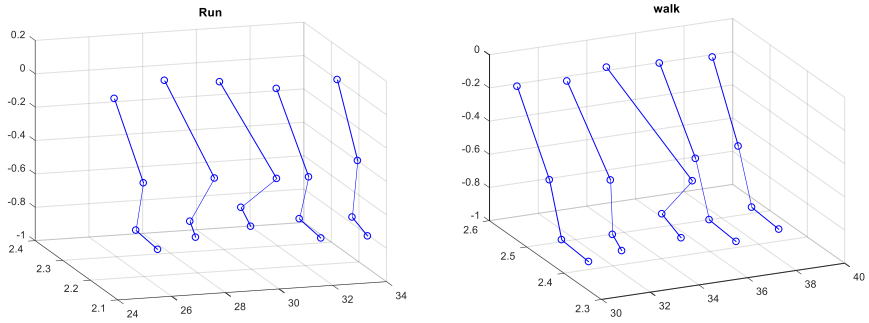
5.1.2 Lower-Limb Data

The two subject data, one normal and one with knee injury is used. The lower limb points are from hip point to foot point. Each point is a 3D data point and the total of 4 points including hip point, knee point, ankle point and foot point. Therefore, in a single data capture, we have 3*4 points making 12 data points for each leg, making a total of 24 data points for both the legs. These 24 data points are processed online, segmenting each section. Figure 5.1 shows the five 3D leg movement skeletal plots with 5 frames. These movements are left leg raise, right leg raise, run walk and stand.

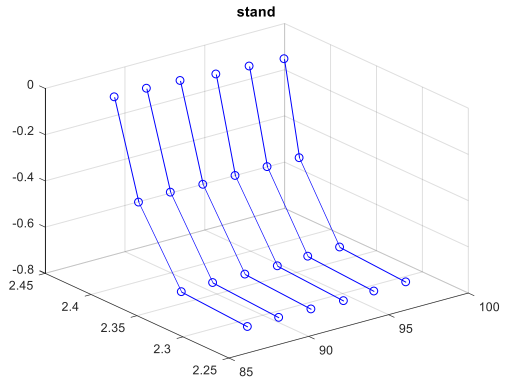




a. Left leg raise *b. Right leg raise*



c. Run *d. Walk*



e. Stand

Figure 5.1: Leg Movement skeletal plot for (a) Left leg raise (b) Right leg raise (c) Run (d) Walk and (e) Stand.

5.2 Online segmentation results

5.2.1 Simulation Results

In this section, we first test our online segmentation algorithm with simulation data. Three 2D waveforms are used, square waveform, triangle waveform, and saw-tooth waveform. The three waveforms are generated randomly with about 18 frames per segment. The noise level of 0.1 is added to the waveforms. Figure 5.2 shows the kernel matrix of DTAK. It shows the similarity of each segment. For example, the mixed waveform segment at around 20 is similar to the mixed waveform segment at around 45, 85 and 95.

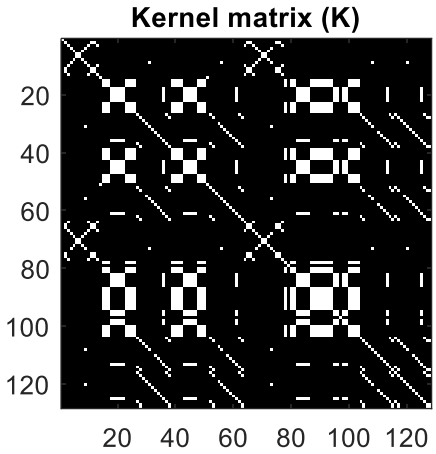


Figure 5.2: Kernel Matrix for toy data

The segments are given in figure 5.3. As shown, three colors represent the three different waveforms. The color doesn't represent the recognition results. The color is initialized randomly. For example in figure 5.3, the first segment the algorithm ACA found gives the color red, the second is colored blue and the third is green. The fourth is similar to the second segment, thus it is given the blue color. The accuracy of segmentation is 96%. Figure 5.4 shows the segment with the bar plot.

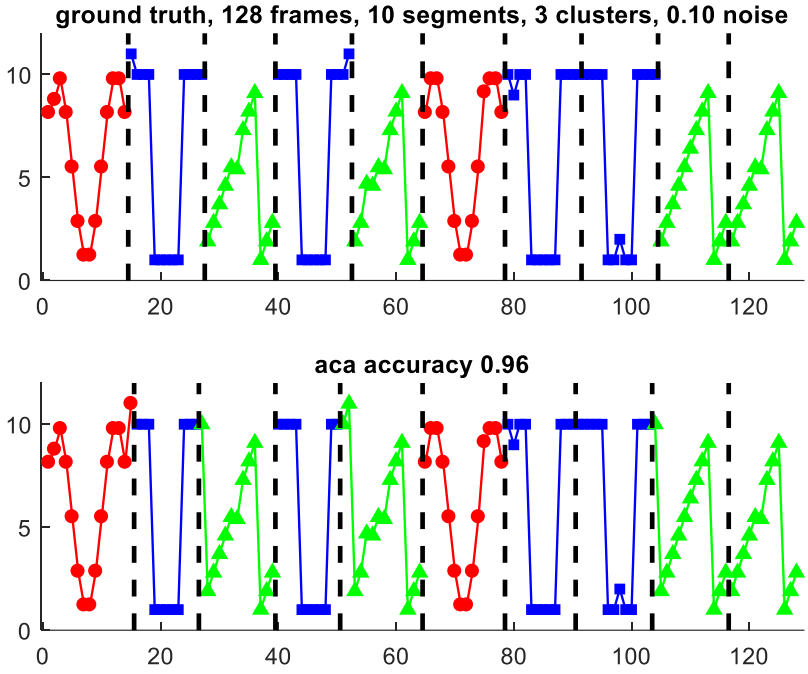


Figure 5.3:2-D Simulations with three Different signals

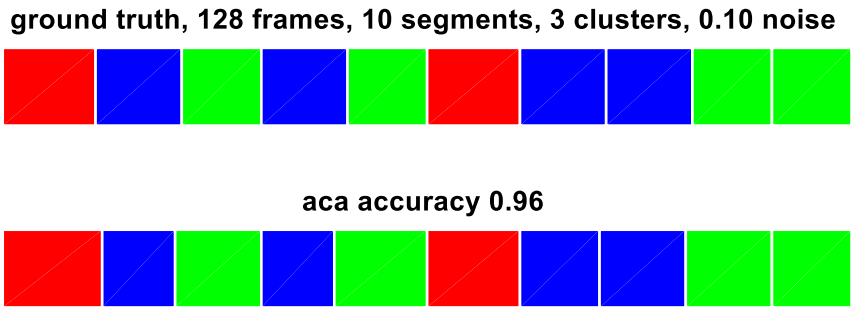


Figure 5.4: Bar plot of Simulation for Segmentation

5.2.2 Real Data Segmentation

The segmentation results with the real data are given below. Here, we use the mixing movement with the three different activities; walk, run and right leg raise. The experiment is set like this. The subject will walk first and then run. After about 100 seconds. The subject will slow down to walk and then does the leg raising activity. All the movements of 8 joint points (4 for each leg) are collected. Figure 5.5 shows the kernel matrix of the real experiment data segments similarity. Note that here, we only show the joint 14 (right knee) waveforms results. Only x-axis data is a plot here. Similarly, the 2D segmented waveform, bar plot with the ground truth is provided in figure 5.6 and figure 5.7.

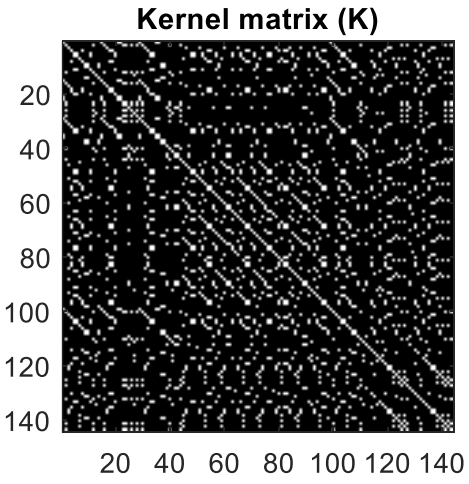


Figure 5.5: Kernel matrix for Real data

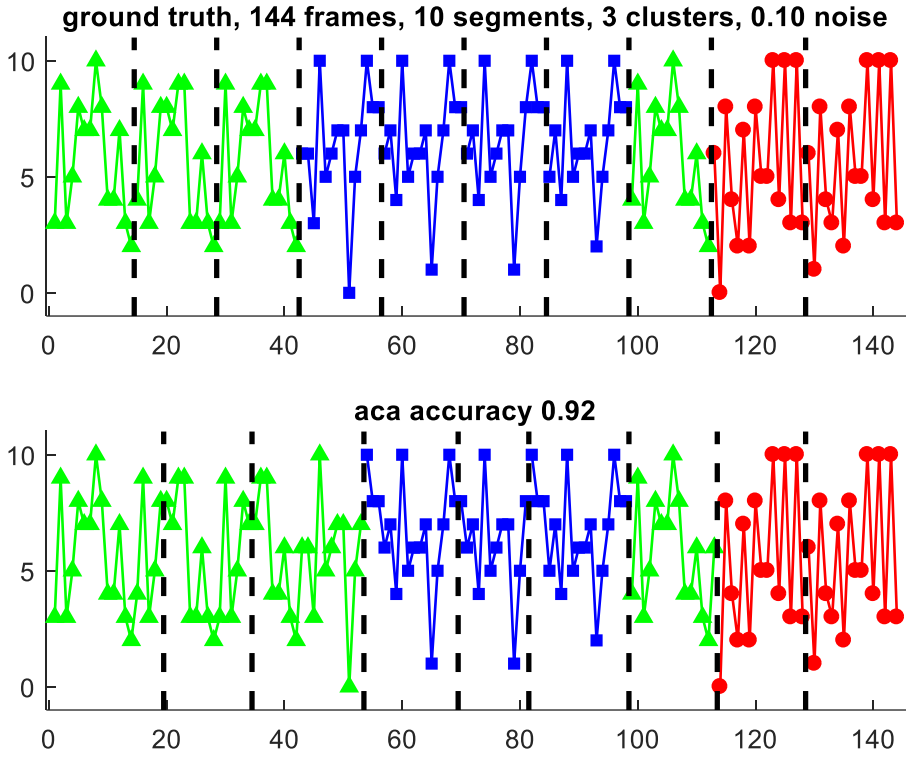


Figure 5.6: X-axis data of Right Knee Point (joint 14 from table 2.1)

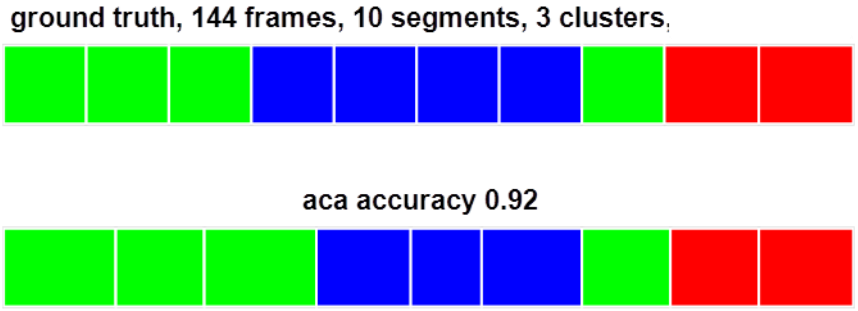


Figure 5.7: Bar plot for Real data

The segmented accuracy based on the joint 14 with x-axis data is approximately 92 percent for three different activities. Please note, we normalize the data value to 0-10. All the 8 joints are segmented with x-y-z-axis waveforms. The final segmentation is given by averaging all the 24 (8x3) segmentation results.

5.3 Real-Time Recognition Results

5.3.1 SVM for Two Activity Recognition

The Support Vector Machine trains the two classes and classifies the two group test data for a walk and stand in figure 5.8. It also shows the support vectors for both the classes in the circles, where the Support vectors are the samples from the trained classes itself. It can be seen that the green data are the walk left leg knee data. The variance between walking data is too large as the walking steps tends to change with the high speed or with the low speed. The trained walking data sets are different from the walking classified data; still the data are correctly classified as the blue tested samples. Similarly in the stand data, but the activity stand variance is very low as the movements are not actually taking place, giving the best accuracy for motion.

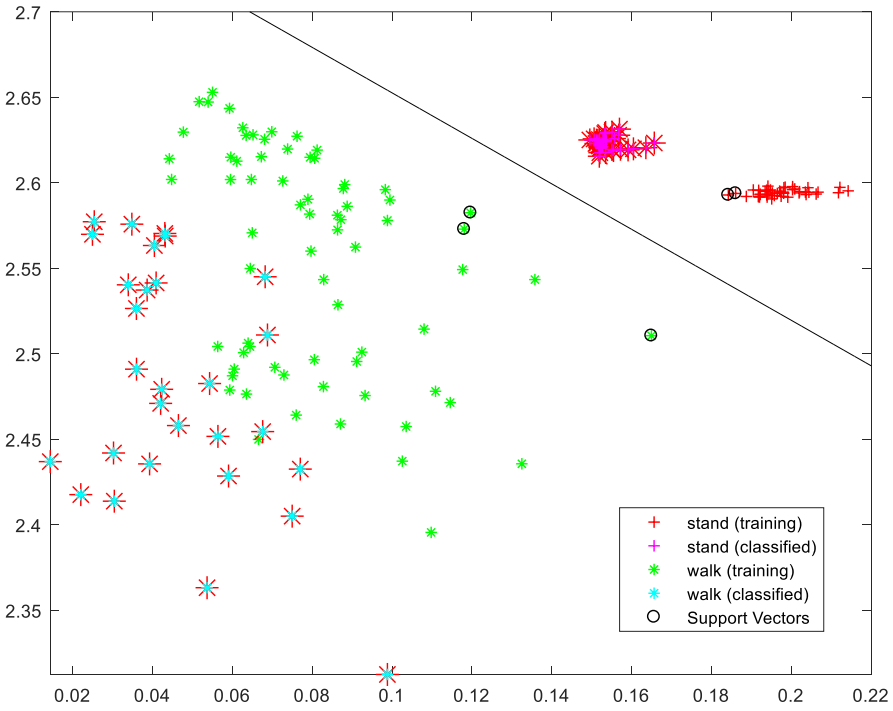


Figure 5.8: Two class SVM plot for Walk and Stand

5.3.2 Comparison Matrix

The two subjects; the one with the normal gesture and another subject with the knee injury conducts the experiment. The first subject stands still for some movements and starts to walk followed by left leg raise. The subject then stands and then raises right leg following by standing, running and walking. The second subject starts by raising his left leg followed by right leg, and then he starts running and after around 80 seconds he slows down to walk and ultimately stops walking, while still in standing position. The two results are trained and tested by SVM. In the confusion matrices, activity 1 is a stand, 2 walks, 3 is run 4 is left leg raise and 5 is right leg raise.

The five diagonal cells show the trained network's percentage of correct classifications and the number of trained data in figure 5.9. Total of 1,314 data samples are tested. From the leftmost block, 216 tests are classified correctly as activity 1, which corresponds to 16.4% of the total 1,314 data. And similarly it can be seen that 268 samples are correctly classified as activity 2, which corresponds to the 20.3% of all the activities performed. It can be seen from row 2 that the 2 activities are misclassified, where each of them corresponds to 0.1% of 1,314 activities. Similarly for row 3, 2 samples are misclassified as activity 4 and one is misclassified as activity 2 (walk), whereas the actual activity was run. This is because, sometimes the person running would slow down which inherently matching the data points of activity walk. So, running is predicted as walk, thus, interpreting activity run as another activity walk.

The confusion matrix of all the five activities is plotted in figure 5.9 and figure 5.10 giving the accuracy of 99.6% for the first subject and 95.7% for the second subject respectively. As seen from figure 5.10, the activity walk (row two), is classified correctly as 21.6 percent of the total 425 performed movements. While the activity is wrongly classified as run and right leg raise as the person moving might be

walking with the same motion as the right leg raising. The most misclassified activity is the activity run from row 3 as 82 samples are correctly classified i.e. 18.5% of 425 movements and at the same time 10 activities are misclassified giving 2.3% of misclassification decreasing the accuracy of activity 3 (run) to 82.8%.

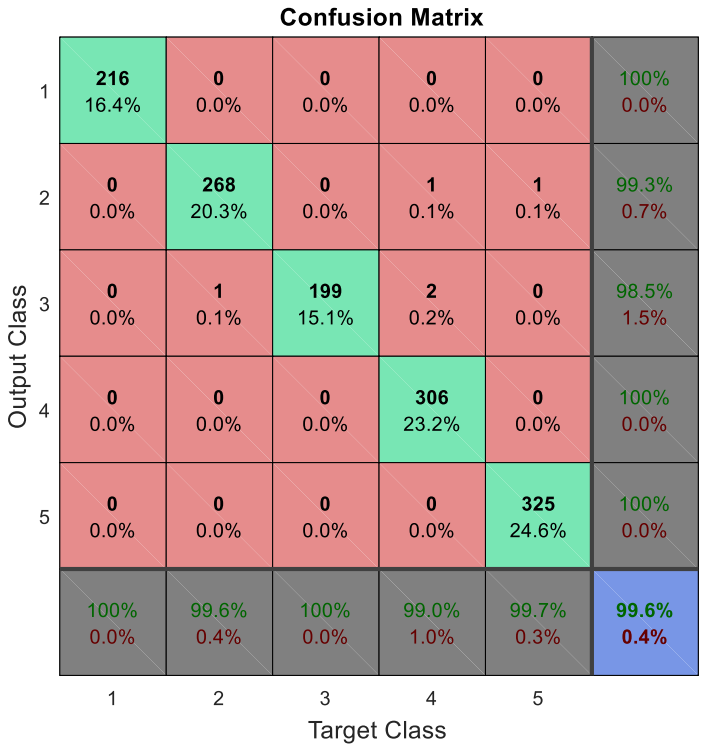


Figure 5.9: 2D Confusion matrix plot for Real 3D Data for Subject 1

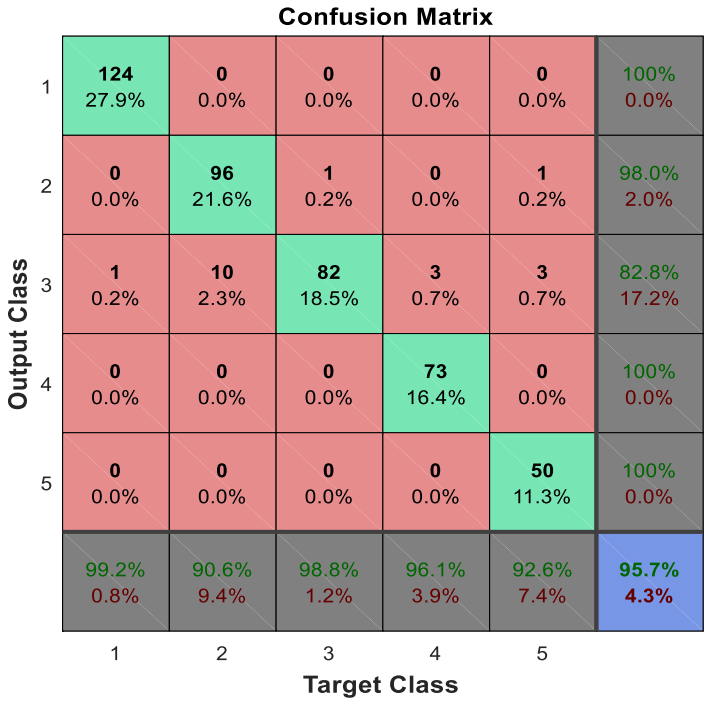


Figure 5.10: 2D Confusion matrix plot for Real 3D Data for Subject 2

The algorithm of Aligned cluster analysis gives the better result with the dynamic time alignment kernel algorithm. This result provides with the segmented data of the raw data points, which gives the different group segmented data.

Some standard classification algorithms such as backpropagation are best suited to distinguishing between two results. But the better result is given when the combination of the two Support Vector Machine and the error correcting code algorithm give the accuracy of more than 99 percent. The result is well-recognized data into their respective group.

These two segmentation and recognition result combined provides the better efficiency and accuracy.

CHAPTER 6:

CONCLUSION AND FUTURE WORK

Activity recognizing is a crucial step for the injured and elderly people. As for the elderly person, it is quite difficult to differentiate in two different routine activities. This thesis helps in recovering the patient indirectly when this research is combined with other techniques. This research segments the three-dimensional data of different leg movements. The accuracy of the segmentation is high enough differentiating the two activities and thus recognizing the three-dimensional data points using SVM if its walk or run or stand. The goal of this approach is to do the three-dimensional data recognition by efficiently identifying dissimilar activities when there are multiple actions in a single frame.

This research can be progressed more with the different machine learning methods, especially the deep learning technique where the three-dimensional data can be fed to Convolution neural network (CNN) and fed back and forth via the feedforward network block. Also, with the multiple layers of neural network, it will be helpful to recognize the data easily. In the planned future work the CNN method extends this three-dimensional data recognition. Also, the system data generation can be fed directly to the distant doctors to analyze the patient from far via Internet of Things applications.

CHAPTER 7:

REFERENCE

[1] A. Nazábal, P. García-Moreno, A. Artés-Rodríguez and Z. Ghahramani, "Human Activity Recognition by Combining a Small Number of Classifiers," *IEEE Journal of Biomedical and Health Informatics,* vol. 20, no. 5, pp. 1342 - 1351, 2016.

[2] J. F.-S. Lin and D. Kulić, "Online Segmentation of Human Motion for Automated Rehabilitation Exercise Analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 22, no. 1, pp. 168 - 180, 2014.

[3] M. Z. U. a. T. K. A. Jalal, "Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home," *IEEE Trans. Consum. Electron.,* vol. 58, no. 3, p. 863–871, 2012.

[4] C. Chen, R. Jafari and N. Kehtarnavaz, " Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors," *IEEE Transactions on Human-Machine Systems,* vol. 45, no. 1, pp. 51 - 61, 2015.

[5] J. Lu, T. Zhang, Q. Sun, S. Kadiwal, I. Unwala and F. Hu, "Monitoring of paces and gaits using binary PIR Sensors with rehabilitation treadmill," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

[6] A. A. a. C. S. M. Soriano, "Curve spreads C a biometric from front-view gait video," *Pattern Recognit. Lett,* vol. 25, no. 14, p. 1595–1602, 2014.

[7] Z. Kang, M. Deng and C. Wang, "Frontal-view human gait recognition based on Kinect features and deterministic learning," in *2017 36th Chinese Control Conference (CCC)*, 2017.

[8] X. Z. F. L. Y. W. a. Q. W. W. Zeng, "A New Kinect-Based Frontal View Gait Recognition Method via Deterministic Learning," in *Proceedings of the 35th Chinese Control Conference*, 2016.

[9] S. N. Reddy, S. R. Dumpala, S. K. Sarna and P. G. Northcott, "Pattern Recognition of Canine Duodenal Contractile Activity," *IEEE Transactions on Biomedical Engineering*, pp. 696-701, 1981.

[10] R. Polana and R. Nelson, "Detecting activities," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New York,, 1993.

[11] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang and P. O. Ogunbona, "Action Recognition From Depth Maps Using Deep Convolutional Neural Networks," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 498 - 509, 2016.

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

[13] M. Ramanathan, W.-Y. Yau and E. K. Teoh, "Human Action Recognition With Video Data: Research and Evaluation Challenges," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 5, pp. 650 - 663, 2014.

[14] N. Rossol, I. Cheng and A. Basu, "A Multisensor Technique for Gesture Recognition Through Intelligent Skeletal Pose Analysis," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 350 - 359, 2016.

[15] Z. Lu, X. Chen, Q. Li, X. Zhang and P. Zhou, "A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp.

293 - 299, 2014.

[16] J. F.-S. Lin, M. Karg and D. Kulić, "Movement Primitive Segmentation for Human Motion Modeling: A Framework for Analysis," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 325 - 339, 2016.

[17] F. Siyahjani, S. Motiian, H. Bharthavarapu, S. Sharlemin and G. Doretto, "Online geometric human interaction segmentation and recognition," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, Chengdu, 2014.

[18] I. Mumtaz, J. Lv and J. Wei, "A novel method for online action segmentation and classification," *2015 5th International Conference on Information Science and Technology (ICIST)*, pp. 569 - 573, 2015.

[19] R. Ma and F. Hu, "An Intelligent Thermal Sensing System for Automatic, Quantitative Assessment of Motion Training in Lower-Limb Rehabilitation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1 - 9, 2017.

[20] F. Zhou, F. D. l. Torre and J. K. Hodgins, "Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 582 - 596, 2013.

[21] K. D. Mankoff and T. A. Russo, "The Kinect: a low-cost, highresolution, short-range 3D camera," *Earth Surf. Process. Landforms*, vol. 38, no. 9, p. 926–936, 2013.

[22] J. Han, L. Shao, D. X. S. Member and J. Shotton, "Enhanced Computer Vision with Microsoft Kinect Sensor : A Review," vol. 43, no. 5, p. 1318–1334, 2013.

[23] L. Jaemin, H. Takimoto, H. Yamauchi, A. Kanazawa and Y. Mitsukura, "A robust gesture recognition based on depth data," in *The 19th Korea-Japan Joint*

Workshop on Frontiers of Computer Vision, 2013.

[24] E. A. T. M. a. J. I. M. N. Kitsunezaki, "KINECTapplications for the physical rehabilitation," *MeMeA 2013 - IEEE Int.Symp. Med. Meas. Appl. Proc.*, p. 294–299, 2013.

[25] M. D. a. C. D. F. Desobry, "An Online Kernel Change Detection Algorithm," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2961-2974, 2005.

[26] Y. Xue, X. Mei, J. Bian, L. Wu and Y. Ding, "Temporal segmentation of facial expressions in video sequences," in *2017 36th Chinese Control Conference (CCC)*, 2017.

[27] M. A. Simão, P. Neto and O. Gibaru, "Unsupervised gesture segmentation of a real-time data stream in MATLAB," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016.

[28] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. Fifth Berkeley Symp. Math. statistical probability*, 1967.

[29] D. Gong, G. Medioni and X. Zhao, "Structured Time Series Analysis for Human Action Segmentation and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1414 - 1427, 2014.

[30] Y. G. a. B. K. I.S. Dhillon, "Kernel k-Means: Spectral Clustering and Normalized Cuts," in *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2004.

[31] R. C. a. L. Davis, "“Robust Real-Time Periodic Motion Detection, Analysis, and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781-796, 2000.

[32] H. J. a. C. F. B.-K. Yi, "Efficient retrieval of similar time sequences under time warping," in *Proceedings 14th International Conference on Data Engineering*, 1998.

[33] K.-I. N. M. N. a. S. S. H. Shimodaira, "Dynamic Time-Alignment Kernel in Support Vector Machine," in *Proc. Neural Information Processing Systems*, 2001.

[34] M. Cuturi, J.-P. Vert, O. Birkenes and T. Matsui, "A Kernel for Time Series Based on Global Alignments," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, II-413 - II-416.

[35] J.-P. V. O. B. a. T. M. M. Cuturi, "A kernel for Time Series Based on Global Alignments," in *Proc. IEEE Int'l Conf.Acoustics, Speech, and Signal Processing*, 2007.

[36] V. D. a. O. K. M. Ostendorf, "From HMM's toSegment Models: A Unified View of Stochastic Modeling for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 5, pp. 360-378, 1996.

[37] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18 - 28, 1998.

[38] S. Dumais, "Using SVMs for text categorization," in *IEEE Intelligent Systems, 1998* .

[39] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition.".

[40] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant features," *Machine learning: ECML-98*, 1998.

[41] S. Dumais, J. Platt, J. Platt and M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," *CIKM'98,* pp. 148-155, 1998.

[42] J. a. C. W. 1. Weston, "Support vector machines for multi-class pattern recognition.," *Proc. European Symposium on Artificial Neural Networks,*, pp. 219-224.

[43] K. a. Y. S. 2. Crammer, "On the algorithmic implementation of multiclass kernel-based machines," *JMLR,* pp. 265-292, 2001.

[44] P.-H. C.-J. L. a. B. S. Chen, "A tutorial on v-support vector machines.," *Applied Stochastic Models in Business and Industry 21,* pp. 111-136, 2005.

[45] I. T. J. T. H. a. Y. A. Tsochantaridis, "Large margin methods for structured and interdependent output variables," *JMLR 6,* pp. 1453-1484, 2005.

[46] A. Berger, "Error-Correcting Output Coding for Text Classification," in *IJCAI'99: Workshop on Machine Learning for Information Filtering,* 1999.

[47] G. &. H. T. James, "The error coding method and PiCT," *Journal of Computational and Graphical Stastistics,* vol. 7, no. 3, p. 377–387, 1998.

[48] G. B. Thomas G. Dietterich, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research,* 1995.