TOWARDS GAIT-BASED HUMAN IDENTIFICATION USING FRONT VIEW DEPTH IMAGES

By

HITESH JHURANI

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

written under the direction of

Prof. Janne Lindqvist

and approved by

New Brunswick, New Jersey

October, 2016

ABSTRACT OF THE THESIS

TOWARDS GAIT-BASED HUMAN IDENTIFICATION USING FRONT VIEW DEPTH IMAGES

By HITESH JHURANI

Thesis Director:

Prof. Janne Lindqvist

There are several competing approaches towards identifying humans based on their biometrics. In this thesis, we designed, implemented a system towards human front gait identification. We leveraged the depth information obtained from front view depth image of a person. We explain how basic features which can be extracted from the depth image of an individual can be used for identifying a person correctly. Basic features such as depth values of legs, the angle between the legs and height of the person are utilized. We conclude the thesis discussing tradeoffs of the approach and avenues for further research.

Acknowledgments

I would express my deepest gratitude to my advisor, Prof. Janne Lindqvist, for giving me the opportunity to be a part of this project and also for his constant guidance and support. I would like to extend a big thank you to the enthusiastic people in my lab especially Can Liu who helped me in the testing and other aspects of my thesis, without which it would have been difficult to complete my thesis.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

A person can be identified by a variety of biometric characteristics such as their fingerprints, voice and face [1]. These features might seem to be sufficient when it comes to the level of security in many cases, but they still require the user to actively participate when it comes to retrieving a particular feature. Often it is inconvenient and difficult for people to keep their fingers clean enough for the fingerprint reader [3, 7]. People who are aware they are being registered while performing certain tasks, for example recording their signature or voice record, could perform the same task in a different way compared to if they did the same task under normal circumstances [24, 30, 31]. These kinds of problems can be avoided if the person's movements are registered continuously and the verification is done automatically whenever necessary.

Gait recognition technology is a suitable solution to this problem. Gait is unique for every person [1]. A Gait based authentication system helps in developing a method that can automatically register a persons movements as well as the algorithms which can be used to analyse the movements. Such a method for authentication will create a more comfortable authentication process for the user.

1.2 Background

This section gives a brief overview of the fundamentals which are necessary to understand a gait based authentication system. We present a detailed information about the different biometric authentication systems currently being used. How the human gait is used for authentication and different approaches to implement a gait based authentication system.

1.2.1 Biometric Authentication

Biometric features from an information security perspective is a process by which the identity of a person can be confirmed. Use of biometric features for authentication is one of the three most widely used approaches. The other two are, the use of hardware tokens for example USB token, key fob, etc. owned by the user, while the other is the use of text based passwords, pass-phrases and other secrets which is known only to the user [10, 13, 31]. Use of passwords and tokens has been most commonly used method for authentication [1, 7]. Since the technology used to measure biometric features has been large and expensive [7, 10, 13, 31].

In recent years, biometric technology has not only become smaller the price has dropped significantly as well. Therefore it has been used more often as an alternative to traditional approaches for authentication, it has even been used in combination with passwords and hardware tokens. Biometric authentication uses one of the many different biometric features to authenticate the person correctly. Some of the most frequently used features today are fingerprints, face, voice and iris.

Authenticating a person is a two step process. In the first step, the user has to be enrolled and is required to register identity and the biometric feature by which the user will be authenticated. For example, in the fingerprint recognition system, the user is required to register one or several fingerprints using a fingerprint reader. The authentication system stores a template of a fingerprint for every person. The fingerprint stored is a digital representation, where unique features related to it are identified and extracted. Once enrollment is performed, the user is known to the system.

The next step is the authentication attempt. This is done either by identification or verification. When a verification attempt is done, the user enters the identity along with the fingerprint. The system compares this fingerprint with the template stored in the system for that particular user, that is a one-to-one comparison. During this identification process, the user enters only the fingerprint and the system will check this with all of the fingerprint templates stored in the database, that is a one-to-many comparison. If the results from the comparison yields a matching score, then there is a similarity between the two templates.

The system has a pre-set threshold value which determines how large the matching score can be for two templates to be recognized as identical. The accuracy of the system depends upon the threshold value. A pre-set threshold value is the reference value which decides whether the person is genuine or an imposter. A small threshold value tolerates a low similarity score, thus resulting in situations where two different people might be recognized as the same person by the system; a false acceptance. False acceptance takes place when a non-authorized person is authorized as genuine. A high threshold value tolerates a high matching scores, which might result in enrolled persons not being recognized by the system; a false rejection. False rejection takes place when a genuine person is rejected by the system. These are two basic errors of any authentication system [1, 10, 31]. The amount of false acceptances and rejections compared to the total number of authentication attempts on a system is known as the false acceptance rate (FAR) and false rejection rate (FRR) respectively.

1.2.2 The Human Gait

One of the first studies of human gait was made in early 1900s by Marks [2] who described how the process of walking can be divided into different phases and observed how different fake limb designs of an amputee gait had an effect on these phases. Nowadays, we divide the human gait into different gait cycles, a gait cycle is defined as the period from an initial contact of one foot to the following initial contact of the same foot. This period is divided into three main steps, which again is divided into eight phases. The first step is the weight acceptance period. In this step, one foot is placed on the ground and the stability is maintained by shifting the body weight in order to absorb shock. The second step, is a single limb support task consisting of a mid-stance phase, a terminal stance phase and a transition to the pre-swing phase. In this step, the contra-lateral foot swings forward while the stable foot maintains the body weight. The last step is the limb advancement, in this step, the stable foot in the second step leaves the ground, which moves the body forward.

1.2.3 Gait Based Human Identification System

A gait based identification system is an automatic recognition of people based on their behavioural characteristics. Human gait is complex, but it has a distinctive pattern which mainly comprises of synced movements of body parts and joints and the interaction between them. Thus, it could be considered as one of the most distinctive components for biometric authentication. As the psychological research discovery from Johannson [42] showed, people are able to recognize their friend's walking style based on the light markers that are fixed to various important body parts. Since then, a lot of research work [7, 9, 10, 12, 17] has been carried out on gait analysis which has shown that gait can be used to recognize people.

Human gait is an unobtrusive biometric component which can be captured from a distance without requiring any intervention from the user. The performance of gait recognition system can be affected by different factors such as light illumination, duration, load carrying, speed of walking, apparel of the subject and camera view-point. This makes designing a gait recognition system a challenging problem. Recently numerous studies [22, 24, 35, 38, 40] have focused on view invariant gait recognition system similar to realistic surveillance situations, i.e. users are expected to walk in many different directions to reach their destination.

In this thesis we have concentrated on a machine vision based gait recognition system. Many of the machine vision based gait recognition systems [18, 22, 31] are initialized by extracting what we call the human silhouette, which is basically attributing image pixels to the shape of an individual, from images or video, to extract spatio temporal behaviour patterns. These silhouettes are then processed for optimizing the registration process. Different computer vision based algorithm and machine learning techniques are used to extract features that are related to the gait. Following this, different gaits are then stored in the database during the registration phase. In the authentication phase, a test sample is recorded and is compared to the stored gait templates from the database to be used later to identify or validate a person.

1.2.4 Different Gait Authentication Techniques

Gait authentication techniques can be divided into three groups [10, 13, 32, 34, 37, 42, 44]:

Machine Vision: In the machine vision approach [10, 13, 14, 16, 18, 21, 22], images of people while walking are captured using a camcorder placed at long distance. After the image is captured, the gait patterns need to be extracted so image processing techniques, such as converting images into black and white and background subtraction methods, can be used. These features include long steps, static parameters of the body, the distance between head and pelvis, the maximum distance between pelvis and legs and the distance between legs. Machine vision based techniques are primarily used in surveillance scenarios.

Ground Sensors: In the ground sensors approach [32, 33, 34, 35], sensors are installed on the ground which makes them suitable for controlling access to areas where high security is required. When a person walks on them they are able to measure the force on the ground. Characteristics which are be measured by sensors include heel strike, length of each step, pace etc. These sensors usually are located in front of places with restricted access.

Wearable Sensors: In the wearable sensors approach [8, 11, 25, 36, 37, 39], the sensors can be accelerometers (measuring the acceleration), gyro sensors (measuring the rotation with number of degrees per rotation), force sensors (measuring force exerted while walking). Sensors may be installed on a person's belt, around their thigh or around their shin. These sensors might be even put in a person's pocket. These sensors were proposed to be utilized in cellphones and portable electronic gadgets for support, protection and authentication purposes. Hence, it can be used for continuous verification of the user without intervention.

1.2.5 Challenges with Gait Authentication

Even though all of the biometric gait authentication approaches mentioned in the previous section are encouraging, different factors affect the accuracy of such approaches. We can divide these factors that influence a biometric gait system in two categories [1, 43, 44] :

External Factors: External factors mostly impose challenges on the recognition approach (or algorithm). For example, different angles from which the person is viewed that is front view or side view, whether the gait is recorded during day or night time, the surrounding environment example indoor/outdoor, apparels of the user, the surface on which people are walking like grass or concrete road, different types of shoes e.g. mountain boots/sandals, if a person is carrying any object like briefcase or backpack.

Internal Factors: Internal factors can cause changes in natural gait due to sickness/injury (example foot injury, lower limb disorder, Parkinsons disease etc.) or some physiological changes in body because of age, drunkenness, pregnancy, gaining or losing weight.

One of the public gait databases mentioned in the paper published by Sarkar et al. [10] discusses the five factors that can affect the authentication rate of the system. Factors include change in the view angle, type of shoe, the surface on which the people walk, carrying or not carrying any object and time elapsed between the samples which are compared. An example in the paper shows the differences between the template and test samples of shoe had type (X vs Y), viewing angle (right camera vs left camera), object (carried vs not carried) and surface (soft surface vs hard surface) recognition rates that were 78 percent, 73 percent, 61 percent and 32 percent respectively.

Few of the external factors can have drastic effects on different gait authentication approaches. If the person is carrying an object that affects the recognition of both wearable sensor and machine vision based techniques, it creates difficulty in extraction of the human silhouette in machine based techniques. The effect of a subject carrying an object was studied by Gafurov et al. [11] they observed an increase in the equal error rate from 7.3 percent to 9.3 percent.

1.2.6 Methods of Gait Authentication

Gait analysis can be divided into two major categories [20, 30, 31, 40, 44], the model-based approach and the model-free approach.

Model Based Approach: The model based approach creates a representation of the human body or motion and extracts the features to match them with different model components. The knowledge of human body shapes and the dynamics of the human gait are incorporated into the feature extraction phase. Gait dynamics are extracted by determining different joint positions. A few examples [4, 5, 12, 22] of this approach are static body parameters, thigh joint trajectories, articulated model and two dimensional stick figures. The advantage of the model based approach is that, it gives us the ability to directly derive the dynamic gait features from the model parameters. There is no background noise and there is also no effect of the person's apparel or the camera view from which the gait is recorded. The problem of creating many parameters from the extracted features is that, it results in a complex model. Because of this, the total computation time and the storage required for data results in a high cost due to complex searching and matching.

Model-Free Approach: The model-free approach differentiates the whole motion pattern of the human body by terse representation, for example a silhouette, and does not consider an underlying structure. The parameters are extracted from static gait features like centroid, width and height of the silhouette. Research examples [12, 15, 20, 24] of this approach are self-similarity eigenvalue gait, kinematic features, unwrapped silhouette, higher order correlation, video oscillations and gait sequences. A few advantages of the model-free approach are high speed processing, low computation cost and less storage required for data. However, the performance of this approach is usually affected by the background noise and can be affected by the apparels worn by the subject.

Chapter 2

Related Work

2.1 Initial Research in Gait Authentication

Studies on identifying humans initially began in the field of psychology in the 1970s. The discovery made by Cutting and Kozlowski [3] describes how friends and family members were able to recognise each other by the way they walked, they showed how the people were recognized by only observing light reflecting markers which were attached to different body parts of the person walking. Initial attempt of automatic gait analysis were performed in 1994 by Niyogi and Adelson [4]. They described a way by which changes in a two dimensional video footage of a walking person could be analysed. They define gait as an idiosyncratic feature of a person that is determined by, among other things, an individuals weight, limb height, footwear, and posture combined with characteristic motion. They extend this definition to include appearance of the person, the aspect ratio of the torso, the clothing, the amount of arm swing, and the period and phase of a walking cycle. From these two definitions, it can be derived, gait is a useful parameter to distinguish different people. Even though humans move in the same basic pattern there is a difference in the relative timing and magnitude of gait motion. Many of these variations have been studied in clinical gait analysis, which is used to distinguish pathological gait from normal gait and not used for identification of humans as shown by Abdelkader et al. [5]. A case described by Lynnerup and Vedel [6] shows how gait analysis was used to identify two bank robbers in Aalsgarde, Denmark. The results obtained from that particular analysis does not result into an evidence against the robbers, but it can be used as strong circumstantial evidence.

2.2 Advantages of Using Gait According to Previous Experiments

Human gait has several advantages compared to other more traditionally used biometric features. If the gait is measured using motion capturing, it is not possible to capture gait using a digital camera. Hence, the gait should be more difficult for anyone to forge and allow for a more secure way for authentication according to Pratheepan [12]. According to Freedman et al. [31] gait recognition does not require high quality images, good results can be obtained using in low resolution. Connie et al. [41] have studied the variation in height, girth and skeletal dimension which can provide a cue for recognition and make the authentication much more difficult to break. The paper also mentions that unlike fingerprints or retina scans, which requires cooperation with the users, the unobtrusive nature of gait makes it much more suitable for surveillance and security applications.

2.3 Disadvantages and Challenges of Gait According to Previous Research Experiments

Human gait has some disadvantages compared to the more traditional authentication methods. Its biggest weakness is that it is not stable when compared to other biometrics. A change in footwear or clothing can manipulate the gait to hinder a person from being correctly recognized by the system. According to Orwell [7], the gait differs when the person walks normally, runs and walks up and down stairs. Analysing a persons gait based on video footage can be used misused, since it is possible to recognize people without their knowledge or approval.

There are a different types of gait a person is able to perform. For simplification, this research focuses on when a person is walking normally, at normal speed along a flat surface. The average speed of walking for a person is 1.32 m/s, which gives an average of 60 gait cycles each minute Chan and Rudins [8]. Chang et al. [14] defines the human gait consists of many different elements which are characteristic for a person. The problem is to detect these features and analyse them in a way which can give a significant result. Tanawongsuwan and Bobicks [9] work mentions, one of the disadvantages with gait is, the gait changes during different walking speeds and over time.

2.4 Gait Analysis Systems

In the last decade gait has been introduced as a biometric feature. It can be divided into three different categories. Machine vision based, which uses video from one or more cameras, to capture gait data and video/image processing to extract features. Floor sensors, which are sensors installed in the floor and are able to measure gait features such as ground reaction forces and heel-to-toe ratio when a person walks on them. Wearable sensors, where the gait data is collected using body-worn sensors. This method is normally used to authenticate a person.

2.4.1 Floor Sensors

Gait recognition using floor sensors have shown good recognition rates. The magic carpet explained in Paradiso et al. [32] is a 16 x 32 grid of piezoelectric wires which are used to sense foot pressure and position. The LiteFoot mentioned by Fernstroem and Griffith [33] is another system developed in parallel with the magic carpet. It is a 1.76 square meter by 10 centimetres high floor element, filled with 1,936 optical proximity sensors. It detects the feet location by calculating the total impact force of the feet on the floor. The floor comes with an embedded micro-controller which scans all sensors at 100 Hz. The Z-tile system designed by McElligott et al. [34] has a modular design. The upper layer has an array of 20 pressure-sensitive elements, individually covered by carbon particles with sizes between 300 and 600 microns, while the inside of a tile houses micro-controllers and connections.

2.4.2 Wearable Sensors

The applications of wearable sensors are large and include clinical monitoring of humans, rehabilitation, motion analysis, athlete training, as well as security and authentication according to Ailisto et al. [37]. Wearable sensors can be a single accelerometer or a set of accelerometers together with a gyroscope to acquire data which can be analysed by biomechanics, and can be worn in different places on the body Sung et al. [36]. The data acquired is then used for classification of activities according to the application. The accelerometer in the palmtop computers and smart phones can be used to acquire data. The accelerometer is mainly attached to these devices to detect the orientation of the device and present the data accordingly. Currently these accelerometers are easily available and commercially in use this has attracted many researchers to employ them in securing different biometrics including voice, and finger prints which are already commercially in use. Ailisti et al. [37] were the first researchers to conduct an experiment, where portable technology was used for acquiring gait data using a portable accelerometer worn on the subject's waist.

2.4.3 Machine Vision

Machine vision is a commonly used gait recognition technique, because it allows the collection of gait features from a distance. Machine vision usually includes image processing techniques which are used to extract features like stride length which are determined by body geometry and body silhouettes. The machine vision-based gait analysis techniques can be classified as model-based [14, 15] and model free [16, 17, 18, 19].

The machine vision based gait analysis can also be categorized according to the technology, as marker-based and marker less. In the marker based system specific points on the subjects body are labelled by markers, these points are tracked in the video and the body motion is tracked and analysed as shown by Soriano et al. [21], Ailisto et al. [37]. Benabdelkader [81] used stride length as a feature and extracted it from 17 silhouettes walking in an outdoor environment for 30 meters at a fixed speed. They achieved an Equal Error Rate of 11 percent using linear regression for classification. Wang et al. [38] used the silhouette over time to characterize gait, by calculating the silhouette center and obtaining its contour they converted the 2 dimensional silhouette into a 1 dimension signal by calculating the distance between the centroid and every pixel on the contour. Principal component analysis was used for dimensionality reduction of normalized distance signals using normalized euclidean distance as a similarity measure and nearest neighbour classifier with respect to the Extended Nearest Neighbour (ENN) classification approach, achieved an Equal Error Rate (EER) of 20 percent, 13 percent, and 9 percent respectively.

2.5 Gait Authentication Using VTM and SVR

Recent work done using VTM and SVR, described the classification of gender from human gait using GEI (Gait Energy Image) [14]. In their experiment GEI is used as the distinguishing feature and the outcome achieved good performance in real time. In addition, Arias-Enriquez et al. [15] used this method in the medical field by presenting a fuzzy system to identify different human gait cycle anomalies during the phase of the cycle for knee and thigh using the sagittal plane. Muramatsu's [16] study proposes a gait based authentication technique that uses a random view transformation arrangement to decrease the accuracy drop due to view changes. A recent study by Iwashita et al. [17] shows promising results in gait recognition by considering changes in the subjects path.

2.6 Previous Work on Frontal Gait Analysis

A number of frontal gait recognition methods have been proposed in the past. The most popular are based on two dimensional frontal-view silhouettes, such as statistical shape analysis by L.Wang et al. [22]. This paper proposes an improved background subtraction procedure that is used for extracting silhouettes of a walking figure from the background. This method captures the structural characteristics of the gait, especially the shape of the body. The algorithm was tested of database consisting of 240 sequences from 20 different subjects with three viewing angles. They got Correct Classification Rate (CCR) using Extended Neighbour Classification (ENN) of 88.75 percent for zero degree viewing angle, 87.50 percent for forty-five degree viewing angle and 90 percent for ninety degree viewing angle.

One of the earliest works on frontal gait analysis was done by Goffredo et al. [29] in which he used a single non-calibrated camera and extracted unique signatures from descriptors of a silhouette's deformation. This approach was mainly designed for surveillance systems like CCTV cameras because the experiment involved the capturing of the upper-front view of the participants. They achieved a mean Correct Classification Rate (CCR) of 96.3 percent which proved, gait recognition of individuals observed from the front could be achieved without any knowledge of the camera parameters. Another interesting work was done by Lee et al.

[27] in which frontal motion analysis was done using a camera. In this experiment, various body parts are attached with a set of light sources which allowed the participants to be identified. The analysis was done through the phase-space analysis of the trajectories made by the light sources attached to various body parts. They achieved a correct classification rate of 85 percent. This approach was the first of its kind, where the trajectories of light sources were used for identification of people.

Soriano et al. [21] introduces curve spreads, an efficient descriptor of the front-view gait of humans walking towards the camera. Curve spread is a two dimensional representation of the time-variations of a moving body outline. The identification tests using the curve spreads yielded a 100 percent recognition rate for 50 frames per second and the recognition rate falls to 92 percent for 80 frames which is approximately two walk cycles. Barnich et al. [20] uses an intra-frame description of silhouettes which consists of rectangles that will fit into any closed silhouette. A dynamic, inter frame is added by aggregating the size distributions of these rectangles over successive frames.

A spatio-temporal approach for the front-view of gait recognition has been done by computing the human point cloud in a three dimensional spherical space by Chen and Gao [23]. They use the small soton gait database for testing which results into a recognition rate of 90 percent. Motivated by the kinect camera which makes it possible to create cloud points in three dimensional sphere and Cartesian spaces with a two dimensional based approach Kamata and Ryu [24] suggest adding depth information with the help of depth sensor. This helps in improving the overall recognition rate of the system by combining a three dimensional human point cloud with two dimensional silhouettes.

Chapter 3

Frontal Gait Analysis Using Depth Images

3.1 Frontal Gait Analysis

Many of the gait recognition approaches [3, 4, 5, 7, 18, 35] analyse side-view gait, for example walking on a plane parallel to camera. It is because most of the gait dynamics can be gathered as legs and hands when they extend to their maximum. However, it is shown in many of the forensic usages and security applications [6, 38] that it is difficult to acquire footages with side-view orientations. CCTV cameras are usually placed on the top corners of buildings, and the subjects pose is generally captured from an upper frontal-view. An example related to forensic analysis is mentioned by Lee et al. [27] which use frontal-gait footages from CCTV cameras for criminal investigation like the case of the bank robber in Noerager and the burglar in Lancashire (United Kingdom).

Another merit of frontal-view is that it only requires smaller physical space than the space needed in the commonly used side-view. For instance, this can be advantageous where an individual needs to verify his/her identity to enter a building or immigration checkpoints. In these type of situations, people need to line up and pass by a narrow space where the cameras/sensors are placed. In order, to capture 8 meters of walking distance in a side-view, a camera distance of a minimum of 9 meters is required, while in front view, only the corridor type of space is sufficient to capture the required gait cycle shown as by Lee et al. [27]. This potential merit explains the requirement of front-view in portal-based security authentication applications.

Nowadays smart-gates are installed at many airports. These smart-gates give travellers; who are eligible; the option to self-process through passport control. But some of these smart gates can have a few limitations. The main limitation of the iris and face-based authentications in these gates is that they need to capture the individual's image in near-field frontal-view, whereas the gait-based system does not require this. Hence, the gait, iris, and face-based coupled system is ideal for providing robust, near and distance field biometric authentication of an individual. However, coupling the gait with these existing implementations requires to place cameras where the frontal view can be captured.

Capturing the front view has it's own limitation. There is a need of compensation for looming effect and a possibility for self occlusion which can occur between the hands, legs and body. Only a small portion of the gait dynamics would be captured through two dimensional image data which can lead to poor performance for recognition. Based on the explanation by Sivapalan [31] three dimensional gait appearance based features are needed to be computed from frontal-view for robust gait recognition. But the three dimensional reconstruction will require multiple camera views. An alternative to acquire this data is to use a depth sensing device. A front based depth image has the advantage to capture all of the features of gait from a single point of view without the problem of self occlusion.

Depth images can be captured using google project tango. The tango tablet comes with a depth sensor, a laser and a camera through which the depth image can be seen and captured. Using Open CV library we can implement the depth map which can be used to capture the frontal view depth image of the user.

3.2 Depth Image Processing

3.2.1 How a Depth Sensor Works

Some depth sensors have a RGB (Red Green Blue) camera, some do not. Two crucial elements which must always be present for depth sensing: An IR (Infra-Red) projector, and an IR camera. The IR projector projects a pattern of IR light which falls on to the objects and forms a pattern of dots around the object. We cannot see the dots because the light is projected in the IR color range. IR camera can see those dots.

An IR camera is similar to regular RGB camera except the images are captured in the IR color range. The camera sends the video information of this distorted dot pattern into the depth sensor's processor, the processor works out depth based on the displacement of the dots. The pattern is spread out for the objects which are near the sensor and the pattern is dense for the objects which are far from the sensor. This depth map can be read from the depth sensor into your computer, or you can extract the information directly from the IR camera. When calibrating the RGBD tool kit, during the correspondence calibration phase, we must take a feed from both the depth map and the IR camera feed.

3.3 Depth Measurement by Triangulation

The Project Tango sensor consists of an IR laser emitter, an IR camera and an RGB camera. The laser emits a beam which splits into different beams by diffraction which results into a pattern of small dots projected onto the screen. IR camera captures this pattern and correlates it with a reference pattern. Reference pattern is obtained by capturing a plane, at a known distance from the sensor, and is stored in the sensor. When a small dot is projected on an object, whose distance to the sensor is smaller or larger than that of the reference plane, the position of the small dot in the infra-red image will be shifted in the direction of the baseline between the laser projector and the perspective center of the IR camera. The shift is measured for all small dots by a simple image correlation procedure, which gives a disparity image. For each pixel, the distance to the sensor can then be retrieved from the corresponding disparity.

3.4 Mathematical Model

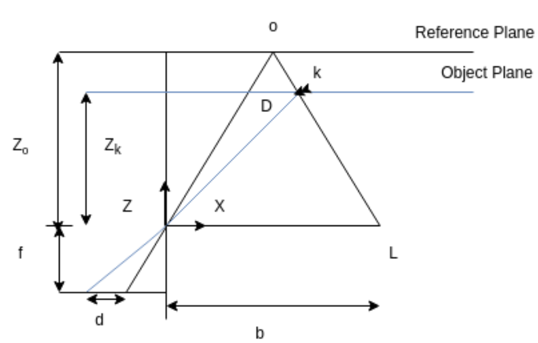

Figure 3.1: Relation between the distance of an object to the sensor relative to a reference plane and the measured disparity d.

To express the three dimensional coordinates for the object, a depth coordinate system is taken into consideration with the origin at the center of the infra-red camera [47]. Z axis is orthogonal to plane of the image in direction of the object, the X axis is perpendicular to the Z axis in the direction of the baseline b between laser and IR camera center, the Y axis is orthogonal to X and Z makes a right handed coordinate system. We assume an object is on the reference plane at a distance Z_o to the sensor, a small dot on the object is captured on the image plane of the IR camera. If the object is shifted closer or further away from sensor the location of the small dot on the image plane will be displaced in the X direction. This is measured in the image space as disparity d corresponding to a point k in the object space. From figure 3.1 using the similarity of triangles [45] we have:

$$D/d = (Z_o - Z_k)/Z_o \tag{3.1}$$

and

$$d/f = D/Z_k \tag{3.2}$$

where Z_k denotes the depth of point k in object space, b is base length, f is focal length of the IR camera, D is displacement of the point k in object space, and d is observed disparity in image space. Substituting D from Equation (3.2) into Equation (3.1) and expressing Z_k in terms of the other variables yields:

$$Z_k = (Z_o)/(1 + (Z_o/f_b)d) \tag{3.3}$$

Equation (3.3) is a mathematical model to derive the depth from the disparity observed, provided that the constant parameters Z_o, f_b, and d can be determined by calibration. The Z coordinate of a point, together with f, defines the imaging scale for that point. The planimetric object coordinates of each point can then be calculated from its image coordinates and the scale:

$$X_k = (-Z_k/f) * (x_k - x_o + dx) Y_k = (-Z_k/f) * (y_k - y_o + dy) \tag{3.4}$$

where x_k and y_k are the image coordinates of the point, x_o and y_o are the coordinates of the principal point, and dx and dy are corrections for lens distortion.

Chapter 4

Implementation

This chapter deals with the various aspects of implementation, such as the experimental set-up, the software functions used for the implementation and the data collection methods incorporated in order to test and verify the proper working of the depth map application.

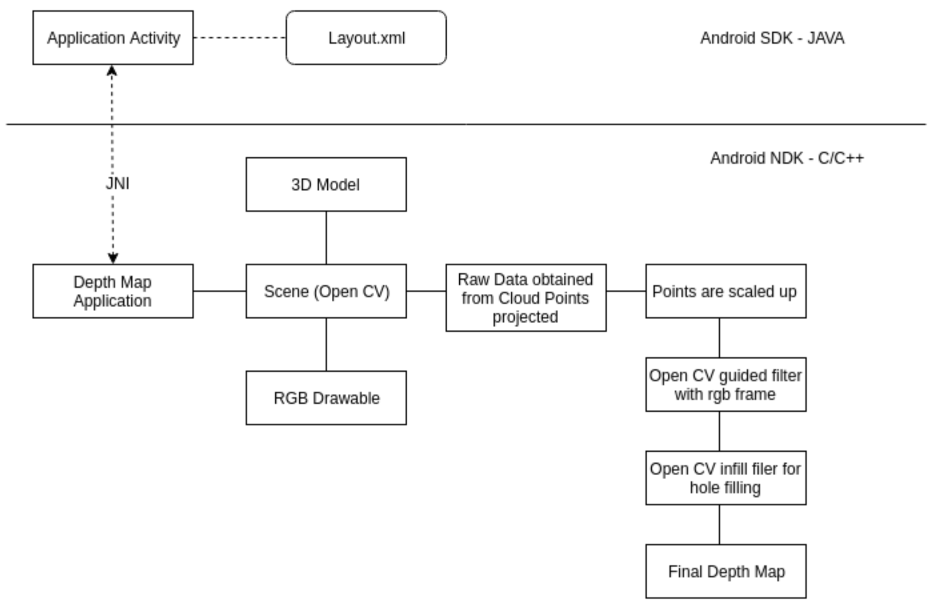

Figure 4.1: Architectural overview of the application implemented using project tango

4.1 Depth Map Application Overview

Tango is a platform which uses computer vision which gives mobile devices the ability to understand the position relative to the world around them. Tango tablet is an android device with a wide angle camera, a depth sensing camera, accurate sensor. Tango offers API in c, java and an sdk for unity. Tango gives the mobile devices to understand the position by using three core technologies: motion tracking, area learning and depth perception. For our application we need the use of depth perception

The depth map application has two sections. The first section is the android sdk for java. This part of the application consists of the user interface which consists of picture in picture. One is the actual image of the participant and other is the depth map of the participant. There is a button present when pressed starts storing the depth image of the participant. When the button is pressed a call to function present in the second section is initialized. The second section is the android ndk for c/c++ also called native android. Both the sections are connected through the Java Native Interface (JNI) library which maps the java functions into c/c++ and vice versa. When the call to native android is made, the raw cloud points collected by the depth sensor are projected onto the screen. In order to increase the number of raw data points another call to the depth sensor is made to increase the number of points getting projected onto the screen. Once this is finished we use scene module from the Open CV library [26]. In order to preserve the edges of the participants body we have to use guided filtering which is an implementation of guided image filtering in Open CV. The first filtering is done for the RGB frame captured and later for further better quality of the depth map another type of guided filter called as infill filter is used in order to decrease the noise in the image further.

Depth perception gives the application ability to understand distance to objects in the real world. In our application's case it is able to detect the participants when they walk towards the device. Application uses the depth sensor along with the infra-red sensor to get the distance between the participant and the device. When the distance is more than five meters which is the maximum range of the depth sensor, the participant is not visible to the depth sensor. As the the participant comes within the range the depth sensor starts to detect the participant with the help of the infra-red sensor which is directly projected towards the participant. When the distance decreases more depth information can be extracted from the participant. For the implementation of the application we have to integrate the Open CV library. Since the depth part is not populated as an API by google we have to integrate the library and add it to the path of the android compiler.

4.2 Open CV

Open CV [26] library is used to implement the depth image on Google Project Tango. Open CV is a computer vision library with functions developed by Intel mainly for real-time computer vision. The primary interface of OpenCV is in C++ with additional interfaces in Matlab, Java and Python. Some of the areas in which Open CV is widely used are:

1. 2D and 3D feature tool kits
2. Facial Recognition
3. Gesture Recognition
4. Human Computer Interaction (HCI)
5. Object Identification

All of the OpenCV classes and functions are placed into the cv name-space. Therefore, to access this functionality from the code, use the cv:: specifier or using name-space cv. We primarily use three classes within the Open CV library for the implementation of the application.

Core :- This is one of the building blocks of the Open CV library. This class is primarily used for manipulation of the images at the pixel level. We use the basic structures like Mat, Scalar and Mat::depth for creating the basic image matrix for the depth map.

Mat::depth returns the depth of a matrix element. The method returns the identifier of the matrix element depth.

Mat represents an n-dimensional numerical single or multiple channel array. It is used to store real or complex value vectors and matrices. This stores the cloud points which are projected by the depth sensor. The information from these cloud points are used for creation of the depth map.

Imgproc :- This is part of the image processing module and it is used for smoothing the edges of of the objects which are projected in the depth map. Geometrical image transformations maintain the natural shapes of the objects in the depth map.

Erode the function erodes a source image using the specified structuring element that determines shape of a pixel over which minimum is taken. The function supports the inplace mode. This method is used for smoothing out the edges of the objects so that there is no loss of information.

cvtColor the function converts an input image from one color to another. In case of transformation to-from RGB color space, the order of the channels should be specified explicitly. This function starts the first step which leads to formation of the depth map. The actual image is first converted into a gray scale image while the depth information is being collected. This step is achieved via cvtColor.

Highgui :- This provides an easy interface to create and manipulate windows that can display images and read and write images to/from disk or memory. We use the imwrite function from this class.

Imwrite helps in saving the depth image to a specified file. The file can be saved in JPEG, or PNG formats. In this thesis we store the image in JPEG format. The parameter provided to these functions are name of the file, the array or matrix of the image and the format in which the image is to be stored.

In order to use the depth perception using Open CV we need to attach an XYZijAvailable callback. A callback function is used to inform a class that some work is being done in another class. XYZijAvailable callback is used for allocating memory and processing of the image buffer which contains the pixel values of the depth map. The callback is called each time a new depth data is available. On the Tango tablet, the depth callback occurs at 5 Hz. An optional argument following the callback pointer can be supplied and can be returned in the callback context parameter.

4.3 Experimental Set-up

Google Project Tango was used for the implementation of the depth map application. The device is based on the Android Operating System. The device is equipped with three dimensional sensors that measure the distance from a device to objects in the read world. Current devices are designed to work best indoors at moderate distance up to 4 meters. The depth data allows an application to understand the distance of visible objects to the device. The depth information is returned to the device in two main formats Point Clouds and a device specific format called XYZ_{ij} designed to aid meshing. In the experiment we use the XYZ_{ij} device specific format. It is a combination of an XYZ point cloud and a 2-D lookup table which allows the generation of a depth map. But since the ij part is not populated as an API we could not use it directly, hence we had to integrate an Open CV library for our experiment. For that reason, we had to transform the depth information into an image like depth map (similar output found in microsoft kinect).

Figure 4.2: Google Project Tango

4.4 Implementation of Depth Map

In order to implement Depth Map, we first had to integrate the native part of Open CV to the C/C++ part of the code. Microsoft kinect has its own version of Open CV which can be directly used for implementation of depth map. But in case of tango tablet the device does not have an implementation of Open CV which can be used directly. In order to use Open CV we have to manually build our own version by downloading the C/C++ variant of Open CV library and adding it to the path of the android ndk compiler.

To integrate we had to build our own variant similar to that described in documentation available for using Open CV in android [28]. In order to build our own variant, the library files for the c/c++ version has to be downloaded from the official Open CV website. The library is to be added to the main depth map application folder. Path of the library has to be included in the Android.mk file so that the compiler can retrieve the library. When the application starts, the depth camera information is obtained from the functions which

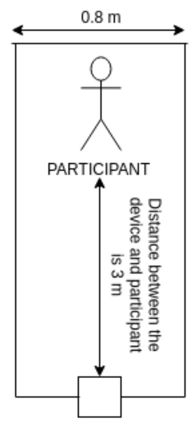

Figure 4.3: The figure shows the set-up of Google Project Tango in the experiment. Since the range of the depth sensor is about four to five meters, to be on the safe side we kept the initial distance between the participant and the depth sensor as three meters. This is the maximum distance between the participant and the device. The participant covers approximately 2.3 meters. After that the collection of data is stopped the width of the area is 0.8 meters.

are already populated by Google API for depth camera intrinsics. In order for proper implementation of the depth map we have to check some paths and files.

Check all of the Open CV related paths in Android.mk which need to be set-up in order to your build path.

Open CV library as gradle-java project in the gradle dependencies.

Check the obj folder for an available libopencv-java3.so file after build to check if the library inclusion is working.

4.5 Data Collection

The data was collected in an enclosed environment in a conference room. The reason being, the sensitivity of depth sensor to light as well as the distance limitation of the depth sensor. We recorded the gait cycles of five different participant whose body shape were different. Every participant was asked to walk three times. We asked the participants to walk three different times because we wanted to incorporate the variance in the walking style for the same participant.

Every cycle consisted of seven samples. These samples were depth images captured while the participant walked towards the depth sensor in the device. The total distance covered by every participant in every cycle is approximately three to four meters. The distance varied with every participant because of the stride of every step for each participant is different.

Figure 4.4: The figure represents the seven samples of the depth image captured for Participant 3. These seven samples combine together to form one complete cycle for each participant. Similarly all the participants were asked to completed three cycles in order to accommodate any variation in the style of walking for same participant across different cycle.

There were two different scenarios: the distance between the device and the participants are same in both the cases, but in the first case more information about the upper body was captured and in the second scenario more information about lower part of the body was captured.

This was done to observe from which part of the body can the maximum number features be extracted from the authentication perspective. After observing the different gait cycles of participants, we decided to use the gait cycles which concentrated more on the lower part of the body. After observing the depth images we came to the conclusion that the upper body shape of a participant can vary. This is substantiated by previous research [16, 38, 40] has shown that authentication systems have failed to recognize a person if the person is carrying an object in hand. Also some systems have failed to recognize the correct person if a person is wearing a jacket. Whereas the lower part of the body is free from all of these variations. It was observed that we could extract features like the distance covered, the swing angle of the feet, maximum angle between the two legs, and the difference in the pixel intensity values from the lower part of the body.

Chapter 5

Analysis of Gait Images Captured

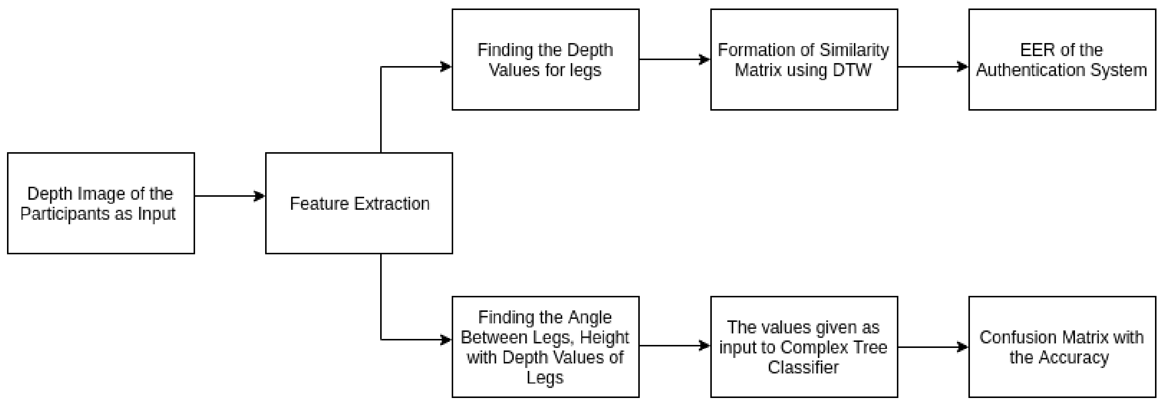

Figure 5.1: Block diagram of the approach. We give depth image as the input for the authentication system. For EER we give the depth values for both legs as the input and the EER is calculated from the similarity table calculated using DTW over depth values. For classification we use the angle between the legs and height along with the depth values of the legs.

Three cycles were captured for every participant and every cycle consists of seven samples. Each sample is a depth image of the participant walking towards the device.

In order to distinguish each user we are extracting three features which are unique to each participant. The features are, the depth values of the legs, the maximum angle between the two legs and height of the person. The first two features are dependent on the lower part of body and the height is from the whole body. We tried classifying every participant using two different techniques, one was using only the depth values of the legs. The second is done using all three features.

5.1 Experiment with Depth Values

The first experiment for differentiating each participant was done using only the depth values for left and right legs. In order to extract the maximum depth information we divided each

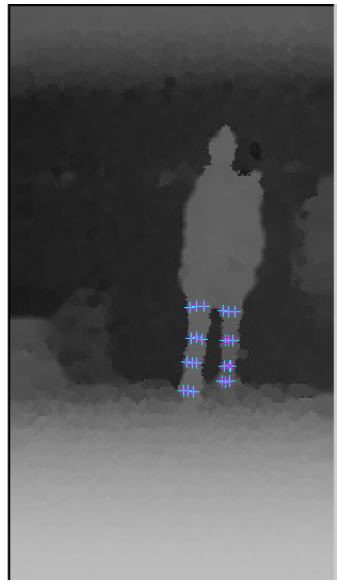

Figure 5.2: The leg is divided into four parts since we needed to extract the maximum information of the change in the depth value for each part for every sample. We have used the Image Processing Toolbox available in MATLAB to divide both leg and right leg into four different parts which in total gives us eight features based on the classification for every participant.

leg into four different parts, so effectively we were getting eight features in total. To do the analysis, we have used Matlab and its applications for extracting the features and classifying the participants.

We used Dynamic Time Warping (DTW) to find the relationship of the depth information extracted from every participant. DTW compares the depth values for both tables from which we make a similarity matrix used for calculating the Equal Error Rate (EER). For every participant multiple points were marked on every part of the leg and then the average value was taken as the depth value for that part of the leg. This process was for every part of the leg done manually using Matlab Impixel command. For reference purpose we also took the depth value for the upper part of the body. All of the depth values for the legs are then subtracted from the depth value of the upper body. This is done to normalize the data and bring it within a specific range. These subtracted values were then given as input for the DTW algorithm which is used because it measures the similarity between two temporal sequences, in this case the depth values that vary with time.

5.1.1 Dynamic Time Warping

DTW is a time series algorithm originally developed for speech recognition. The algorithm aligns two sequences of feature vectors by warping the time axis iteratively until an optimal match between the two sequences is found.

Consider two sequences:

$$A = a_1, a_2, ..., ...a_n \tag{5.1}$$

$$B = b_1, b_2, ..., ...b_n \tag{5.2}$$

Each cell has a distance measure, we get this distance value by comparing the corresponding elements of the two sequences. In order to find the best match between two sequences we need to find a path through the grid which results in a minimum distance between them. To find the overall distance we have to find all possible combinations of routes through the grid and for each one compute the overall distance. The overall distance is the minimum of the sum of the distances between the individual elements on the path divided by the sum of the weighting function. The weighting function is used to normalise for the path length. For any considerably long sequences the number of possible paths through the grid will be very large.

The results from DTW are stored in a similarity matrix, which gives us a view of how close the depth values are when compared to different participants. When the comparison is between the same participant for the same cycle we get zero, hence all the diagonal values in the matrix are zero. The remainder of the values show how different participants depth values are close to each other.

5.2 Experiment with Depth Values, Height and Angle Between the Legs

In this experiment we try to incorporate more features which can be used for classifying different participants. One of the key feature is the angle between the legs. The angle between the legs for each participant is different because of the step size every participant takes. We observed that the smallest angle is for the participant who is shortest among all the participants and the angle is greatest for the participant who is tallest among all the participants. The initial approach was to only calculate the maximum angle between

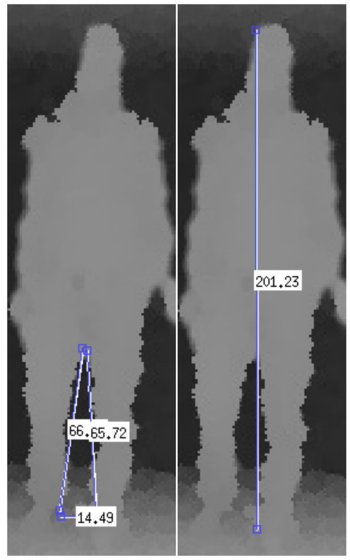

Figure 5.3: In order to measure the angle between the legs impixel command in MATLAB is used. This command enables us to measure the distance from waist to foot for each leg. Once the distance is obtained we apply the cosine rule to find the approximate angle between the leg in degree. The command also enables in measuring the height for every participant.

the legs for every participant. But, after observing the resulting angles, the range between which this angle varies is also different for every participant and hence we need to consider angles for every sample in every cycle.

To calculate the angle we take the sample, we form a triangle using the two legs and the distance from the lower half of the body to the floor, by using the sin and cosine rules we calculate the angles between the legs.

For figure 5.3 the angle will be:-

$$cos(A) = (66^2 + 65.75^2 - 14.49^2)/(2 * 66 * 65.72) \tag{5.3}$$

$$cos(A) = 0.9892 \tag{5.4}$$

$$A = cos^-1(0.9892) \tag{5.5}$$

$$A = 10.89^o \tag{5.6}$$

To extract the second feature, the variance in the height of every participant, we calculate the height for every participant for every sample in every cycle. From the data collected we observed, each participant had a minimum and maximum value between the value of height varied. Though there was some difference in values of height for every person for same sample number in different cycles, the variance was not significant.

To calculate the height of each participant, we measured the distance from each person's head to the floor. We used the image viewer tool in Matlab for calculating both the height and angle between legs for every participant.

We concentrated on the legs because, previous research [10, 21, 24, 27, 31] has proved the recognition rate is affected when upper body parameters like the swing of the arms, width of the shoulders are affected by the apparels of the users hence we did not take the upper body into consideration. Another parameter which we did not consider is the shape of the body which is a commonly used parameter in side view gait authentication [3, 5, 8]. We did not use it because after observing the depth image we saw some of the images had noise which did not show proper body shape of the participants. Initially we tried incorporating the direction in which the foot of the participants was pointing but that could be observed only when the participants were very close to the depth sensor and some of the participants had their foot pointing in the same direction which could have affected the over all recognition rate of the system. One parameter which we could have included is the distance covered by different participants while they were walking towards the depth sensor. But we could not derive a relationship between the depth values and the distance covered by the participants. If we can find this relationship, we can use it as a feature for improving the recognition rate of the system.

Chapter 6

Results

This chapter presents the results obtained for calculation of EER for depth values and classification with height, angle between the legs and depth values. The first part of the results will discuss how we calculated the EER, which shows the efficiency of our system when only depth values of the left and right leg is taken into consideration. The second part of the results consider the angle between the legs and height of the person along with the depth values of legs for each participant. These features are given as input to a machine learning in order to see the accuracy of the system and to differentiate between different participants.

6.1 Equal Error Rate(EER) for Only Depth Values

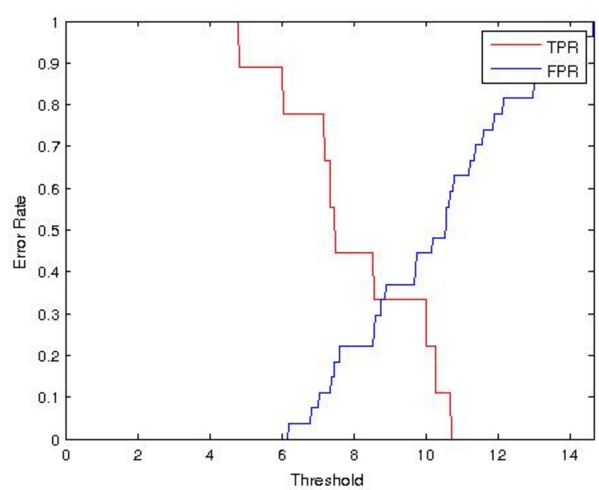

Figure 6.1: The ROC plot for the system with EER of 0.333 and the optimal threshold of 8.7352. The value of TPR at the crosspoint is 0.3313 and the value of FPR is 0.3313.

We created a table for every participant and added all of the depth values for both the legs inside this table. In order to normalize the data and to bring it within a given range, we take the difference between the upper body depth value and the legs depth value. The value for the upper body was taken in a similar manner as described in Chapter 6. In the upper body case we took the average value for the upper part of the body rather than dividing it into four different parts. As described in Chapter 6, these values are further processed by DTW and a similarity matrix is obtained. These values are further used for obtaining the EER to see how much error is present in the system for the given data.

The blue line in the graph below denotes the False Positive Rate (FPR) and the red line denotes the True Positive Rate (TPR). At the intersection of both of these curves we obtain an EER. For the system the EER value comes out to be 0.3333.

FPR is generally stated a percentage at which an imposter is accepted as authentic by the biometric system. TPR is generally stated as percentage at which the correct person is authenticated by the biometric system. For our biometric system the TPR value is 0.3313 and FPR value is 0.3313.

The EER of 0.3333 means that, the proposed gait authentication system takes about 33 out of 100 decisions that are wrong. This means if an intruder tries a brute force attack with a large number of different gait samples, 33 out of 100 attempts will succeed on average. EER is usually used as an indicator of the system's performance. In our case the EER is high when compared to the EER of approaches where the side view is used for authentication [6, 12 13]. On of the key reasons for such a low EER is the resolution of the depth sensor. Previous experiments done with a depth sensor for the front view gait analysis [21, 22, 30] use devices with a resolution much higher than that of the project tango tablet. If the resolution of the depth sensor is improved the depth values returned from the legs will be much more accurate resulting in a lower value of EER. Another reason is that the points projected by the laser from the depth sensor on the participants, the number of points projected is almost $1/3^{rd}$ of the total number of points it can project. This is also another hardware limitation of the device. Because of the limited number of points at times we did not get the proper shape of the participant which could have been used as a feature to distinguish different participants. Another way of improving the EER is to add more features which can be extracted from the front view of the gait. We have extracted in total eight features for all the participants, but a few more features would decrease the value of EER further.

6.2 Classification with Height and Angle Between Legs with Depth Values

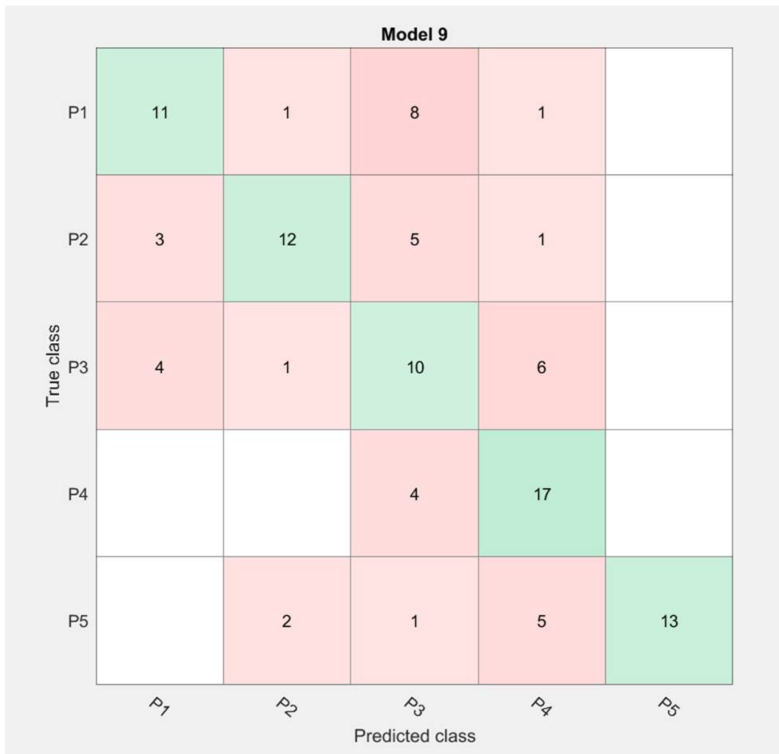

Figure 6.2: Confusion Matrix for Complex Tree Classifier. A confusion matrix is a table that is often used to describe the performance of a classification model (or classifier) on a set of test data for which the true values are known. In this case for P1, out of the 21 samples for P1, 11 were correctly identified and next closest was P3. We obtained the best classification result for Participant P4 where 17 were correctly identified.

In this section we will discuss the results obtained from the classification of the features which are used for authentication of each participant. We use the MATLAB application called classification learner. The data is provided to this application which comes with different classifiers.

For classification we use 2 fold cross validation since this model evaluation method is better than residuals [46]. We use 2 fold cross validation because we did not have a separate training dataset and testing dataset. Hence 2 fold cross validation helps in training and testing the system for classification.

The problem with residual evaluations is that they do not give an indication of how well the learner will perform when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on new data. This is the basic idea for a whole class of model evaluation methods called cross validation.

We use 2 Fold Cross Validation which is K-Fold Cross Validation. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. In our case we use the Complex Tree Classifier which gives us an accuracy of 60 percent.

The accuracy in this case is calculated by taking a ratio of true positive to total the sum of the true positive and false positive. From the confusion matrix in Figure 6.2 we can obtain the true positive and false negative value. The values in the green box are the true positive values and the values in red are the false positive values. In our case there is no true negative and hence the white boxes in the confusion matrix indicates that. Hence, by this definition we can calculate the accuracy for our authentication system using the confusion matrix in figure 6.2 all the values in green are true positive and all the values in red are false positive.

True Positive(TP) = 11 + 12 + 10 + 17 + 13 = 64

False Positive(FP) = 10 + 9 + 11 + 4 + 8 = 42

Accuracy = (TP)/(TP + FP) = 60 percent

The accuracy for our gait authentication system is 60 percent, which when compared with other approaches [21, 22, 29, 31, 40] is less. However we only extracted the basic features. The amount of pre processing required in our case is less because we don't have to extract the human silhouette from the image. The previous approach [21, 22, 29, 31, 40] does not take depth information into consideration. Though our accuracy is less, we can improve this by including more features. For example,we could include the distance covered by each participant while the data is recorded. If we can find the relationship between the depth information obtained and the distance covered that could be used as a feature for classification. One such approach [22] uses the body shape as a distinguishing feature because of the low depth sensor resolution. However, with this approach the proper body shape of the person is lost in some cases and could not be included as a feature for classification. The accuracy is also affected by the variation which occurs in the behaviour of the participants while walking. Also the low resolution of depth sensor results into noise in depth image, if this noise can be removed the accuracy of the will improve further.

Chapter 7

Conclusion

This thesis provides a way in which a person can be authenticated using gait as a biometric feature. We implemented the gait authentication system by using the depth information, height and angle between the legs which is obtained from the depth image of the participants.

Three cycles were captured for every participant to take into account the variance of gait for the same participant. Every cycle had seven samples where we try to see the change in depth value as the participant comes closer to the device. Two different experiments are performed, the first one was on the depth values, where we have concentrated on the lower part of the body, with both legs divided in four parts giving us a total of eight features. These features are then normalized and given to DTW from which we get a similarity matrix which is used to determine EER of the system. The EER for our system is 0.33.

In second experiment we have extracted additional features such as the angle between the two legs and the height in every sample for all three cycles. We need more features to improve the accuracy of the system to distinguish different participants. Here we have collected the data and used the classifier available with MATLAB. One crucial observation was that the accuracy changed with the way we organized the dataset.

Frontally captured data has many advantages over a lateral view for gait based biometric authentication, including easy integration into biometric portals and similar devices, as well as not having field of view issues in confined spaces such as a narrow corridor. The addition of depth also enables more data to be captured than from the side, as there is no issue of self-occlusion.

References

[1] Tricia Olsson. Strengthening Authentication with Biometric Technology. Technical report, SANS Institute, August 2003.

[2] A.A. Marks. Manual of Artificial Limbs. New York, 1905

[3] James E. Cutting and Lynn T. Kozlowski. Recognizing Friends by their Walk: Gait Perception Without Familiarity Cues. Bulletin of the Psychonomic Society, 1977.

[4] Sourabh A. Niyougi and Edward H. Adelson. Analyzing and Recongizing Walking Features in XYT. In IEEE Computer Society International Conference on Computer Vision and Pattern Recongition, pages 469474, June 1994.

[5] Chiraz BenAbdelkader, Ross Cutler, and Larry Davis. Stride and Cadence as a Biometric in Automatic Person Identification and Verification. In IEEE International Conference on Automatic Face and Gesture Recognition. Microsoft Research, 2002.

[6] Niels Lynnerup and Jens Vedel. Person Identification by Gait Analysis and Photogrammetry. Journal of Forensic Science, 50(1), January 2005. Technical Note.

[7] George Orwell. 1984. Plume, Centennial Edition edition, May 2003.

[8] Carl W. Chan and Andrew Rudins. Foot Biomechanics During Walking and Running. Mayo Clinic Proceedings, 69:44861, 1994.

[9] Rawesak Tanawongsuwan and Aaron Bobick. Performance Analysis of TimeDistance Gait Parameters under Different Speeds. In 4th International Conference on Audio- and Video Based Biometric Person Authentication, Guildford, UK, June 2002.

[10] Sudeep Sarkar, P., Jonathon Phillips, Zongyi Liu, Isidro Robledo Vega, Patrick Grother, and Kevin W. Bowyer. The humanID gait challenge problem: Data sets, performance, and analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(2):162177, 2005.

[11] Davrondzhon Gafurov, Einar Snekkenes, and Patrick Bours, Gait authentication and identification using wearable accelerometer sensor, In 5th IEEE Workshop on Automatic Identification Advanced Technologies (AutoID), pages 220225, Alghero,Italy, June 7-8 2007.

[12] Y. Pratheepan, J. V. Condell, G. Prasad. The Use of Dynamic and Static Characteristics of Gait for Individual Identification. In Proceedings of 13th International Machine Vision and Image Processing Conference, Dublin, Ireland, 24 September 2009; pp.111116.

[13] W. Kusakunniran, Q. Wu, J. Zhang, H. Li. Support Vector Regression for Multi-View Gait Recognition Based on Local Motion Feature Selection. In Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 1318 June 2010; pp. 974981.

[14] P.C. Chang, M.C. Tien, J.L. Wu, C.S. Hu. Real-Time Gender Classification from Human Gait for Arbitrary View Angles. In Proceedings of 2009 11th IEEE International Symposium on Multimedia, San Diego, CA, USA, 1416 December 2009; pp. 8895.

[15] O. Arias-Enriquez, M.I. Chacon-Murguia, R. Sandoval-Rodriguez. Kinematic Analysis of Gait Cycle Using a Fuzzy System for Medical Diagnosis. In Proceedings of 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), Berkeley, CA, USA, 68 August 2012; pp. 16.

[16] D. Muramatsu, A. Shiraishi, Y. Makihara, Y. Yagi. Arbitrary View Transformation Model for Gait Person Authentication. In Proceedings of 2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, 2327 September 2012; pp. 8590.

[17] Y. Iwashita, R. Kurazume, K. Ogawara. Expanding Gait Identification Methods from Straight to Curved Trajectories. In Proceedings of 2013 IEEE Workshop on Applications of Computer Vision (WACV), Tampa, FL, USA, 1517 January 2013; pp.193199.

[18] R.C. Jain, R. Kasturi, B.G. Schunck. Machine Vision. McGraw-Hill: New York, NY, USA,1995.

[19] R.R. Jensen, R.R. Paulsen, R. Larsen, Analyzing Gait Using a Time-of-Flight Camera. In Image Analysis; Salberg, A.B., Hardeberg, J.Y., Jenssen, R., Eds.; Springer: Berlin, Germany, 2009; pp. 2130.

[20] O. Barnich, M. Droogenbroeck, Frontal-view Gait Recognition by Intra- and Inter-frame Rectangle Size Distribution, Pattern Recognition Letters, vol. 30, no. 10, pp. 893901, July 2009.

[21] M. Soriano, A. Araullo, C. Saloma, Curve Spreads-A Biometric from Front View Gait Video, Pattern Recognition Letters, vol. 25,no. 14, pp. 15951602, October 2004

[22] L. Wang, T. Tan, W. Hu, H. Ning. Automatic Gait Recognition Based on Statistical Shape Analysis, IEEE Transactions on Image Processing, vol. 12, no. 9, pp. 1120-1131, September 2003.

[23] S. Chen, Y Gao, An Invariant Appearance Model for Gait Recognition, Proc. IEEE Conference on Multimedia and Expo, pp. 1375-1378, Beijing, China, July 2007.

[24] J. Ryu, S. Kamata, Front View Gait Recognition using Spherical Space Model with Human Point Clouds, Proc. International Conference on Image Processing, pp.3209-3212, Brussels, Belgium, September 2011

[25] M.O. Derawi, P. Bours, K. Holien. Improved Cycle Detection for Accelerometer Based Gait Authentication. In Proceedings of 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Darmstadt, Germany, 1517 October 2010; pp. 312317.

[26] https://developers.google.com/project-tango/overview/depth-perception

[27] K. Lee, S. Ranganath, S. Sanei, Frontal View-based Gait Identification using Largest Lyapunov Exponents, Proc. International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 173-176, Toulouse, France, May 200 default.asp

[28] http://code.opencv.org/projects/opencv/wiki/BuildingOpenCV4Androidfromtrunk

[29] M. Goffredo, J. Carter, M. Nixon, Front-view Gait Recognition, Proc. IEEE International Conference on Biometrics: Theory, Applications and Systems, Washington, DC, USA, September 2008

[30] Freedman, B.; Shpunt, A.; Machline, M.; Arieli, Y. Depth Mapping Using Projected Patterns. U.S. Patent 2010/0118123, 13 May 2010.

[31] Sabesan Sivapalan. 2014 Human Identification from Video Using Advanced Gait Recognition Techniques

[32] Paradiso, J., Abler, C., Hsiao, K.-y., Reynolds, M. 1997. The magic carpet: physical sensing for immersive environments. In CHI 97 extended abstracts on Human factors in computing systems: looking to the future, CHI EA 97, 277278, New York, NY, USA. ACM

[33] Fernstroem, M. Griffith, N. 1998. Litefoot - auditory display of footwork. University of Glasgow, U.K. British Computer Society, British Computer Society

[34] McElligott, L., Dillon, M., Leydon, K., Richardson, B., Fernstrm, M., Paradiso, J. 2002. Forse fields - force sensors for interactive environments. In UbiComp 2002: Ubiquitous Computing, Borriello, G. Holmquist, L., eds, volume 2498 of Lecture Notes in Computer Science, 321328. Springer Berlin / Heidelberg

[35] Orr, R. J. Abowd, G. D. 2000. The smart floor: a mechanism for natural user identification and tracking. In CHI 00 extended abstracts on Human factors in computing systems, CHI EA 00, 275276, New York, NY, USA. ACM

[36] Sung, M., Marci, C., Pentland, A. 2005. Wearable feedback systems for rehabilitation. Journal of NeuroEngineering and Rehabilitation, 2(1), 17

[37] Heikki J. Ailisto, Mikko Lindholm, J. M. E. V. Makela, S.-M. 2005. Identifying people from gait pattern with accelerometers. Proc. SPIE ; doi:10.1117-12.603331, Vol. 5779, pp.714.

[38] Hu, W., Tan, T., Wang, L., Maybank, S. 2004. A survey on visual surveillance of object motion and behaviors. IEEEJSMCC, 34(3), 334352

[39] Hazem Ali. 2011 A report on Gait Recognition using Time of Flight Sensor

[40] Gait Analysis using a Single Depth Camera Minxiang Ye, Cheng Yang, Vladimir Stankovic, Lina Stankovic, and Andrew Kerr Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, GXW, UK Biomedical Engineering Department, University of Strathclyde, Glasgow, GXW, UK

[41] Tee Connie1, Michael Kah Ong Goh1, Andrew Beng Jin Teoh Multimedia University . 2013 Gait Recognition using Sparse Grassmannian Locality Preserving Discriminant Analysis

[42] Johansson, G Visual Perception of biological motion and a model for its analysis. Perception and Psychophysics, 1973, 14, 201-211

[43] Davrondzhon Gafurov A Survey of Biometric Gait Recognition: Approaches, Security and Challenges 2007

[44] Prerna Arora, Rajni Survey on Human Gait Recognition International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015

[45] www.mathopenref.com/similartriangles

[46] www.salford-systems.com/videos/tutorials/how-to/an-introduction-to-cross-validation

[47] Wolfgang Paier Acquisition of 3-D Head Models using SLR Cameras and RGBZ Sensors