Model Based Gait Recognition Using Skeletal Data

Authors

Md. Bakhtiar Hasan (144401)

Naeer Jawad Amin (144403)

Supervisor

Dr. Md. Hasanul Kabir

Professor

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of
Bachelor of Science in CSE

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Gazipur, Bangladesh

October 2018

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Md. Bakhtiar Hasan and Naeer Jawad Amin under the supervision of Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

- -

Md. Bakhtiar Hasan
Student ID: 144401

Naeer Jawad Amin
Student ID: 144403

Supervisor:

- -

Dr. Md. Hasanul Kabir
Professor,
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

Being the very first in the category of low-cost consumer-level depth sensors, Microsoft Kinect has opened the door to a new generation of computer vision and biometric security applications after its release. This thesis focuses on designing new methodologies for Kinect-based gait recognition systems that utilize the Kinect 3D virtual skeleton to construct effective and robust motion representations. Our goal is to propose a gait recognition method that focuses on designing a feature descriptor that can capture person-specific distinct motion patterns, caused by the influence of human physiology and behavioral traits. In this regard we use pre-existing representations of skeletal data namely Joint Relative Distance and Joint Relative Angle. The proposed methodologies contain more representations using mean and standard deviation of the data which can effectively handle view and pose variations. We used a dynamic time warping-based kernel that takes a collection of sequences as parameters and computes a dissimilarity measure between the training and the unknown sample. It can effectively handle variable walking speed without any need of extra pre-processing. The effectiveness of the proposed methodologies are evaluated using 3D skeletal gait database captured with a Kinect v1 sensor. In our experiments, fusion of mean and standard deviation achieves promising results, as compared against the previous implementations.

Acknowledgement

It is an auspicious moment for us to submit our thesis work by which are eventually going to end our Bachelor of Science study. At the very beginning, we want to express our heartfelt gratitude to Almighty Allah for his blessings bestowed upon us which made it possible to complete this thesis research successfully. Without the mercy of Allah, we would not be where we are right now.

We would like to express our grateful appreciation to Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology for being our adviser and mentor. His motivation, suggestions and insights for this thesis have been invaluable. Without his support and proper guidance, this thesis would not see the path of proper itinerary of the research world. His valuable opinion, time and input provided throughout the thesis work, from the first phase of thesis topics introduction, research area selection, proposition of algorithm, modification and implementation helped us to do our thesis work in proper way. We are grateful to him for his constant and energetic guidance and valuable advice.

We would also like to thank Mr. Faisal Ahmed, Data Scientist, City of Calgary, Canada, whose work on Gait Recognition using Kinect inspired us to take on this difficult task. Throughout the research, he extended his helping hands in every way possible.

We would like to extend our vote of thanks to all the respected jury members of our thesis committee for their insightful comments and constructive criticism of our research work. Surely they have helped us to improve this research work.

Last but not the least, we would like to express our sincere gratitude to all the faculty members of the Computer Science and Engineering department of Islamic University of Technology. They helped make our working environment a pleasant one by providing a helpful set of eyes and ears when problems arose.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first present an overview of our thesis that includes the signification of the problem and the problem statement in detail. Research challenges to be faced in the whole scenario is also discussed based on the problem statement. Thesis objectives, motivations and our contribution are noted in sections. The end of this chapter has the description of the organization of the thesis.

1.1 Overview

1.1.1 Gait

Gait is the pattern of movement of the limbs of animals, including humans, during the locomotion over a substrate. Most animals use a variety of gaits, selecting gait based on speed, terrain, the need to maneuver, and energetic efficiency. Different animal species may use different gaits due to differences in anatomy that prevent use of certain gaits, or simply due to evolved innate preferences as a result of habitat differences.

Human gait refers to locomotion achieved through the movement of human limbs. Human gait is defined as bipedal, biphasic forward propulsion of center of gravity of the human body, in which there are alternate sinuous movements of different segments of the body with least expenditure of energy. Different gait patterns are characterized by differences in limb-movement patterns, overall velocity, forces, kinetic and potential energy cycles, and changes in the contact with the surface (ground, floor, etc.). Human gaits are the various ways in which a human can move, either naturally or as a result of specialized training [1].

A (bipedal) gait cycle as shown in figure 1 [2] is the time period or sequence of events or movements during locomotion in which one foot contacts the ground to when that same foot again contacts the ground, and involves

forward propulsion of the centre of gravity. A single gait cycle is also known as a stride.

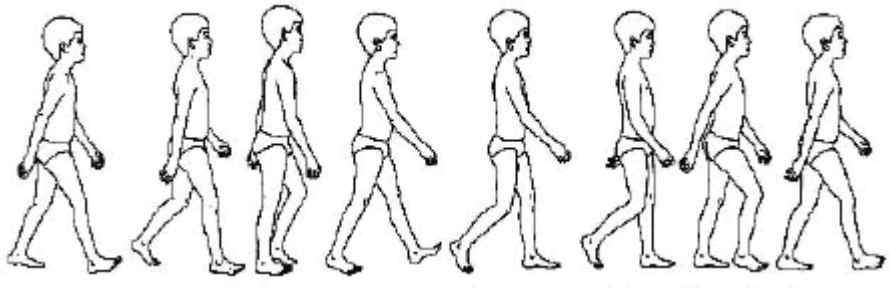

Figure 1: Human gait cycle

1.1.2 Gait Recognition

Gait recognition refers to an algorithm that essentially aims to recognize a person by automatically extracting movement characteristics of the walking person in a video [3].

Biomechanics literature denotes gait recognition procedure as, "A given person will perform his or her walking pattern in a fairly repeatable and characteristic way, sufficiently unique that it is possible to recognize a person at a distance by their gait" [4].

1.1.3 Tracking Skeletal Data

Microsoft Kinect can be used to track skeletal data. Kinect(codenamed Project Natal during development) as shown in figure 2 is a line of motion sensing input devices that was produced by Microsoft for Xbox 360 and Xbox One video game consoles and Microsoft Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/-computer without the need for a game controller, through a natural user interface using gestures and spoken commands. Kinect uses structured light and machine learning to detect skeletal data. It is a two-stage process:

- Compute depth map using structured light
- Infer body position using machine learning

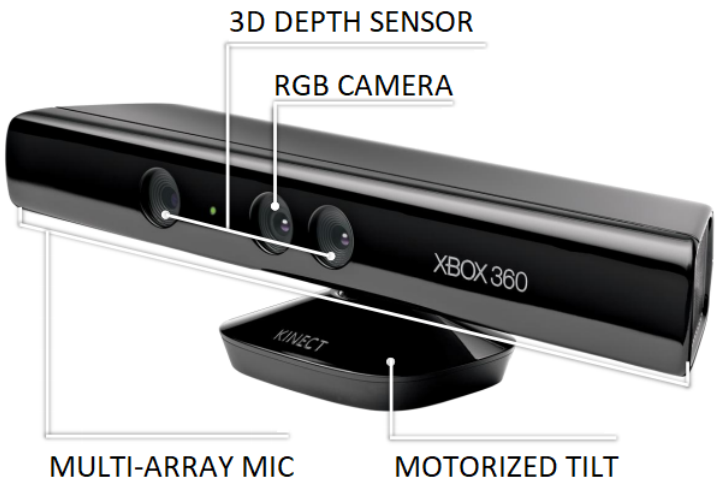

Figure 2: Kinect for XBOX 360

The depth map is constructed by analyzing a speckle pattern of infrared laser light. Microsoft licensed this technology from a company called PrimeSense. The depth computation is all done by the PrimeSense hardware built into Kinect. Details of this hardware are not publicly available. Based mostly on PrimeSense patent applications [5], the technique of analyzing a known pattern is called structured light. General principle of this approach is to project a known pattern onto the scene and inferring depth from the deformation of that pattern. Kinect uses infrared laser light, with a speckle pattern of infrared laser light. It combines structured light with two classic computer vision techniques: depth from focus, and depth from stereo.

Depth from focus uses the principle that stuff that is more blurry is further away. Kinect dramatically improves the accuracy of traditional depth from focus. It uses a special "astigmatic" lens with different focal length in x- and y-directions. A projected circle then becomes an ellipse whose orientation depends on depth. On the other hand, if we look at a scene from another angle, stuff that is closer gets shifted to the side more than stuff that is far away. This phenomenon is known as parallax. Kinect analyzes the shift of the speckle pattern by projecting from one location and observing from another.

Body parts are inferred using a randomized decision forest, learned from over 1 million training examples, mapping depth images to body parts [6].

1.2 Problem Statement

Gait recognition has recently gained more and more interests from researchers due to its several attractive properties. It is an emerging and ever changing field of technology that can be implemented into just about anything that requires a security protocol. Generic gait recognition systems typically comprise three basic components: i) a sensor or camera to collect the data, ii) a feature representation, and iii) a classifier. The feature representation describes the characteristics of the captured gait data (obtained from the sensor), which is then used by the classifier to recognize the person. However, even using the best classifier will result in poor recognition performance, if provided with features having low discriminating ability or inadequate information. Hence, designing an effective and discriminative feature representation is one of the main challenges in gait recognition. An effective and discriminative feature descriptor can be characterized as: i) having high inter-class variations and low intra-class variations, ii) providing robustness in uncontrolled environment under the presence of view and scale changes, and iii) having a low-dimensional feature space to facilitate low computational cost [7, 8]. Since gait recognition using Kinect is an emerging area of research, most of the existing studies have been conducted under simplifying assumptions, such as constant walking or movement speed, fixed distance and viewpoint, relatively smaller datasets, etc. However, in order to deploy gait recognition systems in real-world settings, constructing effective feature representations that can work in unconstrained environment is an important task.

The primary goal of our thesis is "Developing effective methodology for gait recognition using Skeletal Joint Data". In particular, we aim to investigate how the 3D skeletal motion data obtained from the Kinect can effectively be utilized to design real-world gait recognition systems. The challenges we aim to tackle are: i) finding a set of discriminative features that are robust against view and scale changes as well as the speed of walking, and ii) utilizing these features in designing effective gait recognition systems that can attain high recognition performance in uncontrolled environment.

1.3 Significance of Gait Recognition

Biometric is a technical term for body measurements and calculations and is used in computer science for identifying people by their physical characteristics. Biometrics is becoming an important field in science for several, not the least of which is the heightened demand for security in a variety of situations. Some of the biometric identification systems are very accurate, like those that use patterns of blood vessels in the retina (the back of the eye). But generally, in order for a system like that to identify us, we have to voluntarily enroll ourselves in the database. Then, when we try to gain access to a facility or to classified information, the system takes a scan of our retina and matches it to that stored file to verify our identity. This works great when we are trying to selectively identify a small group of people with special access privileges. But we cannot positively identify anybody who is not enrolled in the system, we can tell who they are not but not who they are. Furthermore, we have to generally interact closely with these systems in order for them to check our identity. Someone with malicious intentions would obviously avoid this type of identification system. So if we want to identify dangerous or suspicious individuals from a distance, these systems would not help.

Secure real-time user authentication and access control management are crucial for a wide variety of systems and services, such as secure access to facilities, cell phones, automated teller machines (ATM), computers, etc. Traditional approaches to the problem involve use of certain tokens, such as identification card or password verification. However, these solutions have some drawbacks. For example, ID cards can be lost, stolen, or forged, while passwords can be compromised or forgotten [9]. As a result, these approaches are vulnerable to forgery and unable to provide sufficient security [10]. In recent years, rapid development of biometric technologies has opened the door to a new class of fast and reliable identity management solutions [11], which are being actively researched and deployed in both corporate and academic settings [12, 13, 14].

Human biometric traits can roughly be divided into two categories: physio-

logical and behavioral. Physiological biometric systems utilize certain physical characteristics, such as face, iris, ear, fingerprints, palmprints, etc. for individual recognition. On the other hand, behavioral biometric systems rely on human behavior-mediated activities, such as gait, voice, handwriting, signature, etc. In most cases, a biometric system requires direct participation or cooperation of the persons (in the form of physical contact or pose) being recognized [15, 16]. However, gait recognition is one kind of biometric technology that can be used to monitor people without their cooperation. Gait can be defined as the movement patterns of certain body limbs and their interactions with the surrounding environment during walking [17, 18]. It is a complex and dynamic behavioral trait, which makes it difficult to disguise someone's own gait or imitate some other person's gait [19]. This characteristic makes gait recognition particularly useful in scenarios where other biometric traits are obscured (often intentionally, such as a crime scene) or user cooperation is not intended (such as surveillance in public places like airports, bus stations etc.) [19]. In addition, gait analysis can potentially be utilized in virtual and augmented reality, motion and video retrieval [20], 3D human body and animation [21, 22], healthcare [23].

1.4 Research Challenges

The developed methodologies should satisfy the following criteria:

1. The constructed feature descriptors should be view and scale-invariant and should effectively represent the human gait patterns in a robust manner.

2. To ensure low computational cost, the developed methodologies should incorporate effective feature selection techniques to obtain highly discriminating feature representations in a low-dimensional space.

3. The developed methodologies should not require individuals to walk only to a specific direction (fronto-normal or fronto-parallel).

4. The developed methodologies should be robust against variations in the speed of walking resulting in variable length video sequences.

5. Fusion of disparate feature representations should be experimented to check whether they improve the gait recognition performance.

1.5 Contributions

In this thesis, we presented new methodologies for gait recognition that utilizes the 3D skeletal motion data captured using the Kinect depth sensor. The strength of the proposed methods lies in constructing view and scale-invariant feature representations that can effectively capture the underlying spatio - temporal motion patterns of different skeletal joints. A brief overview of the contributions of this thesis is as follows:

1. We managed to obtain better performance using less features than other state of the art methods that were compared. The result is better in scale and view variant situations. The performance is similar or better in situations where the distance from the camera and user remain same and user walks in a single direction.

2. New feature named angle difference is introduced for gait motion representation. This is robust against view and scale variations and along with calculated angles can effectively capture the underlying motion patterns of different skeletal joint-pairs.

3. Old feature representations methods like mean and standard deviation have been rejuvenated to be combined and used with our feature representations.

4. Dynamic time warping (DTW)-based classifier is introduced for gait classification that can effectively handle the differences in walking speed, thus eliminating the need of extra pre-processing steps such as resampling.

5. More robustness against noise compared to other methodologies. This also contributed to the elimination of pre-processing steps such as noise removal or gait cycle detection.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 gives an overview of different approaches for the gait recognition problem. This chapter also describes the reasons for choosing angles and mean, standard deviation for gait recognition.

Chapter 3 proposes a solution to increase accuracy and robustness against view and scale variant data. It contains the framework, implementation of the proposed methodologies and also contains other methodologies that we tested.

Chapter 4 presents result analysis and comparison with other implementations and studies.

Chapter 5 presents conclusions and discusses future work.

Chapter 2

Literature Review

The first section of this chapter gives a brief overview of the Microsoft Kinect sensor with a discussion on some of the precursory works related to range sensing. Next, we present a generic classification of different types of existing gait recognition approaches and provide justifications for selecting model-based marker-less approaches as the primary focus of our thesis. In the subsequent sections, we explore existing approaches to human gait recognition including some of the most recent Kinect-based methods. We also discuss the limitations of some of these methods. Finally we will discuss the classification methods used in these papers.

2.1 Background

From the advent of computer vision in the early 1960s, researchers are investigating how computers can acquire a realistic interpretation of the complex three-dimensional world around us. One of the main objectives is to enable computers to make sense of a complex environment with the help of sensory input processing, which in turn, has the potential to be utilized in situation-aware intelligent surveillance and access control systems in a natural and unobtrusive manner. Early computer vision methods were mostly based on 2D intensity images, where mathematical techniques were used to model 3D shapes and structures. However, recovering 3D information from 2D intensity images is a challenging task, since projecting a 3D scene to a 2D space incurs significant loss of data. In fact, it is mathematically impossible to construct a 3D representation of an object given only a single 2D intensity image [24]. Hence, researchers are actively seeking new sensor technologies in order to recover 3D shapes directly from the sensor, leading to more robust representations and interpretations of 3D environments.

Early range sensors introduced in 1980s were typically based on sonar, infrared, and laser range finders [25]. One of the precursory works on laser

range sensors presented by Gil et al [26]. They addressed the problem of combining range and intensity data obtained from the same scene by extracting edge maps from both representations and reducing the problem to combining the two edge maps. Magee and Aggarwal [27] presented an extensive review on 3D object description and recognition based on both intensity and laser-based range imagery. They argued that combining the advantages of these two modalities could potentially lead to more robust and computationally inexpensive interpretation and recognition of 3D objects and structures. This argument was further supported by the study presented in [28], where intensity information was utilized to reduce the time required for range sensing. Instead of finding the range for every point in a scene, the authors used intensity image to guide selective range sensing by first detecting potential points of interest. Another work by Vemuri et al [29]. They utilized intrinsic surface properties extracted from range data to construct 3D surface and object representations. However, while the range sensors opened the door to a new class of computer vision techniques, early sensors suffered from several limitations. For example, sonar sensors are susceptible to noise caused by echo and reflections. On the other hand, early infrared and laser range finders were expensive and could only estimate the range of a single point in a scene. In addition, typically these sensors were not suitable for capturing human motion data [24].

Figure 3: Different data streams obtained from Kinect Sensor

Being the very first sensor in the category of low-cost consumer-level depth sensing, the recent release of the Microsoft Kinect has potentially opened the door to a new generations of computer vision and biometric security applications [15]. It was originally introduced for the Xbox 360 gaming system as an add-on device that can detect physical movements or voice commands of the

user and thus enable the user to play games without any physical controller [12]. Kinect is made up of an array of sensors, which include i) a color camera, ii) a depth sensor, and iii) a multi-array microphone setup as shown in figure 3. The depth sensor comprises a monochrome CMOS camera and an infrared (IR) emitter. Using these two components, Kinect can build 3D maps of objects by emitting human eye-invisible IR and then analyzing the light and shadow of the image captured by the CMOS camera. The multi-array microphone has an ambient noise cancellation feature and can also be used to detect the source location of voice. In addition, Kinect can construct a 3D virtual skeleton of a human body using the depth information [6]. With all these capabilities along with its small compact size, the Microsoft Kinect has attracted a significant attention from the computer vision, biometrics, and robotics research community, leading to its application in home monitoring [30], face and facial expression analysis [31], 3D object modeling [32], indoor navigation and mapping [33], healthcare and rehabilitation [34], etc. In addition, some of the recent works on pose estimation [35], human body modeling [21, 22], motion retrieval [36], and activity recognition [37], etc have utilized the depth information and computationally inexpensive 3D skeletons obtained from Kinect.

2.2 Overview of Gait Recognition Methods

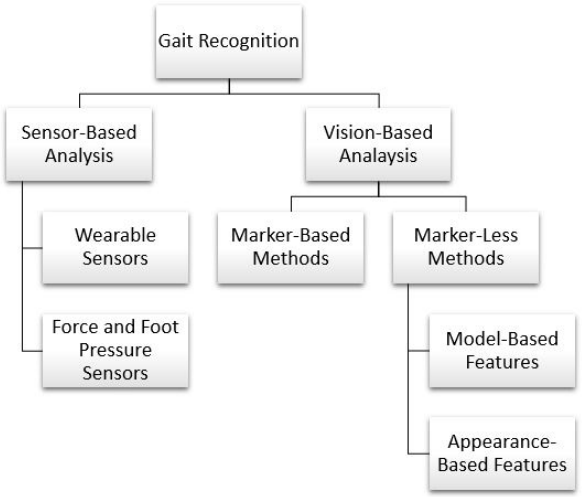

Figure 4: Categories of gait recognition methods

Human gait recognition techniques can be divided into two main categories: i) sensor-based approach and ii) vision-based approach. This section presents a brief overview of these methods.

2.2.1 Sensor-Based Methods:

Sensor-based techniques typically exploit wearable motion sensors attached to different joints or body parts of the subject to measure various characteristics of gait performed by the subject [38]. Some commonly used wearable sensors include accelerometer, gyroscope, magneto-resistive sensors, flexible goniometer, electromagnetic tracking system (ETS), electromyography (EMG) sensor, etc. [38]. Force sensors and pressure plates that can measure foot pressure have also been successfully applied in human gait analysis [39]. However, although sensor-based techniques can acquire reliable gait data, applications are limited to diagnosis of medical conditions and rehabilitation research, typically conducted under controlled laboratory environment [40].

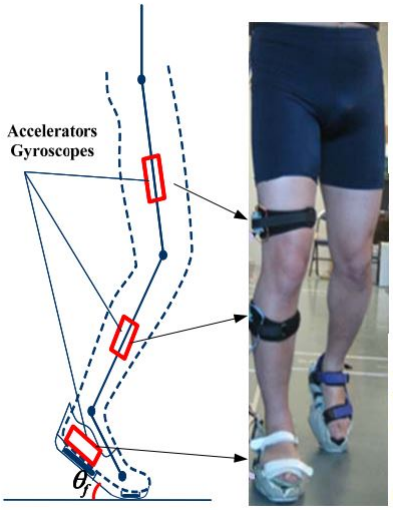

Figure 5: Kinematic measurement based on accelerators and gyroscopes attached on the foot, calf and thigh separately

2.2.2 Vision-Based Methods:

While sensor-based gait recognition techniques exploit motion and pressure sensors, vision or image-based systems utilize videos of gait recorded using a single or multi-camera setup in indoor or outdoor environment. The recorded

video data is then processed to extract salient characteristics related to human gait motion. This category of gait recognition systems can further be subdivided into marker-based and marker-less analysis. In marker-based approach, active or passive markers are attached to body parts of the subject, which facilitates extraction of accurate joint motion data from the video without extensive video processing. On the other hand, in marker-less gait analysis, videos are recorded with normal clothing with no marker attached. Different computer vision and image processing techniques are then applied on the recorded videos to extract human silhouette and motion data.

In our thesis, we focused on vision-based marker-less gait recognition methods. This choice was motivated by the social acceptability of vision-based approaches, which is well-demonstrated by the widespread deployment and general acceptance of video surveillance systems in public places like airports, banks, bus and train stations, etc. On the other hand, sensor-based approaches are typically difficult to accommodate in many real-world scenarios.

2.3 Vision-Based Gait Recognition Systems

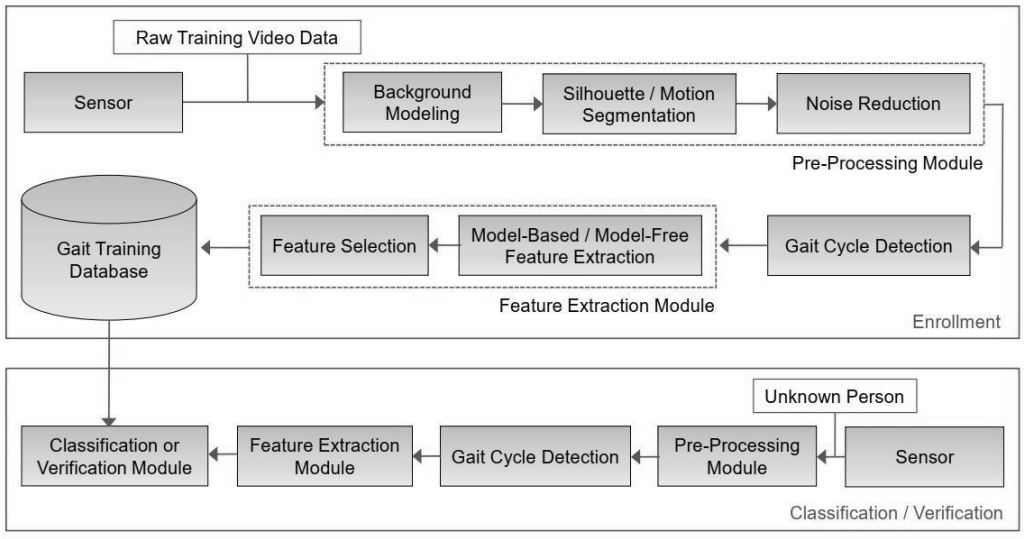

Figure 6: Overview of a generic vision-based gait recognition system

A vision-based gait recognition system involves multistage processing of video data acquired from a single or multi-camera setup. Figure 6 shows an overview of a generic video camera-based gait recognition system. As shown

in this figure, the first step involves pre-processing the raw sequence images obtained from the camera to isolate human silhouette or motion. Typically, this is achieved by modeling the background scene using a set of sequence images and highlighting the difference in the current image with respect to the background. To facilitate robust background modeling, traditional gait data capturing is performed with a fixed camera position in an environment where the background is relatively uniform [41]. This approach of foreground segmentation is often followed by noise reduction methods to reduce noise and distortion caused by sub-optimal threshold selection [42]. Regular human walking is considered to be a cyclic motion, which repeats in a relatively stable frequency [43]. Hence, the next step involves detecting gait cycles since features extracted from a single gait cycle can represent a complete gait pattern. Once the gait cycle is detected, the corresponding silhouette or motion data is passed to a feature extraction module which extracts the underlying characteristics of individual gait. Success of a gait recognition system critically depends on the discriminating ability of the extracted feature representation. Hence, it is important to remove redundant and noisy features and utilize only the most informative and discriminating features to construct the final gait signature representation. Lastly, the constructed feature representation is passed to a recognition or verification module that uses machine learning techniques to match unknown gait samples with the training gait models stored in a database.

Depending on the type of features being used, vision-based gait recognition methods found in literature can be divided into two categories: i) model-based approaches and ii) model-free or appearance-based approaches [44]. In the following subsections, we present brief reviews of the past works in these two categories. In addition, we also present a review of some of the most recent works on Kinect-based gait recognition.

2.3.1 Model-Based Gait Recognition

In model-based approaches, movements of different body parts, such as legs, arms, etc. are modeled explicitly based on a set of estimated parameters [43]. The variations of the parametric values are tracked over time, which is then

used as the gait signature representation. However, constructing the model, fitting it on the captured gait data, and estimating the parametric values are computationally expensive, which makes model-based gait recognition approaches time-consuming and difficult to accommodate in many real-world applications [43]. BenAbdelkader et al. [45] proposed one of the early model-based gait recognition methods, where two spatio-temporal parameters, namely cadence and stride length were estimated to represent the gait biometric. Later, Urtasun and Fua [46] utilized 3D temporal motion model-fitting to synchronized video sequences in their proposed gait recognition method. Individual gait signature was represented based on the estimated motion parameters. A similar approach was adopted by Yam et al. [47], where gait signatures obtained from walking and running was differentiated by modeling human leg structure and motion. Although their proposed gait recognition method offers view and scale invariance, it depends heavily on the quality of the gait sequences [48].

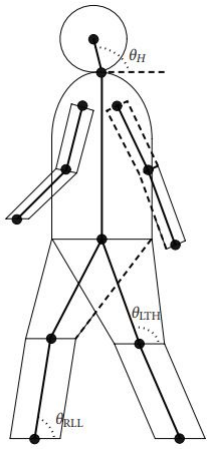

Figure 7: The layered deformable model

More recently, Lu et al. [49] introduced layered deformable models (LDM) (shown in figure 7) to represent shapes and dynamics of different human body parts based on 22 different parameters. The parameters were used to capture the size, position, and orientation of the body parts from fronto-parallel gait sequences. While many of the existing model-based gait recognition methods focus on modeling the lower-body parts, the LDM models were applied to construct a full-body representation of gait. Another full-body gait analysis approach presented by Arai and Andrie [50] utilizes morphological oper-

ations to obtain skeletal models from extracted silhouettes. Discrete wavelet transformation (DWT) and Haar wavelets were applied on the extracted models to reduce feature dimensionality. However, morphological operation-based skeleton extraction is not invariant against view and scale changes.

2.3.2 Appearance-Based Gait Recognition

While model-based approaches focus on modeling individual body parts and their movements, model-free approaches involve constructing a compact holistic representation of gait motion appearance by utilizing the silhouette sequences extracted from the video [43]. Shutler et al. [51] introduced velocity moment features to represent object and motion in image sequences for gait analysis. In practice, the velocity moments capture the differences between the center of mass of a moving object in successive images.

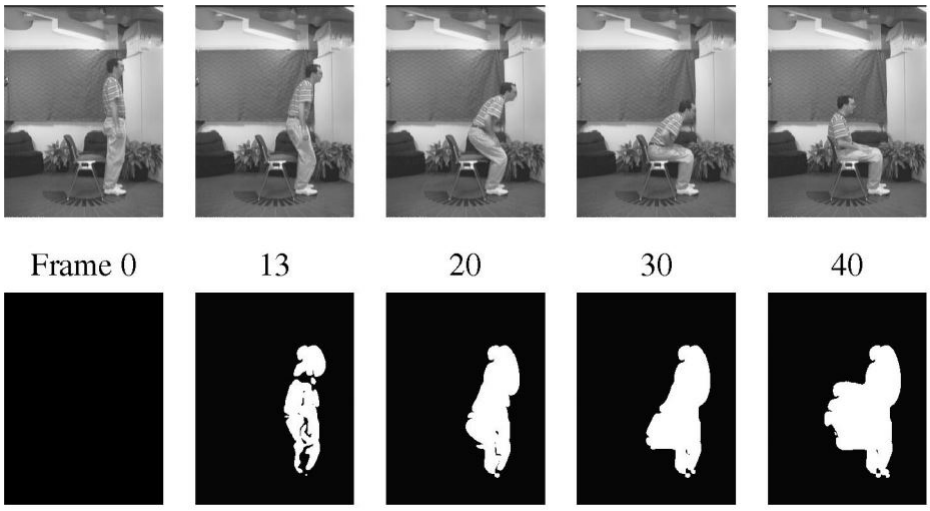

Figure 8: Key frames containing Motion Energy Image (MEI) of a person sitting

Later, Bobick and Davis [52] proposed the motion energy image (MEI), which is a temporal template representation of human movement. As shown in figure 8, the MEI representation comprises a static vector image, where each point is a function of the motion attributes of the corresponding spatial location of the point in a sequence image. BenAbdelkader et al. [53] utilized self-similarity plots constructed from pairwise correlation of extracted silhouettes in the image sequences. The obtained plots were then projected into a subspace, namely

the EigenGait space using principal component analysis (PCA), which effectively reduces the feature dimensionality. Both the motion energy image (MEI) and the EigenGait based representation of human movement for gait analysis contributed to the development of one of the most popular appearance-based gait recognition methods, namely the gait energy image (GEI) [54].

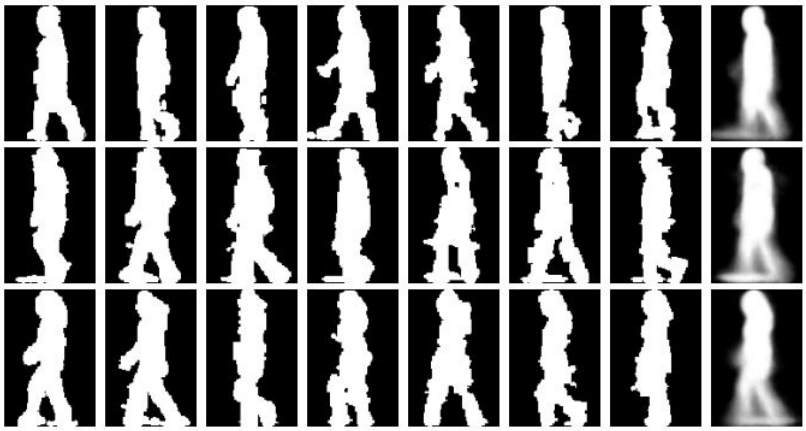

Figure 9: Examples of normalized and aligned silhouette frames in different human walking sequences. The rightmost image in each row is the average silhouette image over the whole sequence - Gait Energy Image (GEI)

As shown in figure 9, the GEI is a spatio-temporal representation of all the silhouette motion sequences accumulated in a single energy image. The GEI method utilizes a fusion of principal component analysis (PCA) and multiple discriminant analysis (MDI) [55] to reduce the feature dimensionality, while maintaining a high class separability at the same time.

Some of the more recent model-free gait recognition methods focus on extending GEI to a more robust representation. One example is the frame difference energy image (FDEI) proposed by Chen et al. [56], which handles silhouette incompleteness by utilizing denoising and clustering techniques. Another approach proposed by Li and Chen [57] involves fusing foot energy image (FEI) and head energy image (HEI), which facilitates the construction of a more informative gait signature representation. Nevertheless, although appearance-based approaches present a computationally inexpensive set of gait recognition methodologies, their performance suffer due to scale and view variations in an uncontrolled or changing environment [15].

2.3.3 Gait Recognition Using Kinect

While vision-based biometric gait recognition has been a topic of interest over the past twenty years, the invention of the low-cost Kinect sensor has opened up new opportunities to address the problems related to real-time motion analysis, resulted in a spike in the interest in gait recognition using Kinect. In addition to different data streams that can be obtained from Kinect (RGB, depth, audio), it can also construct a 3D virtual skeleton from human body and track it in real-time rendering the traditional video pre-processing tasks (e.g. background modeling, silhouette extraction, etc.) unnecessary. As a result, some of the recent gait recognition methods found in literature utilize the computationally inexpensive real-time depth sensing and skeleton tracking in order to model the gait signature.

One of the precurosry work on Kinect-based gait analysis was done by Ball et al [18], whose study investigated the possibility of recognizing individual persons from their gait pattern using three-dimensional 'skeleton' data from an inexpensive consumer-level sensor, the Microsoft Kinect. The skeletal information was extracted using Kinect SDK and suitable walk half-cycles were subsequently extracted by hand. A walk cycle is defined here as the movement where the person's feet are a maximum distance apart, with the same foot in front. Due to the limited distance over which skeletons could successfully be extracted, walk half-cycles were used rather than full walk cycles as this choice yielded many more data sets. The features used for clustering were based on the lower limb joint angles. It was found that the limb length assigned by the Microsoft Kinect SDK skeletal algorithm changed significantly as the person walked across the camera field of view, so the limb length was not scale and view invariant. The features selected for clustering were the mean, standard deviation and maximum value of three angles for each of the left and right legs of the extracted skeleton. The angle of the upper leg relative to the vertical, the angle of the lower leg relative to the upper leg, and the angle of the foot relative to the horizontal were used, giving a total of 18 angular features. They used K-means clustering algorithm. A combined clustering accuracy of 43.6% was obtained.

Preis et al [58] presented an approach for gait recognition using Microsoft Kinect by evaluating a number of body features together with step length and speed. The Kinect SDK offers the detection and tracking of 20 different skeletal points, from head over hips to the feet. Using these points, thirteen biometric features were calculated for person identification. The height, the length of legs, torso, both lower legs, both thighs, both upper arms, both forearms, the step length, and the speed. The first eleven features are static and the rest two are dynamic. The performance of the chosen features were evaluated with the help of three classifiers: 1R, C4.5 decision tree and a Naïve Bayes Classifier. The accuracy of the identification are $62.7\%, 76.1\%$ and 85.1% respectively. However, as shown in figure 10, the users kept a certain distance from the sensor while walking and walked in a specific direction.

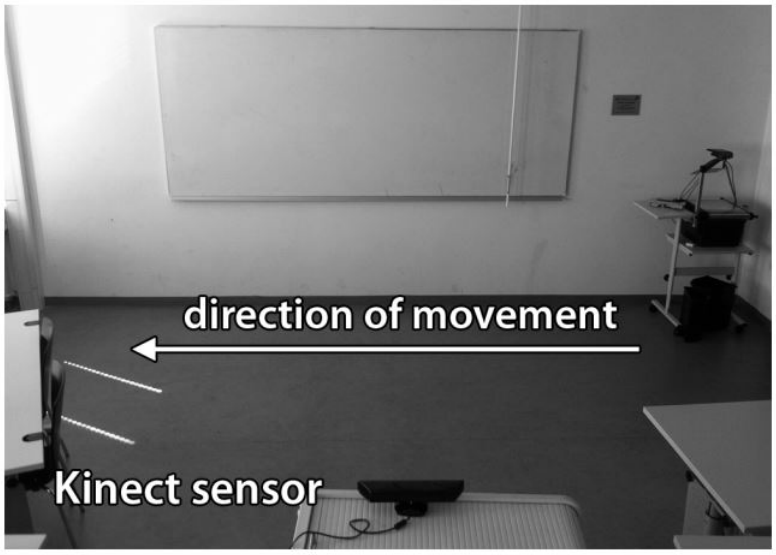

Figure 10: Experimental setup for Preis et al.

Faisal et al [15] proposes a new 3D gait recognition method that utilizes the Kinect skeletal data for gait representation. It proposes two new features, namely joint relative distance (JRD) and joint relative angle (JRA), which are robust against view and pose variations. There were several steps in this proposed method. At first, complete gait cycles were detected (shown in figure 12) from video sequences recorded by the Kinect Sensor.
Next, the relevant JRDs and JRAs (shown in figure 11) were computed over the complete gait cycle. Then, these variable length JRD and JRA sequences

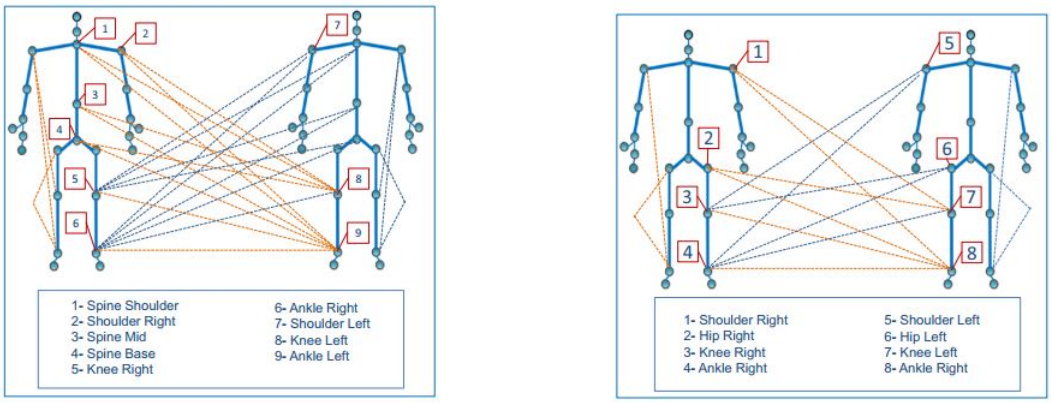

Figure 11: A graphical representation of the selected best JRDs and JRAs

were then matched with an unknown sample sequence. The sequences with the lowest dissimilarity were selected and then a rank level fusion of the selected JRD and JRA sequences was carried out. Gait cycle was detected by tracking the horizontal distance between `ANKLE_LEFT` and `ANKLE_RIGHT` joints over time. During the walking motion, the distance between the two ankle joints will be the maximum when the right and the left legs are farthest apart (heal strike) and will be the minimum when the legs are in the rest (standing) position. Therefore, by detecting three subsequent maxima, it is possible to find the two subsequent occurrences of the same leg in the heal strike position, which corresponds to the beginning and ending points of a complete gait cycle, respectively [59]. Joint relative distance (JRD) between any two skeletal joints $p_1(x_1, y_1, z_1)$ and $p_2(x_2, y_2, z_2)$ can be defined as the the Euclidean distance between these two joints in a 3D space [60]. It can be denoted as:

$$\delta(p_1, p_2) = sqrt(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \tag{1}$$

The Joint Relative Angle (JRA) between two joints p_1 and p_2 can be defined as the angle formed by p_1 and p_2 with respect to a reference point r. Given the coordinates of 3 points p_1, p_2 and r in a 3D space, the angle:

$$\theta_{p_1,p_2} = cos^{-1} \frac{\vec{p_1 r} \times \vec{rp_2}}{|\vec{p_1 r}| \times |\vec{rp_2}|} \tag{2}$$

`SPINE_BASE` was selected as the reference point in this study, since this joint remains almost stationary during walking. A total of 300 JRD or JRA sequences

was possible as there were 25 skeletal joints. Genetic algorithm was used to find the relevant JRD and JRA sequences. Dynamic Time Warping (DTW) classifier was utilized to design a kernel for gait recognition that takes a collection of JRD/JRA time series data originated from different joint-pairs as the parameter and outputs the dissimilarity measure between two given gait samples. The JRD/JRA sequences with the lowest dissimilarity measure was selected. Use of DTW in this case allows the alignment of different length JRD/JRA sequences, which enables the classifier to match gait samples without any intermediate resampling stage. Then a rank level fusion of JRD and JRA was done. Both JRD and JRA were used individually to generate rank lists of candidates and the top 3 candidates from both rank lists were selected for majority voting. The candidate with the highest number of votes was selected as the match. Accuracy of identification was 92.1%.

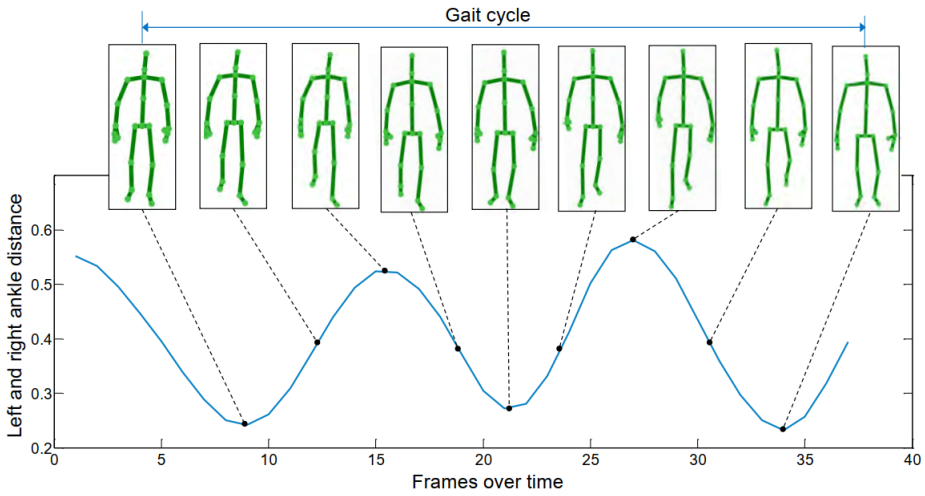

Figure 12: Detection of complete Gait Cycle by tracking the distance between the left and right ankle joints

Another one by Faisal et al [16] introduces a new 3D skeleton-based gait recognition method for motion captured by a low-cost consumer level camera, namely the Kinect. A new representation of human gait signature was proposed based on the spatio-temporal changes in relative angles among different skeletal joints with respect to a reference point. JRA sequences originated from different joint pairs were then evaluated to find the most relevant JRAs for gait description. It used similar method as discussed before to detect gait cycle

(shown in figure 12) and calculate JRA. A statistics based relevant joint pair selection approach was used that utilizes histogram of JRA features to evaluate the level of engagement of the corresponding joint pair. For joint pairs that has high relative motion during gait, the joint relative angles computed over the full gait cycle should have high temporal changes. On the other hand, joint pairs that remains stationary or moves little during gait should have little variation of JRA over the full gait cycle. This can also be represented using histogram of JRA values. For a particular joint pair that has high relative motion during gait, the histogram should have a wide distribution. On the other hand, for joint pairs that has little relative movement the JRA values will occupy only a few number of bins in the histogram. The more the number of occupied bins in the JRA histogram of a particular joint pair, the more relevant is that joint pair. Again DTW was utilized to design a kernel for gait recognition that takes a collection of selected JRA time series data originated from different joint-pairs as the parameter and outputs the dissimilarity measure between two given gait samples. The JRA sequence with the lowest dissimilarity measure was selected. Use of DTW in this case allows the alignment of different length JRA sequences, which enables the classifier to match gait samples without any intermediate resampling stage.

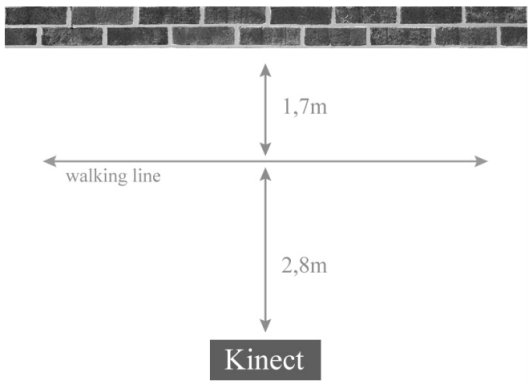

Figure 13: Experimental setup for Dikovski et al

Dikovski et al [61] the feature sets were constructed and evaluated with the purpose of finding out the role of different types of features and body parts in the gait recognition process. Here the distance between camera and the user remained same and the user walked in a specific direction (shown in figure

13). The feature sets were constructed from skeletal images in three dimensions made with a Kinect sensor. Distance between each pair of adjacent joints (19 in total), person height and distance between two ankle joints were calculated using Euclidean distance. The height was calculated as the sum of the following distances between joints

$$\begin{aligned} Height = (&d(l\_ankle, l\_knee) + d(l\_knee, l\_hip) + d(r\_ankle, r\_knee) \\ &+ d(r\_knee, r\_hip))/2 + d(c\_hip, spine) + d(spine, c\_shoulder) \\ &+ d(c\_shoulder, head) \end{aligned} \tag{3}$$

9 joint triplets were considered. They were Head - CenterShoulder - CenterHip, LeftWrist - LeftElbow-LeftShoulder, RightWrist - RightElbow - RightShoulder, LeftAnkle - LeftKnee - LeftHip, RightAnkle - RightKnee - RightHip, LeftHip - RightHip - LeftKnee, LeftHip - RightHip - RightKnee, LeftShoulder - RightShoulder - LeftElbow, LeftShoulder - RightShoulder - RightElbow. Angle between triples of joints was calculated. Angle between joints i, j and k was calculated in the following way:

$$A = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{4}$$

$$B = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2} \tag{5}$$

$$C = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \tag{6}$$

$$\theta = cos^{-1}\frac{B^2 - A^2 - C^2}{2AC} \tag{7}$$

The angle of rotation between the line of the shoulder joint points and the line of the hip joint points was added. Next the distances between the centroid of the hip, shoulder and spine joint points, and both arm (shoulder, elbow and wrist joint points) and leg (hip, knee and ankle joint points) centroids. A centroid of N joints was calculated as:

$$C = \frac{\sum_i^N \{x_i, y_i, z_i\}}{N} \tag{8}$$

Finally, all these features were aggregated for the whole gait cycle and the

mean, standard deviation, minimum, maximum and mean difference between subsequent frames were calculated. Using these feature vectors, 7 different datasets were generated. SMO, J48 and MLP was performed. MLP provided 89.80% accuracy on dataset 3.

2.4 Classifiers

In this section, we will discuss the classifiers used in different methods.

2.4.1 k-means Clustering

k-means clustering[62] is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

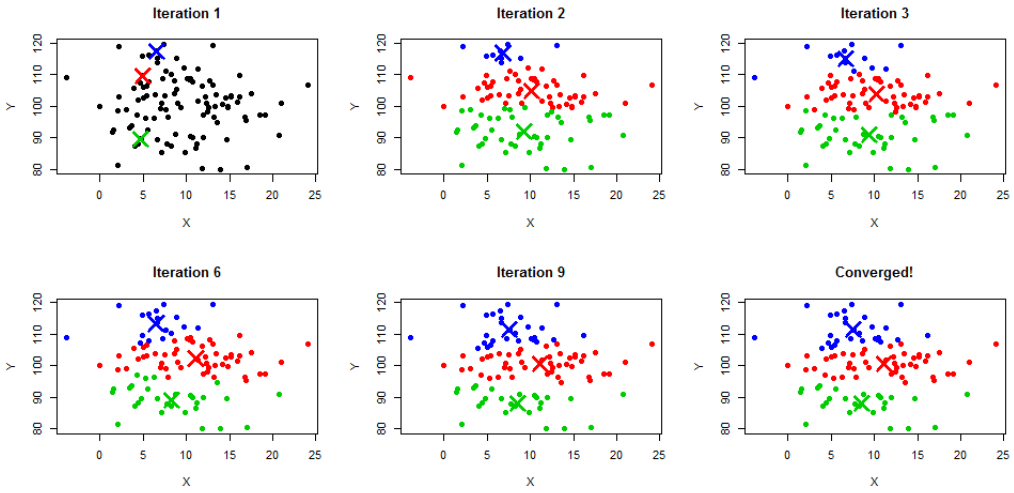

Figure 14: k-means clustering

The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. Additionally, they both use cluster centers to model the data; however, k-means cluster-

ing tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means due to the k in the name. One can apply the 1-nearest neighbor classifier on the cluster centers obtained by k-means to classify new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k(\leq n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$arg\,min \sum_{i=1}^{k} \sum_{x \in S_i} |x - \mu_i|^2 = arg\,min \sum_{i=1}^{k} |S_i| Var\, S_i \tag{9}$$

where μ_i is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$arg\,min \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y \in S_i} |x - y|^2 \tag{10}$$

The equivalence can be deduced from the identity

$$\sum_{x \in S_i} |x - \mu_i|^2 = \sum_{x \neq y \in S_i} (x - \mu_i)(\mu_i - y) \tag{11}$$

Because the total variance is constant, this is also equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares, BCSS), which follows easily from the law of total variance.

2.4.2 1R Classifier

The 1R procedure[63] for machine learning is a very simple one that proves surprisingly effective on the standard datasets commonly used for evaluation.

This paper describes the method and discusses two areas that can be improved: the way that intervals are formed when discretizing continuously-valued attributes, and the way that missing values are treated. Then we show how the algorithm can be extended to avoid a problem endemic to most practical machine learning algorithms – their frequent dismissal of an attribute as irrelevant when in fact it is highly relevant when combined with other attributes.

Like other empirical learning methods, 1R takes as input a set of examples, each with several attributes and a class. The aim is to infer a rule that predicts the class given the values of the attributes. The 1R algorithm chooses the most informative single attribute and bases the rule on this attribute alone. The basic idea is:

```
For each attribute a, form a rule as follows:
    For each value v from the domain of a,
        Select the set of instances where a has value v
        Let c be the most frequent class in that set
        Add the following clause to the rule for a:
        If a has value v then the class is c
    Calculate the classification accuracy of this rule
Use the rule with the highest classification accuracy
```

The algorithm assumes that the attributes are discrete. If not, then they must be discretized.

Any method for turning a range of values into disjoint intervals must take care to avoid creating large numbers of rules with many small intervals. This is known as the problem of "overfitting", because such rules are overly specific to the data set and do not generalize well. Holte achieves this by requiring all intervals (except the rightmost) to contain more than a predefined number of examples in the same class. Empirical evidence led him to a value of six for datasets with large numbers of instances and three for smaller datasets (with less than about 50 instances) [64].

Missing values are handled in the algorithm by treating them as a separate value in the enumeration of an attribute Missing values are treated by Holte's system as a separate value that an attribute may assume. This implies that

whether or not an attribute is missing constitutes information that is useful for prediction. In some circumstances this is plausible, but it is a risky assumption across all datasets. When using 1R as a filter, it can be particularly misleading to choose attributes with large numbers of missing values that seem to make highly accurate predictions.

2.4.3 C4.5 Algorithm

C4.5[65] is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. Authors of the Weka machine learning software described the C4.5 algorithm as "a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date" [66].

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, \ldots$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, \ldots, x_{p,i})$, where the x_j represent attribute values or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the partitioned sublists.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.

- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.

- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

According to [67], in pseudo code, the general algorithm for building decision trees is:

- Check for the above base cases.
- For each attribute a, find the normalized information gain ratio from splitting on a.
- Let a_best be the attribute with the highest normalized information gain.
- Create a decision $node$ that splits on a_best.
- Recur on the sublists obtained by splitting on a_best, and add those nodes as children of $node$.

2.4.4 Naive Bayes classifier

In machine learning, Naive Bayes classifiers[68] are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s [69].

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $x = (x_1, \ldots, x_n)$ representing some n features (independent variables), it assigns to this instance probabilities $p(C_k|x_1, \ldots, x_n)$ for each of K possible outcomes or classes C_k.

The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \tag{12}$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$posterior = \frac{prior \times likelihood}{evidence} \tag{13}$$

2.4.5 Sequential Minimal Optimization

Sequential minimal optimization (SMO) [70] is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. It was invented by John Platt in 1998 at Microsoft Research. SMO is widely used for training support vector machines.

SMO is an iterative algorithm for solving the optimization problem described above. SMO breaks this problem into a series of smallest possible subproblems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2,

the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C \tag{14}$$

$$y_1\alpha_1 + y_2\alpha_2 = k \tag{15}$$

and this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. k is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration.

The algorithm proceeds as follows:

Find a Lagrange multiplier α_1 that violates the Karush-Kuhn-Tucker (KKT) conditions [71] for the optimization problem.

Pick a second multiplier α_2 and optimize the pair (α_1, α_2).

Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are $n(n-1)/2$ possible choices for α_i and α_j.

2.4.6 J48 Decision Trees

A decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes, the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable.

The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset.

The J48 Decision tree classifier is an implementation of C4.5 [65] developed by the WEKA [72] project team. It follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess.

Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance.

2.4.7 Multilayer perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. [73, 74]. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable [75].

A multilayer perceptron (MLP) is a deep, artificial neural network. It is

composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

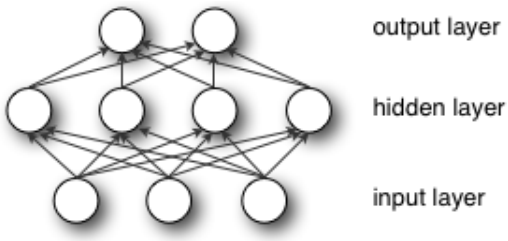

Figure 15: Example multilayer perceptron network

Multilayer perceptrons are often applied to supervised learning problems: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

Feedforward networks such as MLPs are like tennis, or ping pong. They are mainly involved in two motions, a constant back and forth. You can think of this ping pong of guesses and answers as a kind of accelerated science, since each guess is a test of what we think we know, and each response is feedback letting us know how wrong we are.

In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth labels.

In the backward pass, using backpropagation and the chain rule of calculus, partial derivatives of the error function w.r.t. the various weights and biases are back-propagated through the MLP. That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the MLP one step closer to the error minimum. This can be done with any gradient-based optimization algorithm such as stochastic gradient

descent. The network keeps playing that game of tennis until the error can go no lower. This state is known as convergence.

2.4.8 Dynamic Time Warping

In time series analysis, dynamic time warping (DTW)[76] is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed. For instance, similarities in walking could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data – indeed, any data that can be turned into a linear sequence can be analyzed with DTW. A well known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

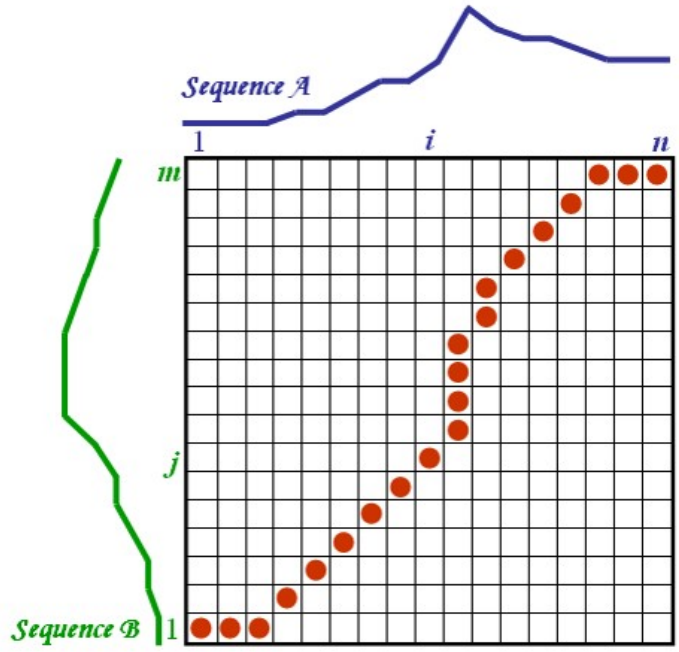

Figure 16: Sequence alignment in Dynamic Time Warping

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restriction and rules:

- Every index from the first sequence must be matched with one or more

indices from the other sequence, and vice versa

- The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)
- The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)
- The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa, i.e. if $j > i$ are indices from the first sequence, then there must not be two indices $l > k$ in the other sequence, such that index i is matched with index l and index j is matched with index k, and vice versa

The optimal match is denoted by the match that satisfies all the restrictions and the rules and that has the minimal cost, where the cost is computed as the sum of absolute differences, for each matched pair of indices, between their values.

The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

In addition to a similarity measure between the two sequences, a so called "warping path" is produced, by warping according to this path the two signals may be aligned in time. The signal with an original set of points $X(\text{original})$, $Y(\text{original})$ is transformed to $X(\text{warped})$, $Y(\text{warped})$. This finds applications in genetic sequence and audio synchronization. In a related technique sequences of varying speed may be averaged using this technique.

The pseudocode for DTW between two sequences s and t looks like:

```
DTW := array[0..n, 0..m]
for i := 1 to n
    DTW[i, 0] := infinity
end for
for i := 1 to m
```

```
    DTW[0, i] := infinity
end for
DTW[0, 0] := 0
for i := 1 to n
    for j := 1 to m
        cost := d(s[i], t[j])
        DTW[i, j] := cost + min(DTW[i-1, j],
                                DTW[i, j-1],
                                DTW[i-1, j-1])
return DTW[n, m]
```

where $DTW(i, j)$ is the distance between $s[1 : i]$ and $t[1 : j]$ with the best alignment. Here for two symbols x and y, $d(x, y)$ is the distance between the symbols.

The time complexity of DTW algorithm is $O(NM)$, where N and M are the lengths of the two input sequences. More generally, without loss of generality, assuming that $N \geq M$, the time complexity can be said to be $O(N^2)$. The same is true for space complexity.

Chapter 3

Proposed Methodologies

This chapter presents our proposed Kinect-based 3D gait recognition method that utilizes the skeleton data to construct a robust representation of a human gait signature. The first section provides an overview of the proposed methodology by outlining the components of the system. In the subsequent sections, these components are described in details.

3.1 Overview

The proposed model-based gait recognition system comprises multi-stage processing of the 3D full-body skeleton data obtained from the Kinect sensor. In this regard, we have proposed two methods for feature extraction.
In Angles and Angle Differences, first, joint data is extracted from the Kinect sensor. In case of databases, the walk action sequences are extracted. Then 8 angles are calculated. These angles are most active angles according to [18, 61]. These angles are calculated over multiple frames. After that difference between 4 pairs are calculated. This is similar to the rate of change. The values are normalized to ensure that it has the same effect as other 8 angles. The distance from camera and direction of walking can cause change in angle, but the difference between two angles remain the same. This ensures view and scale invariant feature.
In Angles, Mean and Standard Deviation(SD), first the same 8 angles are calculated. Then the mean and standard deviation of each 8 angles over multiple frames are calculated.

3.2 Framework

The framework is developed based on joint angle, angle difference and mean. The generalized classification steps look like:

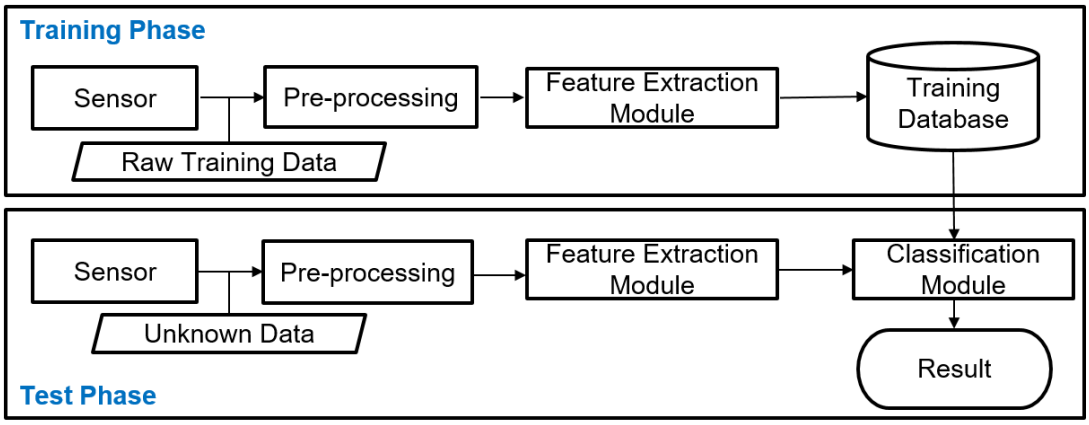

Figure 17: Gait Recognition Framework

3.3 Training Phase

3.3.1 Sensor

Kinect v1 can extract 20 joint points from a single person as seen in figure 18.

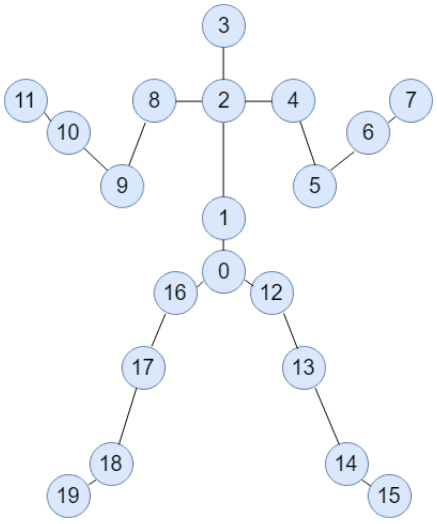

Figure 18: Joints Extracted By Kinect v1

3.3.2 Pre-Processing

As seen in the figure 17 contains collection of raw video sequence is captured from Kinect sensor. The joint data is then extracted from the raw data. The joint data contains 20 skeletal joints coordinates $(x_1, y_1, z_1), \ldots, (x_m, y_m, z_m)$ where m is the number of frames in the captured video sequence.

3.3.3 Feature Extraction

Multiple frames for each user is recorded. For each frame, two types of angles are calculated.

- Angle Between Joints
- Angle Between Lines

Angle between 3 joints a, b, c can be calculated using:

$$\theta_a = cos^{-1}\frac{P_{ba}^2 + P_{bc}^2 - P_{ac}^2}{2P_{ba}P_{bc}} \tag{16}$$

$$P_{ba} = sqrt((b_x - a_x)^2 + (b_y - a_y)^2 + (b_z - a_z)^2) \tag{17}$$

$$P_{bc} = sqrt((b_x - c_x)^2 + (b_y - c_y)^2 + (b_z - c_z)^2) \tag{18}$$

$$P_{ac} = sqrt((a_x - c_x)^2 + (a_y - c_y)^2 + (a_z - c_z)^2) \tag{19}$$

$$\tag{20}$$

The angles are then converted to degree using

$$\theta_d = \frac{\theta_r \times 180}{\pi} \tag{21}$$

The angles are now in range $[0, 180]$.

Given two lines having endpoints a, b and c, d, the angle between these two lines are calculated like this:

$$P_x = b_x - a_x \tag{22}$$

$$P_y = b_y - a_y \tag{23}$$

$$P_z = b_z - a_z \tag{24}$$

$$Q_x = d_x - c_x \tag{25}$$

$$Q_y = d_y - c_y \tag{26}$$

$$Q_z = d_z - c_z \tag{27}$$

$$\theta_l = cos^{-1}\frac{P_x.Q_x + P_y.Q_y + P_z.Q_z}{|\vec{P}|.|\vec{Q}|} \tag{28}$$

Again the angles are converted to degree.

Angles and Angle Differences

The angles that were considered from the figure 19:

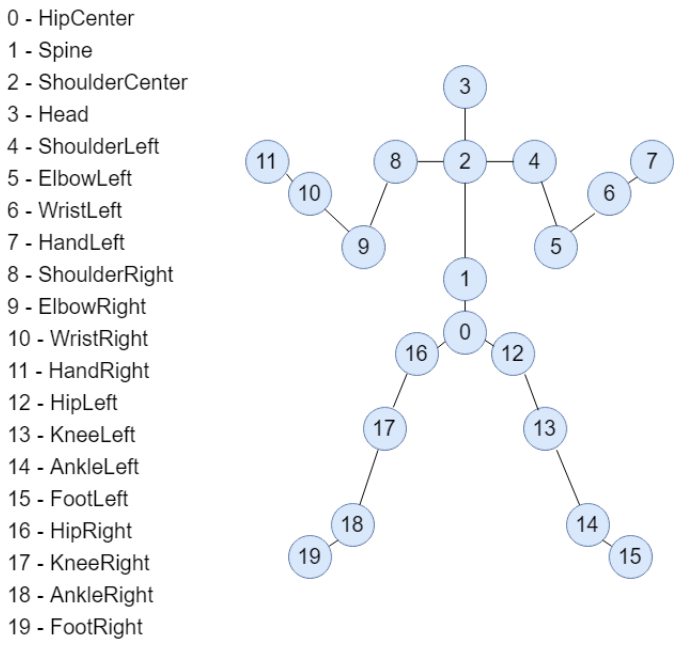

Figure 19: Extracted Joints

- Upper Body
  1. Shoulder Center(2) - Shoulder Right(8) - Elbow Right(9)
  2. Shoulder Center(2) - Shoulder Left(4) - Elbow Left(5)
  3. Shoulder Right(8) - Elbow Right(9) - Wrist Right(10)
  4. Shoulder Left(4) - Elbow Left(5) - Wrist Left(6)
- Lower Body
  1. Hip Center(0) and Spine(1) Line - Hip Right(16) and Knee Right(17) Line
  2. Hip Center(0) and Spine(1) Line - Hip Left(12) and Knee Left(13) Line
  3. Hip Right(16) - Knee Right(17) - Ankle Right(18)
  4. Hip Left(12) - Knee Left(13) - Ankle Left(14)

Figure 20 shows the angles that are considered. Some differences between angles are calculated. They are:

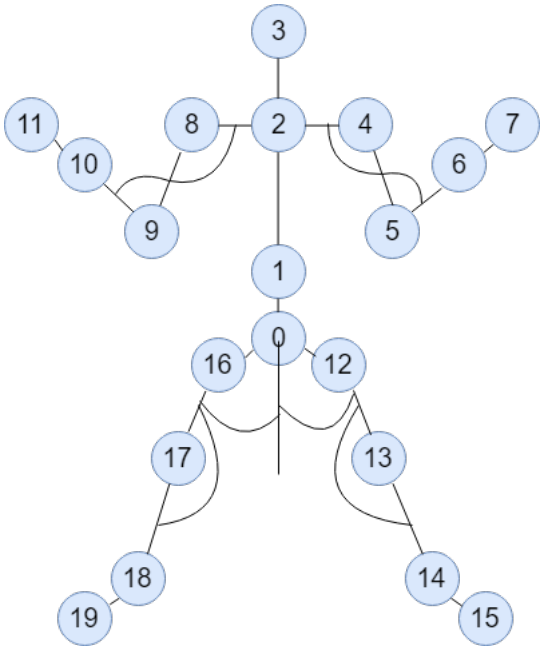

Figure 20: Considered Angles

- $Upper\ Body\ 1 - Upper\ Body\ 3$
- $Upper\ Body\ 2 - Upper\ Body\ 4$
- $Lower\ Body\ 1 - Lower\ Body\ 3$
- $Lower\ Body\ 2 - Lower\ Body\ 4$

These angles are then normalized using 0-1 normalization and multiplied by 180 so that they have same effect in classification:

$$\theta_{i'} = \frac{\theta_i - min(\theta)}{max(\theta) - min(\theta)} \times 180 \tag{29}$$

For each user, we get $(8 + 4) = 12$ sequences of length m.

Angles, Mean and Standard Deviation

In this method, 8 angles are calculated similar to the previous method. In addition to that, for each angle mean and standard deviation are calculated over all frames.

We calculate mean using:

$$mean_j = \frac{1}{m} \sum_{i=1}^{m} \theta_a \tag{30}$$

where j is in range $[1, 8]$.

In similar manner, we calculate standard deviation:

$$SD_j = \sqrt{\frac{(\theta_{ai} - mean_j)}{m}} \tag{31}$$

where j is in range $[1, 8]$.

We get two sequences each having 8 values. For each user, we get $(8 + 1 + 1) = 10$ sequences. The first 8 have length m, and the last 2 have length 8.

3.3.4 Training Database

The calculated feature vectors are stored in database. For each user, one file is created.

3.4 Testing Phase

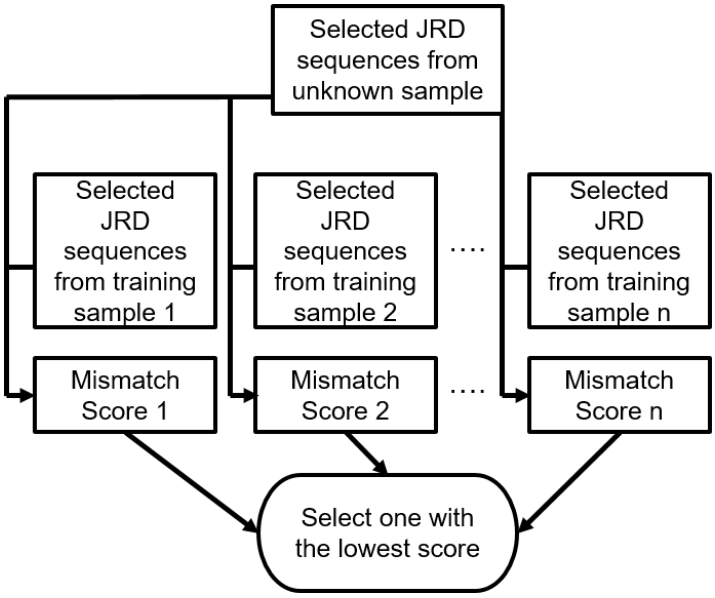

Figure 21: Gait Recognition Process in Testing Phase

In testing phase, unknown sequence goes through similar method as described to get feature vector. This feature vector is then tested against the data stored in the training database.

For each training sample, the unknown feature vector is compared with the training feature vector. Here the feature vectors are considered as sequences.

Dynamic Time Warping algorithm is used to calculate the dissimilarity between each user and the unknown sample. The training sample with the lowest dissimilarity value is selected as the user. This is shown in figure 21.

Chapter 4

Experimental Analysis

In this chapter, we discuss about the experimental setup, comparison, dataset, result analysis based on different criteria.

4.1 Experimental Setup

The experiment was conducted on a personal computer having:

- Processor: AMD Ryzen 7 1700
  - 8 core
  - 16 CPUs
  - 3.0Ghz
- Ram: 16GB
- Cache Size:
  - L2 Cache: 4MB
  - L3 Cache: 16MB

4.2 Evaluation Methodologies

These evaluation methodologies were used to compare our proposed methodologies with previous literature:

4.2.1 Feature Count

A feature is an individual measurable property or characteristic of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression. Feature count is the total number of features used in the feature representation of a gait cycle. The total number of feature actually

varies with each user as it depends on the number of frames for each user. The actual feature count is equal to the number of features multiplied by the frame count for each user.

4.2.2 Execution Time

Execution time is the time during which a program is running. In our case, it is the time taken for a program to execute for both the training phase and the testing phase. In our case, we trained each implementation with half the dataset and matched them with other half. The time was calculated for all the matches.

4.2.3 Accuracy

Accuracy is the quality or state of being correct or precise. It is the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard. In our case, accuracy can be defined as the percentage of unknown users correctly identified by using gait analysis.

4.3 Dataset

To evaluate the effectiveness of the proposed method, two publicly available Kinect action databases were used, namely the UTKinect-Action3D Dataset [77] and UPCV Action Dataset [78].

4.3.1 UTKinect-Action Dataset

Figure 22: UTKinect-Action Dataset

The UTKinect-Action Dataset comprises 10 types of human actions (walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands) in

indoor settings captured using a Kinect sensor at 30 frames per second (FPS). There are 10 subjects and each subject performs each action twice. The length of each action sequence is usually between 9-151 frames.

4.3.2 UPCV Action Dataset

Figure 23: UPCV Action Dataset

The UPCV Action dataset consists of 10 actions performed by 20 different individuals (10 males and 10 females), between the age of 22-50, in two separate sessions for each one. The actions set was chosen in order to contain usual indoor and outdoor activities performed by pedestrians like walking, grab something from the floor, looking at the wrist watch, scratching the head, answer a cell phone, crossing arms and sitting on a chair, and some unusual actions such as throwing a punch, kicking and waving hands. The length of action sequences varies between 8 and 500 frames, with the 90% of the sequences having a length in the range of 23- 167 frames.

4.4 Result Analysis

Number of features, execution time (both training and test) and accuracy was used to compare the proposed methodologies with the previous ones:

4.4.1 Feature Count

The table 1 shows the number of feature sequences that we get in feature vector for the different methods. Total number of features varies with each user as it depends on the number of video frames for each user. Actual feature count is equal to given count multiplied by frame count. The two proposed methods has the lowest feature count as can be seen from the table.

Method	Count
JRD [79]	380
JRA	171
Most Relevant JRA [16]	40-50
JRD JRA Rank Level Fusion [15]	(15+25)=40
JRD + JRA	(380+171) = 551
Angles + Angle Differences	**12**
Angles + Mean + SD	**10**

Table 1: Comparison using Feature Count

Rahman et al. [79] used all possible combinations of angles relative to Spine. They didn't consider whether that angle was relevant or not. Faisal et al. [16] tried to fix it using most relevant JRA. It was calculated using histogram. The number of bins occupied in histogram by each angle joints were used to reduce the number of features. In another paper, Faisal et al. used genetic algorithm based approach to find relevant JRDs and JRAs to find out the angles that provide better classification.

In our methodologies, the angles chosen were inspired by Ball et al. [18] and Tafazzoli et al. [80]. Ball et al. [18] argued that while walking the links of the lower body was relevant as it is possible for the arms to be variously occupied while walking and this variation of arm data may lead to poor clustering. The features used for clustering were based on the lower limb joint angles. It was found that the limb length assigned by the Microsoft Kinect SDK skeletal algorithm changed significantly as the person walked across the camera field of view. This variation precludes distance-based features from being used for clustering, as the same person could walk past the camera twice and have two different skeletal representations. It was found that noise was also present in the link angle data. As the skeletal algorithm maintains link end-point connectedness for the duration of the walk, however, the relative change in joint position adds much less noise to angle-based features.

On the other hand, Tafazzoli et al. [80] chose a simple characteristic of normal human walking as the basis. They argued that the swing amplitude of the arms during normal walking is observed to be much larger than that of the legs. One of the unique properties of walking bilateral symmertry; that is, when one walks or runs the left arm and right leg interchange direction of

swing with the right arm and left leg, and vice versa, with a phase shift of half a period [47].

4.4.2 Execution Time

As we can from table 2, , the execution time for the proposed methods during both testing and training is shorter than all the other methods. This is true for both UTKinect-Action3D Dataset and UPCV Action Dataset. As our

Method	Execution Time (seconds)			
	UTKinect		UPCV	
	Train	**Test**	**Train**	**Test**
JRD [79]	1.249	6.286	3.756	90.226
JRA	0.958	6.289	3.004	84.165
Most Relevant JRA [16]	0.714	2.591	2.718	26.873
JRD JRA Rank Level Fusion [15]	0.859	2.738	2.922	27.403
JRD + JRA	2.107	12.356	6.964	172.743
Angles + Angle Differences	**0.142**	**0.211**	**0.409**	**1.066**
Angles + Mean + SD	**0.134**	**0.187**	**0.401**	**0.966**

Table 2: Comparison using Execution Time

methodologies used less features than the number of features used in other methodologies, it took less time to calculate and compare the angles. Additionally, our proposed methodologies were more robust against the noise than the other methods. As a result, we were able to avoid preprocessing.

4.4.3 Accuracy

UPCV Action Dataset:

Method	**Accuracy**
JRD [79]	77.50%
JRA	71.25%
Most Relevant JRA [16]	65.00%
JRD JRA Rank Level Fusion [15]	66.25%
JRD + JRA	78.75%
Angles + Angle Differences	**73.75%**
Angles + Mean + SD	**87.50%**

Table 3: Comparison result for UPCV Action Dataset

As we can see from table 3, the proposed method of Angles + mean + SD performs better than the other proposed method. This can be attributed to the dataset containing sequences of users walking in the same direction and the distance from the camera remaining constant for all the users. This is evident as we can see that other distance based methods perform better here.

UTKinect-Action3D Dataset:

Method	Accuracy
JRD [79]	60.00%
JRA	62.50%
Most Relevant JRA [16]	60.00%
JRD JRA Rank Level Fusion [15]	72.50%
JRD + JRA	72.50%
Angles + Angle Differences	**77.50%**
Angles + Mean + SD	**75.00%**

Table 4: Comparison result for UTKinect-Action3D Dataset

As we can see from the table 4, the proposed method of Angle + Angle Differences performs better as the features are view and scale invariant and as the dataset contains users walking in different directions and the distance of the camera from the user did not remain constant as well. Other JRD based methods perform much worse here.

Overall:

Method	Accuracy
JRD [79]	68.75%
JRA	66.88%
Most Relevant JRA [16]	66.25%
JRD JRA Rank Level Fusion [15]	69.38%
JRD + JRA	75.63%
Angles + Angle Differences	**75.63%**
Angles + Mean + SD	**81.25%**

Table 5: Comparison result overall

Both the proposed methods perform better or similar compared to the previous methods in overall. The proposed method of Angles + Angle Difference method requires less memory and time than JRD + JRA method, so Angles +

Angles Difference method is better even though they both have the same overall accuracy. The proposed method of Angles + Mean + SD performs the best in overall. This is evident in table 5.

Chapter 5

Conclusion

5.1 Summary

Model-based gait recognition is one of the most widely-studied problems in the areas of biometric and computer vision, with potential applications in security and surveillance, human computer interaction, health-care, intelligent systems, etc. The recent popularization of the Kinect sensor has resulted in a spike in the interest in using the Kinect for gait recognition. Our work in this thesis focuses on designing new methodologies for Kinect-based gait recognition that utilize the 3D virtual skeleton model to construct effective and robust feature representations. In biometric gait recognition we aim to extract person-dependent motion patterns which are unique to a person, typically caused by the influence of human physiology and behavioral traits. For gait recognition, we propose two new methods for effective gait signature representation. One method took 8 angles and their angle differences and the other method took the angles and the mean and standard deviation of those angles. These angle sequences were matched with an unknown sample sequence using Dynamic Time Warping (DTW). The angle sequence with the lowest dissimilarity measure is selected for biometric identification. The experimental result show that the proposed method of the angles and the mean and standard deviation of those angles worked best with an overall accuracy of 81.25%.

5.2 Future Work

Gait features are typically sensitive to changes in clothing and carrying conditions, which makes recognition in dynamic environment a challenging task. In addition, a few studies have shown that it is possible to spoof gait biometric by imitating clothing and selecting individuals with similar physiological builds and attributes. However, due to the lack of the availability of any Kinect-based gait spoofing dataset, robustness of the proposed methodology under such at-

tacks was not evaluated in this work. One possible alternative is to incorporate other biometric modalities, such as face or voice, in order to increase the robustness of the system. Another promising direction is to incorporate context information with gait features to boost the recognition performance. Another way of improving the proposed method is to combine the mean and standard deviation features with view and scale invariant angle different features so that the features are view and scale invariant. We used only one classifier, Dynamic Time Warping (DTW), in our proposed method. We would like to make an in depth analysis of the change in accuracy when used for different classifiers. We would like to incorporate more datasets to see if our results change or not. We would also like to experiment with various new feature selection methods so as to improve our biometric identification accuracy.

References

[1] A. E. Minetti, "The biomechanics of skipping gaits: a third locomotion paradigm?" *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 265, no. 1402, pp. 1227–1233, 1998.

[2] A. Nandy, S. Mondal, L. Rai, P. Chakraborty, and G. C. Nandi, "A study on damping profile for prosthetic knee," in *proceedings of the international Conference on Advances in Computing, Communications and Informatics*. ACM, 2012, pp. 511–517.

[3] N. A. Makhdoomi, T. S. Gunawan, and M. H. Habaebi, "Human gait recognition and classification using similarity index for various conditions," in *IOP Conference Series: Materials Science and Engineering*, vol. 53, no. 1. IOP Publishing, 2013, p. 012069.

[4] D. A. Winter, "The biomechanics and motor control of human gait. waterloo," 1987.

[5] A. Shpunt and Z. Zalevsky, "Depth-varying light fields for three dimensional sensing," Nov. 1 2011, uS Patent 8,050,461.

[6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Ieee, 2011, pp. 1297–1304.

[7] K. H. Kim, Y.-S. Chung, J.-H. Yoo, and Y. M. Ro, "Facial feature extraction based on private energy map in dct domain," *ETRI journal*, vol. 29, no. 2, pp. 243–245, 2007.

[8] T. Jabid, M. H. Kabir, and O. Chae, "Robust facial expression recognition based on local directional pattern," *ETRI journal*, vol. 32, no. 5, pp. 784–794, 2010.

[9] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE security & privacy*, no. 2, pp. 33–42, 2003.

[10] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *Proceedings of 2nd Int'l Conference on Audio-and Video-based Biometric Person Authentication, Washington DC*, 1999, pp. 182–187.

[11] M. P. Down and R. Sands, "Biometrics: An overview of the technology, challenges and control considerations," *Information Systems Control Journal*, vol. 4, pp. 53–56, 2004.

[12] F. Ahmed and M. Gavrilova, "Biometric-based user authentication and activity level detection in a collaborative environment," in *Transparency in Social Media*. Springer, 2015, pp. 165–180.

[13] I. Deutschmann, L. Nilsson, and P. Nordstrom, "Continuous authentication, using behavioral biometrics, with keystroke and mouse," *IT Professional*, p. 1, 2013.

[14] X. Zhou and B. Bhanu, "Integrating face and gait for human recognition at a distance in video," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1119–1137, 2007.

[15] F. Ahmed, P. P. Paul, and M. L. Gavrilova, "Dtw-based kernel and rank-level fusion for 3d gait recognition using kinect," *The Visual Computer*, vol. 31, no. 6-8, pp. 915–924, 2015.

[16] F. Ahmed, P. Polash Paul, and M. L. Gavrilova, "Kinect-based gait recognition using sequences of the most relevant joint relative angles," 2015.

[17] R. Tanawongsuwan and A. Bobick, "Gait recognition from time-normalized joint-angle trajectories in the walking plane," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–II.

[18] A. Ball, D. Rye, F. Ramos, and M. Velonaki, "Unsupervised clustering of people from'skeleton'data," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 225–226.

[19] M. S. Nixon, J. N. Carter, M. G. Grant, L. Gordon, and J. B. Hayfron-Acquah, "Automatic recognition by gait: progress and prospects," *Sensor Review*, vol. 23, no. 4, pp. 323–331, 2003.

[20] L. Zhou, Z. Lu, H. Leung, and L. Shang, "Spatial temporal pyramid matching using temporal sparse representation for human motion retrieval," *The Visual Computer*, vol. 30, no. 6-8, pp. 845–854, 2014.

[21] M. S. Bae and I. K. Park, "Content-based 3d model retrieval using a single depth image from a low-cost 3d camera," *The Visual Computer*, vol. 29, no. 6-8, pp. 555–564, 2013.

[22] Y. Zhang, J. Zheng, and N. Magnenat-Thalmann, "Example-guided anthropometric human body modeling," *The Visual Computer*, vol. 31, no. 12, pp. 1615–1631, 2015.

[23] J. Barth, J. Klucken, P. Kugler, T. Kammerer, R. Steidl, J. Winkler, J. Hornegger, and B. Eskofier, "Biometric and mobile gait analysis for early diagnosis and therapy monitoring in parkinson's disease," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 868–871.

[24] L. Xia *et al.*, "Human detection and action recognition using depth information by kinect," Ph.D. dissertation, 2012.

[25] F. Blais, "Review of 20 years of range sensor development," *Journal of electronic imaging*, vol. 13, no. 1, pp. 231–244, 2004.

[26] B. Gil, A. Mitiche, and J. Aggarwal, "Experiments in combining intensity and range edge maps," *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 3, pp. 395–411, 1983.

[27] M. J. Magee and J. K. Aggarwal, "Using multisensory images to derive the structure of three-dimensional objects—a review," *Computer vision, graphics, and image processing*, vol. 32, no. 2, pp. 145–157, 1985.

[28] M. J. Magee, B. A. Boyter, C.-H. Chien, and J. Aggarwal, "Experiments in intensity guided range sensing recognition of three-dimensional objects,"

IEEE transactions on pattern analysis and machine intelligence, no. 6, pp. 629–637, 1985.

[29] B. Vemuri, A. Mitiche, and J. Aggarwal, "3-d object representation from range data using intrinsic surface properties," in *Three-Dimensional Machine Vision*. Springer, 1987, pp. 241–266.

[30] E. E. Stone and M. Skubic, "Evaluation of an inexpensive depth camera for passive in-home fall risk assessment," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*. Ieee, 2011, pp. 71–77.

[31] F. Malawski, B. Kwolek, and S. Sako, "Using kinect for facial expression recognition under varying poses and illumination," in *International Conference on Active Media Technology*. Springer, 2014, pp. 395–406.

[32] M. Jaiswal, J. Xie, and M.-T. Sun, "3d object modeling with a kinect camera," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–5.

[33] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *Robotics and automation (ICRA), 2011 IEEE international conference on*. IEEE, 2011, pp. 20–25.

[34] Y.-J. Chang, S.-F. Chen, and J.-D. Huang, "A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities," *Research in developmental disabilities*, vol. 32, no. 6, pp. 2566–2570, 2011.

[35] M. Burke and J. Lasenby, "Fast upper body joint tracking using kinect pose priors," in *International Conference on Articulated Motion and Deformable Objects*. Springer, 2014, pp. 94–105.

[36] M.-C. Hu, C.-W. Chen, W.-H. Cheng, C.-H. Chang, J.-H. Lai, and J.-L. Wu, "Real-time human movement retrieval and assessment with kinect sensor," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 742–753, 2015.

[37] G. T. Papadopoulos, A. Axenopoulos, and P. Daras, "Real-time skeleton-tracking-based human action recognition using kinect data," in *International Conference on Multimedia Modeling*. Springer, 2014, pp. 473–483.

[38] W. Tao, T. Liu, R. Zheng, and H. Feng, "Gait analysis using wearable sensors," *Sensors*, vol. 12, no. 2, pp. 2255–2283, 2012.

[39] T. C. Pataky, T. Mu, K. Bosch, D. Rosenbaum, and J. Y. Goulermas, "Gait recognition: highly unique dynamic plantar pressure patterns among 104 individuals," *Journal of The Royal Society Interface*, vol. 9, no. 69, pp. 790–800, 2012.

[40] T. Amin, "Dynamic descriptors in human gait recognition," Ph.D. dissertation, 2013.

[41] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: a challenging signal processing technology for biometric identification," *IEEE signal processing magazine*, vol. 22, no. 6, pp. 78–90, 2005.

[42] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[43] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*. IEEE, 2010, pp. 320–327.

[44] J. Han and B. Bhanu, "Statistical feature fusion for gait-based human recognition," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–II.

[45] C. BenAbdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 372–377.

[46] R. Urtasun and P. Fua, "3d tracking for gait characterization and recognition," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. IEEE, 2004, pp. 17–22.

[47] C. Yam, M. S. Nixon, and J. N. Carter, "Automated person recognition by walking and running via model-based approaches," *Pattern recognition*, vol. 37, no. 5, pp. 1057–1072, 2004.

[48] A. Sinha, K. Chakravarty, and B. Bhowmick, "Person identification using skeleton information from kinect," in *Proc. Intl. Conf. on Advances in Computer-Human Interactions*, 2013, pp. 101–108.

[49] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A full-body layered deformable model for automatic model-based gait recognition," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 62, 2008.

[50] K. Arai and R. Andrie, "Gait recognition method based on wavelet transformation and its evaluation with chinese academy of sciences (casia) gait database as a human gait recognition dataset," in *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*. IEEE, 2012, pp. 656–661.

[51] J. D. Shutler, M. S. Nixon, and C. J. Harris, "Statistical gait description via temporal moments," in *ssiai*. IEEE, 2000, p. 291.

[52] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[53] C. BenAbdelkader, R. Cutler, and L. Davis, "Motion-based recognition of people in eigengait space," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 267–272.

[54] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 316–322, 2006.

[55] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," Yale University New Haven United States, Tech. Rep., 1997.

[56] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, "Frame difference energy image for gait recognition with incomplete silhouettes," *Pattern Recognition Letters*, vol. 30, no. 11, pp. 977–984, 2009.

[57] X. Li and Y. Chen, "Gait recognition based on structural gait energy image," *Journal of Computational Information Systems*, vol. 9, no. 1, pp. 121–126, 2013.

[58] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, "Gait recognition with kinect," in *1st international workshop on kinect in pervasive computing*. New Castle, UK, 2012, pp. P1–P4.

[59] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanid gait challenge problem: Data sets, performance, and analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[60] J. K. Tang, H. Leung, T. Komura, and H. P. Shum, "Emulating human perception of motion similarity," *Computer Animation and Virtual Worlds*, vol. 19, no. 3-4, pp. 211–221, 2008.

[61] B. Dikovski, G. Madjarov, and D. Gjorgjevikj, "Evaluation of different feature sets for gait recognition using skeletal data from kinect," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1304–1308.

[62] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[63] C. G. Nevill-Manning, G. Holmes, and I. H. Witten, "The development of holte's 1r classifier," in *Artificial Neural Networks and Expert Systems, 1995. Proceedings., Second New Zealand International Two-Stream Conference on*. IEEE, 1995, pp. 239–242.

[64] R. C. Holte, L. Acker, B. W. Porter *et al.*, "Concept learning and the problem of small disjuncts." in *IJCAI*, vol. 89. Citeseer, 1989, pp. 813–818.

[65] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[66] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[67] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[68] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[69] R. Stuart and N. Peter, "Artificial intelligence-a modern approach 3rd ed," 2016.

[70] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[71] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 247–258.

[72] G. Holmes, A. Donkin, and I. H. Witten, "Weka: A machine learning workbench," in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*. IEEE, 1994, pp. 357–361.

[73] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," CORNELL AERONAUTICAL LAB INC BUFFALO NY, Tech. Rep., 1961.

[74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[75] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[76] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[77] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE, 2012, pp. 20–27.

[78] I. Theodorakopoulos, D. Kastaniotis, G. Economou, and S. Fotopoulos, "Pose-based human action recognition via sparse representation in dissimilarity space," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 12–23, 2014.

[79] M. W. Rahman and M. L. Gavrilova, "Kinect gait skeletal joint feature-based person identification," in *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2017 IEEE 16th International Conference on*. IEEE, 2017, pp. 423–430.

[80] F. Tafazzoli and R. Safabakhsh, "Model-based human gait recognition using leg and arm movements," *Engineering applications of artificial intelligence*, vol. 23, no. 8, pp. 1237–1246, 2010.