TECHNISCHE UNIVERSITÄT
CHEMNITZ

Estimation of Gait Parameters based on 3D Human Skeleton

Research Project

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc. Embedded Systems

Dept. of Electrical Engineering and Information Technology
Chair of Digital Signal Processing and Circuit Technology

Submitted by: Mr. Ankit Jaiswal
Matriculation No.: 479899
Date: 08.02.2020
Supervising tutor: M.Sc. Jyothsna Kondragunta
Examiner: Prof. Dr.-Ing. Gangolf Hirtz

Acknowledgements

I would like to express my sincere gratitude to M.Sc. Jyothsna Kondragunta for providing me with an opportunity to work on the "Estimation of Gait Parametes based on 3D Human Skeleton" project and also for the constant supervision and support during the entire project work.
I am thankful to Prof. Dr.-Ing. Gangolf Hirtz for being my supervisor. I would also take an opportunity to thank Dipl.-Ing. Frank Klöpfel for his constant support in providing proper resources. I would like to thank the Chair of Digital Signal Processing and Circuit Technology for providing me with all the facilities required for the completion of this project. I am highly indebted to the staff of the Department of Electrical Engineering and Information Technology, Technische Universität Chemnitz for their valuable support rendered for the successful completion of this project.

Abstract

Independent mobility is essential for elderly people for their routine activities like walking, standing, getting up, turning, leaning etc. These independent mobilities also called as gait depends on the nervous, musculoskeletal and cardiorespiratory systems. Each person's gait pattern is influenced by lot of factors like mood, personality, age and largely health condition. In order to understand the health condition of an elderly person, analysis of gait patterns is an important aspect. Gait parameters such as cadence, step length, step duration etc. analysis has proved as an important factor in estimation of the healthy daily living. Problems with gait characteristics can lead to different neurological diseases.

The primary goal of this project is to provide a low-cost, easy-to-use solution for capturing and analysis of human gait with the use of Computer Vision techniques. The targeted application area is diagnostics and modelling of motion impairments common for neurological diseases such as Parkinson's or Alzheimer.

In this project, a simple methodology for recording and computing gait parameters has been developed as a part of the present studies. To fulfil the requirement of low-cost solution, motion capture is performed by the Kinect motion sensor and analysis of human gait patterns and parameters is obtained using computer vision techniques. The extracted gait parameters will be used in future to assess the health condition of the individual.

Keywords: 3D pose estimation, gait parameter estimation, gait analysis, elderly persons

Contents

List of Figures

List of Tables

List of Abbreviations

RGB Red, Green, Blue

PAFs Part Affinity Fields

JSON JavaScript Object Notation

2D 2-Dimensional

3D 3-Dimensinal

CNNs Convolutional Neural Networks

CMPs Convolutional Pose Machines

HMM Hidden Markov Model

wrt with respect to

PCA Principal component analysis

SVM Support Vector Machine

NLDR Nonlinear dimensionality reduction

VGA Video Graphics Array

DLN Deep Linear Network

1 Introduction

Currently, many countries are experiencing a substantial shift in age demographics, with a rising proportion of the elderly population. With the increase in elderly population, there is an increase in handling their health issues. In handling health issues, especially, complex condition such as dementia needs an extreme care. It is observed from the literature that dementia, age and gait deviations are tightly coupled [1, 2] . Dementia is considered to be the one of the most common age-related neurodegenerative disorder. It is estimated that around 50 million people worldwide have dementia [3]. Many scientific researches prove that gait abnormalities can be seen in people with dementia [4]. This is due to the varying combinations of hypokinesia, rigidity as well as from the defects of posture and equilibrium. This includes the characteristics of shuffling gait with small steps and poverty of movements in the trunk. Due to this reason, it is evident that observation of gait in a long term can be advantageous in giving a better care to the elderly people.

While walking, older people requires more attention for the motor control when compared to a younger adult [5]. Falls, often with serious injuries and/or psychosocial consequences can be the outcome. Apart from falls, other age-related health issues can also be a reason for the gait abnormalities. In elderly, when the gait is no longer automatic, any other tasks performed during a walk can lead to a gait disturbance and in worst-case scenario resulting in fall. Detection of such gait disorders not only allows the timely intervention of individually tailored gait improvement strategies but also helps to identify the health-related issues such as dementia in advance [6]. Gait analysis is a well-researched field and has diverse application in sports, rehabilitation and health diagnostics [7]. During assessment of patient, Clinicians can measure various gait parameters. For illustration, Clinicians can use a person's gait speed [8], gait symmetry [9], gait endurance [10], adaptability of gait [11], and/or dual-task performance during gait [12] to measure the gait performance. These research findings indicate that identification of gait parameters is advantageous in assessing the health condition of an elderly person.

Many techniques are available for the identification and analysis of human gait. They can be broadly classified as wearable and non-wearable technologies. Wearable technologies are not considered in the scope of this research and it is assumed that the wearable devices may hinder the natural gait of the elderly people. Non-wearable technologies using computer vision techniques for gait analysis are considered in the scope of this research. To overcome huge cost involved in gain analysis technologies, low cost and easy-to -use system the Kinect sensor was chosen to implement the data gathering system.

The purpose of this work is to develop an assisting system that will work on captured motion and recording from Kinect and provide basic tools for the analysis of individual's gait parameters. One of the important aspects of this study is extraction of the gait parameters from raw data. To achieve this, first step is to generate the RGB and depth images from the raw data collected by Kinect. For this the XPCV software [13] of TU Chemnitz is used. Once the images are extracted, we generate the rendered image and JSON files giving 25 key-joints of human skeleton using OpenPose library [14]. Then using the JSON files of OpenPose, we form a 2D skeleton. These 2D skeletons are then used for alignment with Depth image and 3D point cloud is formed. Using the 3D point cloud results, we calculate gait parameters. These estimated gait parameters are compared with the state-of-the-art approaches and the performance is evaluated.

2 Related Concepts

2.1 About Microsoft Kinect Sensor

2.1.1 The device

Microsoft Kinect [15] is a novel depth sensor released by Microsoft for Xbox 360 which can be used as a gesture capture system. This device consists of different sensors like a depth sensor and infrared emitters, RGB (three channel color) camera and a microphone array for voice recognition (figure 2.1).

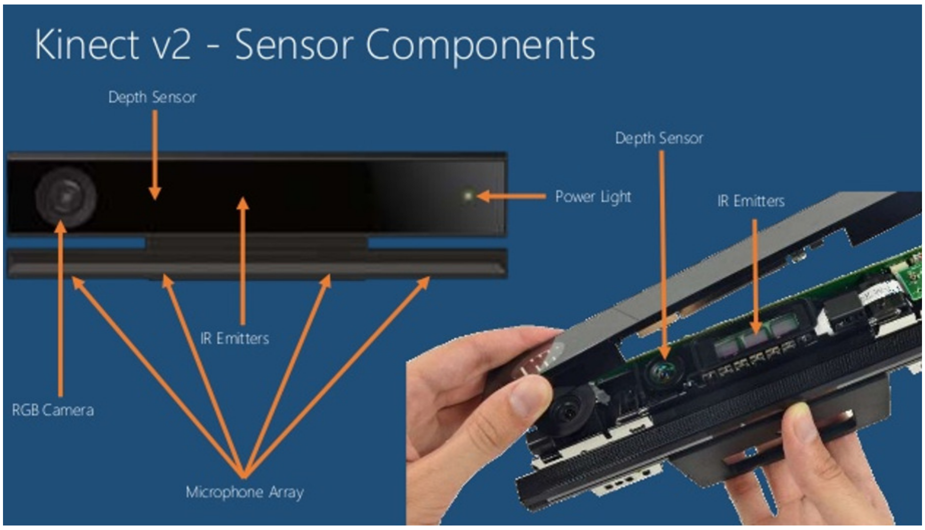


Figure 2.1: Kinect Sensor [15]

Kinect sensor was initially designed for gaming industry to give users a more realistic experience. This device acted as an interactive device between console and PC-player. The interaction was achieved not with a game controller but over gestures and voice commands in real-time. However, Kinect's impact has stretched far beyond the gaming industry by its advanced sensing technology, drawing the attention of researchers from other fields (computer science, electronic engineering, and robotics, etc) and confirming in this sense the potential that Kinect has in widespread range of application areas.

In the field of medical, Kinect sensor offers various opportunities to the handling and prevention of illness, injury or even diseases. Maximum of applications in the field of medical, especially in eHealth (healthcare done the usage of technology), use the Kinect for tracking actions of the person in order to perform rehabilitation or monitoring of patient.

2.2 Gait Cycle and Its Analysis

2.2.1 Gait Terminology: Gait Cycle and its Events

A gait cycle is defined as the time interval between repetitive sequence of walking actions (figure 2.2). It can also be termed as human locomotion. The cycle involves serial and periodic movements of body limbs and trunk, that results in a forward movement of our body. While any movement can be selected to define the cycle, by definition it is appropriate to take the cycle start as when one-foot let's say left foot hits the ground and the cycle is complete when same left foot contacts again with the ground.

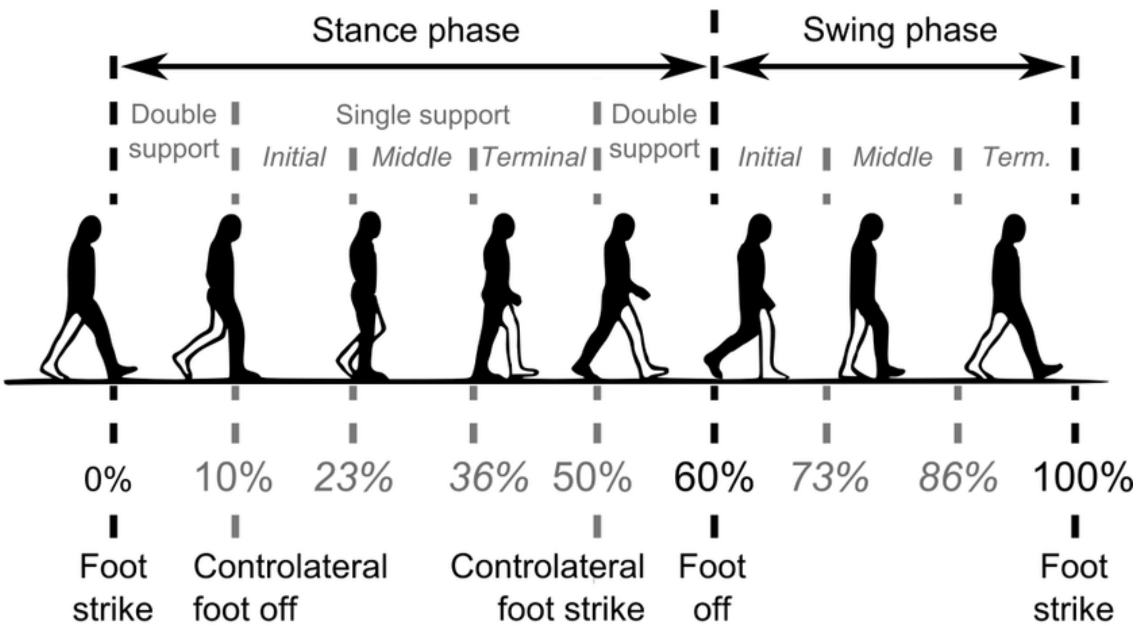


Figure 2.2: The Gait Cycle and its events [16]

Gait cycle is majorly divided into two phases: stance phase and swing phase. The stance phase is stated as the period in which the foot is touching with the ground. Stance phase covers the first 62% of the gait cycle. On the other hand, the swing phase covers the remaining 38% of whole gait cycle. Swing phase is stated as the phase in which the foot does not touch the ground. In swing phase, the foot is in the air and advancing forward. In a gait cycle, we can also distinguish the double-foot support and the single-foot support. The period in which both the foot is touching with the ground is called as Double support and it happens twice during the whole gait cycle i.e. once at the beginning and the second one at the end. The period in which only one of the feet is touching the ground supporting the whole-body weight is called as single-foot support and this also takes place twice as shown in figure 2.2.

2.2.2 Kinematic Gait Analysis

Kinematics refers to the study and description of the human motion (gait displacements, velocities and accelerations of body joints and segments) without consideration of its causes. Kinematic gait analysis usually uses different type of systems (laboratory equipment) and a wide variety of sensors and in order to capture the motion data, such as accelerometers, gyroscopes, and magneto resistive sensors. Most simple gait analysis has been carried out to obtain the spatial and temporal parameters: cycle time, stance time, swing time, step length, stride length and stride time, cadence and gait velocity, briefly defined in the following points:

- **Cycle Time or Stride Time:** Duration of one gait cycle (amount of time spent it takes to complete one stride, where STRIDE = CYCLE).
- **Step Time:** It is the time between the initial point of contact of one foot and the initial point of contact of the opposite foot.
- **Step Length:** It is the distance between the initial point of contact of one foot and the initial point of contact of the opposite foot.
- **Stride Length:** It is the distance between two consecutive heel contacts of the same foot.
- **Cadence:** Number of steps taken by unit time. Usually measured as step/min.

- **Speed:** Gait distance by unit time. Usually measured in km/h or m/s.

2.2.3 Gait Analysis and Its Main Application Areas

Clinical Context

In a clinical context, the main goal of gait analysis is to provide quantitative measures that characterize the degree of gait disorder of an individual. The data extracted in gait analysis has been reported to be of special interest in the evolution of diseases such as multiple sclerosis or Parkinson (neurological diseases), heart disease, strokes, physiological diseases caused by ageing, etc. (as all these conditions the gait is clearly affected). In this sense, gait analysis provides potential information for clinicians.
It is important to note that gait analysis is used as an evaluation tool of an already diagnosed movement disorder, not as a diagnostic tool. Despite of this fact, through complete kinematic and kinetic measurement and description of gait characteristics may aid to the early identification of movement disorders.

Sports Context

The applicability of gait analysis in the sports training field is two-fold: it can be used to increase the efficiency for performance improvement and, on the other hand, to analyse how the risk of injuries during the performance can be effectively prevented.

2.3 3D Pose estimation

2.3.1 OpenPose (Convolutional Pose Machine)

OpenPose is an open source library for real-time multiple person body-joint detection and multi-threading which is written in C++ language using OpenCV and Caffe [14]. Using OpenPose a key-point detection of joint of human body, facial key-points and hand (in total 130 key-points) can be shown in real-time. An added advantage of OpenPose is that the computational performance of system in predicting body-joint does not depend on the detection of number of persons in the image. For predicting pose estimation, Convolutional Neural Networks (CNNs) can be incorporated for learning features and dependent spatial models of image into the pose machine framework. As an input, at every stage image features and belief maps obtained from earlier stage are used. The belief maps give the successive stage an expressive non-parametric encoding of the spatial uncertainty of body-joint location. This allows the CPM to acquire rich image-dependent spatial model relationship between parts. The complete proposed multi-stage architecture can be differentiated completely and thus can be trained in an end-to-end manner using backpropagation [14].
At each stage in the CPM, the spatial context of part beliefs delivers strong disambiguating cues to a subsequent stage. As a result, each stage of a CPM produces belief maps with progressively advanced approximations for the locations of each part. In order to capture long-range interactions between parts, the design of the network in each stage of our sequential prediction framework is motivated by the goal of achieving a large receptive field on both the image and the belief maps [14].
Based on CPM architecture, OpenPose is an efficient method for multi-person pose estimation which uses a nonparametric representation of association scores via Part Affinity Fields (PAFs), a set of 2d vectors fields that encode the location and orientation of limbs over the image domain.

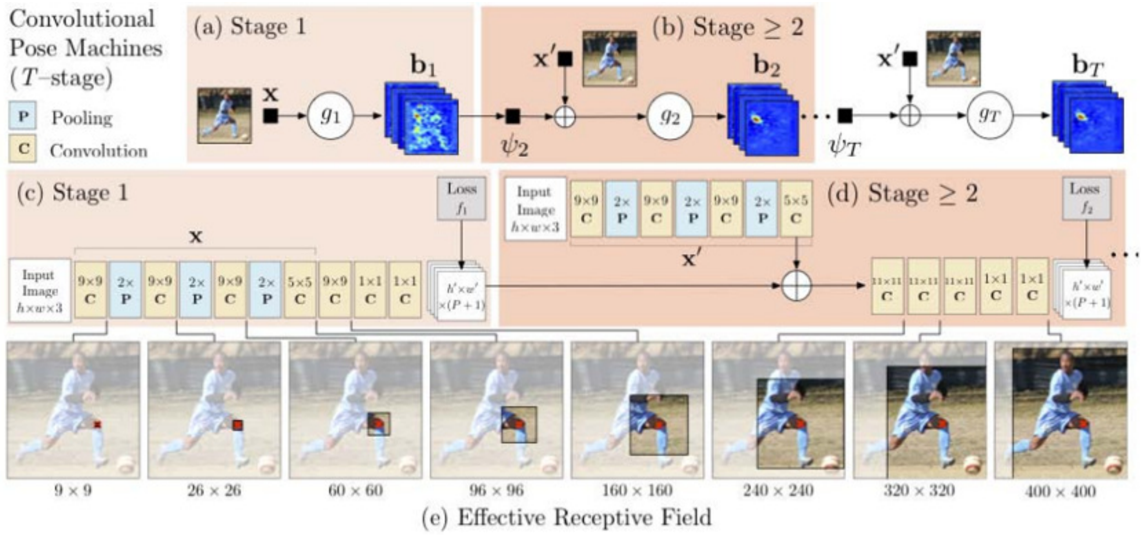


Figure 2.3: Architecture of Convolutional Pose Machines (CMPs) [14]

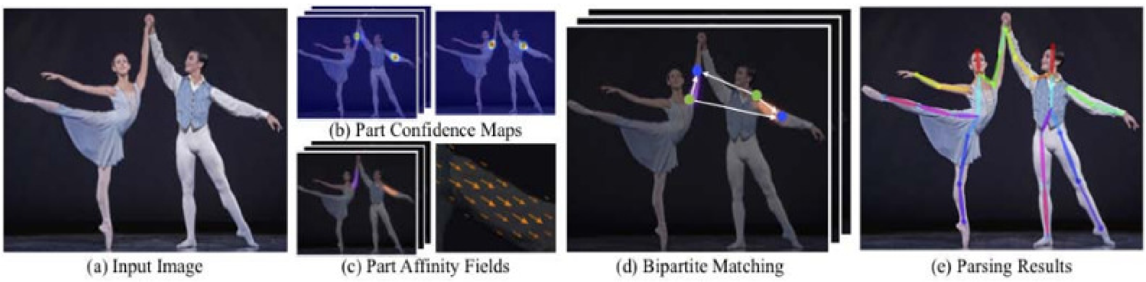


Figure 2.4: Illustrates the overall pipeline of OpenPose software [14]

The part affinity is a 2d vector field for each limb for each pixel in the area belonging to a particular limb, that encodes the direction that points from one part of the limb to the other. Each type of limb has a corresponding affinity field joining its two associated body parts. A greedy parsing algorithm is sufficient to produce high-quality parses of body poses, that maintains efficiency even as the number of people in the image increase [14].

- OpenPose takes, as input, a color image of size w x h (figure 2.4a) and produces, as output, the 2d locations of anatomical key-points for each person in the image (figure 2.4e).
- First, a feed-forward network simultaneously predicts a set of 2d confidence maps of body part locations (figure 2.4b) and a set of 2d vector fields of part

affinities, which encode the degree of association between parts (figure 2.4c).

- Finally, the confidence maps and the affinity fields are parsed by greedy inference (figure 2.4d) to output the 2d key points for all people in the image.

The architecture of OpenPose, shown in figure 2.3, simultaneously predicts detection confidence maps and affinity fields that encode part-to-part association. The network is split into two branches: the top branch, shown in beige, predicts the confidence maps, and the bottom branch, shown in blue, predicts the affinity fields. Each branch is an iterative prediction architecture, Following the typical structure of a CMP, which refines the predictions over successive stages, with intermediate supervision at each stage [14]. The OpenPose library main functionalities are:

- Multi-person 18 or 25-keypoint body pose estimation and rendering.
- Running time invariant to the number of people on the image.
- Image, video and webcam reader.
- Able to save and load the results in various formats (JSON, XML, PNG, JPG).

State of the Art

Literature review

Estimation of human pose based on computer vision technologies is an interesting research topic. Especially, the combination of computer vision and machine learning for pose estimation has wide applications ranging from biometric to surveillance and human-computer interaction. There exists huge literature on vison-based human pose estimation ranging from 2D to 3D pose estimation algorithms. Here, a brief review focuses only on 3D pose estimation algorithms and gait parameters estimation using pose estimation.

3D pose estimation

3D human pose contains more information than a 2D pose estimation model [17]. In 2D pose estimation, coordinates estimated by the model are in the identical x-y coordinate space as the input, which makes it straight forward to build a simple CNN which can map RGB inputs to a x-y heat-maps. However, for 3D pose estimation the output coordinates exist in a space with one more dimension than the input (xyz-space); an x-y space heat-map does not encode enough information to recover the depth of a pose joint. There are many 3D pose estimation systems [18]. Despite these techniques can extract 3D pose, they undermine spatial equivariance which can hinder generalization. This is exemplified by the inferior performance of such approaches in the context of 2D pose estimation.
One approach of estimating 3D pose includes the use of volumetric heat maps. [19] developed the model by gradually building up the depth resolution of the activations throughout the network in a coarse-to-fine fashion. [20] use the soft argmax operations to calculate coordinates from volumetric heat maps. Even though these techniques are advantageous, addition of another dimension to the spatial informa-

tion considerably increases the memory requirements of the system.
[21] introduce the concept of "location maps" which are spatial representations of location where each pixel contains an estimate of a particular value of the co-ordinate. For the z-coordinate, this is conceptually similar to a depth maps. Since, the location maps are 2D, they can be produced by models that are less memory intensive than volumetric heat maps. Unfortunately, evaluation of this approach is not competitive with the accuracy of other existing techniques such as volumetric heatmaps.
Many approaches such as [22, 23, 24] uses multi-view 3D pose estimation techniques. These techniques initially define a body model as simple primitives and then optimize the model parameters to align the projection of the body model with the image features. These approaches differ in terms of the used image features and optimization algorithms.
Fusing features from different sources is another common practice in the computer vision-based 3D pose estimation techniques. [25] propose to wrap the features of neighbouring frames (in a video sequence) to the current frames according to the optical flow in order to robustly detect the objects. But these techniques include a very challenging task of identifying the corresponding features across different views. Recent models such as [17], explore 3D human pose from a single RGB image and it is straight forward implementation with off-the-shelf 2D pose estimation systems and 3D mocap libraries. It outperforms, many of the other existing approaches. Other existing high accuracy 3D pose estimation techniques are also challenging due to its non-linear optimization problem [26].

Gait parameter estimation

Most existing work on gait measurement and analysis is done in a short period of time (e.g. when they attend a clinic or when they go to podiatrist). These approaches provide a biased evaluation and the gait data acquisition is not done for a long term. Following vision-based approaches provide an initial effort in development of gait analysis using pose estimation techniques.
[27] try to identify humans using spatial-temporal silhouette analysis. This technique implicitly captures the structural and transitional characteristics of the gait based on an eigenspace transformation applied to a time varying distance signals derived

from a sequence of silhouette images. [28] defines a Hidden Markov Model (HMM) to identify the dynamics of individual gait. The HMM consists for five states that are associated with five representative binary silhouettes. For each individual, a HMM is generated during the training phase and during the recognition phase, the HMM with the largest probability identifies the individual.

[29] defines gait signatures based on a Fourier analysis of the variation in the motion of the thigh and lower leg. The leg motion is extracted by a temporal template matching with a model defined by force coupled oscillators as a basis. [30] recognizes the gender of a person based on the gait. Trajectories from motion capture devices of female and male subjects are factored using a three-mode Principal component analysis (PCA) into components interpreted as posture, time and gender.

Gait analysis based on human pose estimation is very challenging due to many degrees of freedom of the human body. To resolve this issue, model-based generative approaches using 3D geometric representation of human shape and kinematic structure to reconstruct pose in an analysis-by-synthesis manner. In general, these approaches try to optimize the similarity between the projected model and the observed image [31, 32].

In many of the existing approaches, a common assumption is that an initialization of the human pose in the first frame of a sequence is given. [33] uses a matching approach to compare the edges of a projection of a 3D body model to those identified in the image. [34] uses a non-parametric belief propagation of human pose estimation. The inference method is applied to a loose-limbed graphical model, where each node represents one body part. [35] proposed a person detector which can detect and localize limbs of people in lateral walking poses. Some other approaches use some specific detections to identify body parts such as 2D shape [36], Support Vector Machine (SVM) classifier [37], "shouters" [34], multi-view eigenspaces [34], skin color [38], AdaBoost [38, 39], or locally initialized appearance models [35].

All these mentions are easily affected by shape changes like clothing, background etc. The recognition accuracy could also drop rapidly when evaluated under different environment conditions. Nonlinear dimensionality reduction (NLDR) based motion models have been successfully applied to various video.

3 Methodology

The architecture of the proposed approach is depicted in the figure 3.1. From figure 3.1 , we can see that the proposed approach consists of three main blocks. They are data acquisition, 3D pose-estimation and gait parameter extraction. In data acquisition block, the data is collected using the Microsoft Kinect device and is stored for further processing using XPCV software [13]. The stored data is then used to estimate 3D Pose estimation of the person as shown in the 3D pose-estimation block. Using the 3D pose estimation results, the gait parameters are extracted as shown in the gait parameter extraction block. The tasks performed in the scope of individual blocks is explained below in detail.

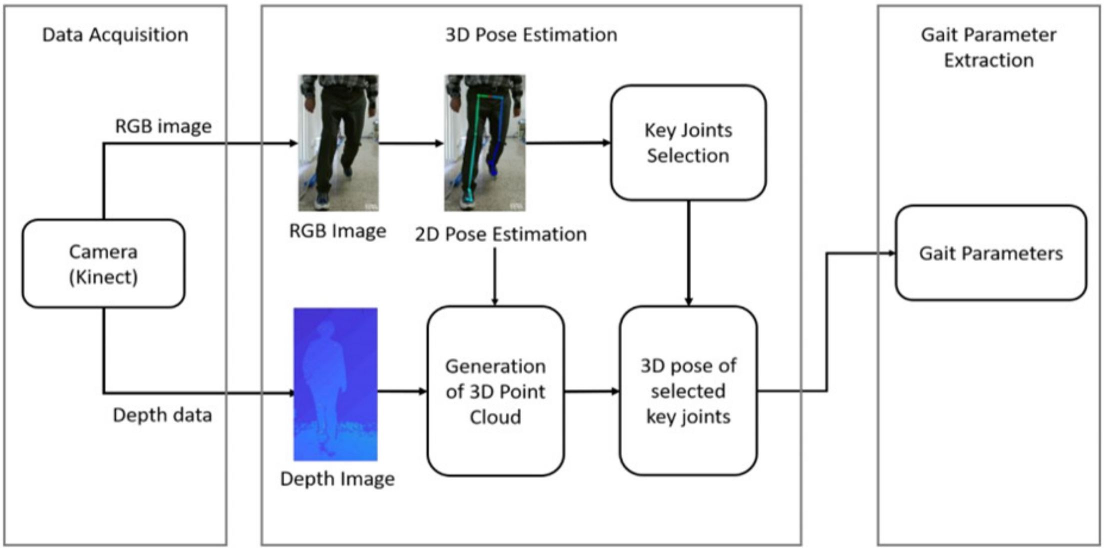


Figure 3.1: Proposed architecture for gait parameter estimation

3.1 Data acquisition

The acquisition of the elderly person's gait data is performed using Microsoft Kinect. The pictorial representation of gait data acquisition is presented in figure 3.2. From figure 3.2, we can see that a person is walking between the railings of the Optogait [39] device. This is considered as the walking path for gait data acquisition. At the end of the Optogait device, the Kinect is placed.

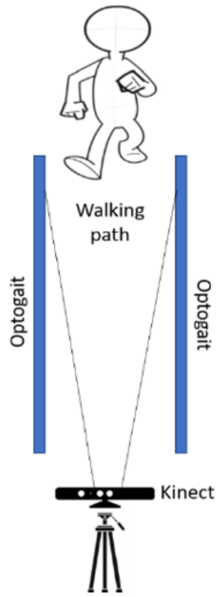


Figure 3.2: Data acquisition

The person walks near to the camera and then turns back and walk to the starting point. Kinect consists of three sensors. They are RGB color Video Graphics Array (VGA) video camera, a depth sensor and a multi-array microphone.

For the scope of this project, the data from RGB color VGA video camera and depth sensor data of each and every individual is stored. Due to some drawbacks in gait parameter estimation through Optogait, the data acquired from it is used only for benchmarking the estimated gait parameters out of 3D pose. To capture data from Kinect, XPCV software [13] of Technische Universität Chemnitz is used. XPCV software is a project supported by Chair of Digital Signal Processing and Circuit Technology (DST) at the Technische Universität Chemnitz. XPCV is a modular cross-platform framework for rapid prototyping of computer vision systems [13]. From the figure 3.3, we can see that there are two modules Kinect2Input and DataSink. The Kinect2Input module of XPCV captures the input data of Kinect Sensor i.e Skeletal Frames, Color Frames and Depth Frames and the captured data is stored in raw format using DataSink Module. As soon as we start the project, Kinect2Input starts capturing the raw data and DataSink stores this data to the desired location. The raw data consists of RGB frames, Depth frames, skeleton frames, and the timestamp file. Here the skeleton frames are the skeleton images generated by Kinect and the timestamp files consist of the time instant when the frame was captured.

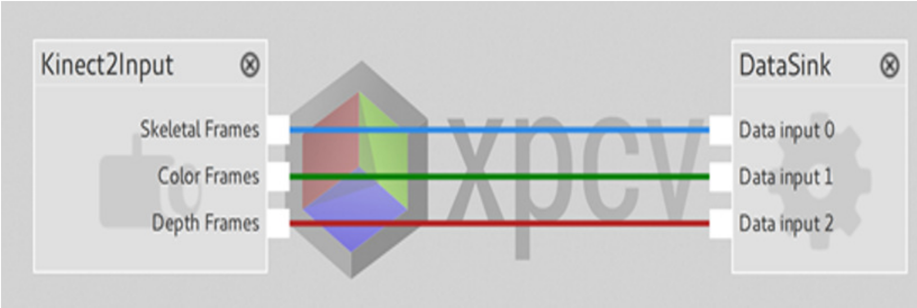


Figure 3.3: Capturing of Data using Kinect in XPCV software [13]

Once the data is stored, to extract the data from the file, DataSource module and ImageFileOutput module of XPCV Software is used as shown in the figure 3.4. As the skeleton generated by Kinect doesn't provide with good skeleton data, we only extract the Color frames and Depth frames. The data stored using the DataSink module is given as input to the DataSource module and the raw .dat files and timestamp files are extracted and stored at the desired location using ImageFileOutput.

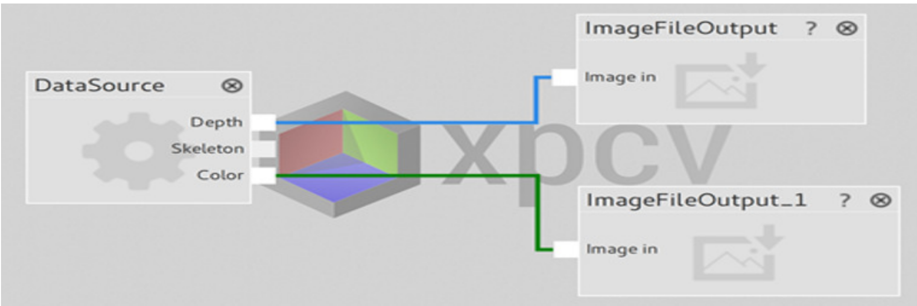


Figure 3.4: Converting raw data collected from Kinect to RGB and Depth Images

Now to use the frames for further processing, we need frames to be in some image format. As the extracted frames are in .dat format, we convert the .dat files to .bmp files using an algorithm written in MATLAB. This process is done for both RGB and Depth frames.
Now it is known that the frame rate of Kinect for Color camera and depth camera is different, so we see the difference in the number of frames of RGB and depth frames. Now to know exactly which frame of RGB corresponds to the particular depth frame, we do the comparison of timestamps files of RGB and Depth. All the

timestamp files of RGB and Depth frames are read. Then each timestamp value of RGB images is subtracted by each timestamp value of depth images. So let's say there are m numbers of RGB images and n number of depth images then after subtraction there are m x n values generated. Now for each RGB image, there is a depth image that has a minimum difference of timestamp value. That particular depth image having minimum difference of timestamp value is taken as the correct image for the corresponding RGB image. Here the timestamp values play an important role in sorting out the right Depth image for the corresponding RGB image.

The sample data acquired from RGB color VGA video camera is presented in figure 3.5. The persons in the image are blurred for privacy.

A sample of depth data from Kinect is presented in figure 3.6. These data are used for the estimation of 3D pose of the individual.

Figure 3.5: RGB gait data in laboratory

From figure 3.6, we can see that the grey color data is the acquired depth data also termed as a point cloud. The depth data which is farther than 5 meters cannot be obtained. For e.g. the person standing on the right-hand side of figure 3.5 cannot be acquired by Kinect.

Figure 3.6: Depth data of Kinect in Laboratory

3.2 3D pose estimation

From figure 3.1, we can see that the 3D pose estimation is an important aspect of the gait parameter estimation. The estimation of 3D pose involves two stages. The first stage involves 2D pose estimation and the second stage involves mapping of 2D pose estimation results with depth image to generate a 3D point cloud.

3.2.1 2D pose estimation

The RGB data acquired from the Kinect is used to estimate the 2D pose. For the estimation of 2D pose, one of the most popular 2D pose estimation method "OpenPose" [14] is used. OpenPose uses Convolutional Neural Networks to train the 2D labelled data and provide the key joints. For the given input images, all the human body in the image are detected and it gives out two files as output. One file is the rendered image in which the human skeleton is drawn wherever human body is detected and the other file named JSON file has a people array of objects, where each object has an array pose_keypoints_2d containing the body part locations and detection confidence formatted as x1,y1,c1,x2,y2,c2,... The coordinates x and y are normalized to the range [0, source size] while c is the confidence score in the range [0,1]. With OpenPose, Body_25 Model as shown in figure 3.7, we can detect 25 key joints of a human body. Those 25 key joints are Nose, Neck, Right-Shoulder, Right-Elbow, Right-Wrist, Left-Shoulder, Left-Elbow, Left-Wrist, Mid-Hip, Right-Hip, Right-Knee, Right-Ankle, Left-Hip, Left-Knee, Left-Ankle, Right-Eye, Left- Eye, Right-Ear, Left-Ear, Right-Big-Toe, Right-Small-Toe, Right-Heel, Left-Big-Toe, Right-Small-Toe, Right-Heel. Based on the quality of the image, the identification of key joints may vary. From the acquired gait data, it is possible to get all the mentioned key joints using OpenPose. Once the JSON file is generated, the 2D pose can be generated using the JSON files.

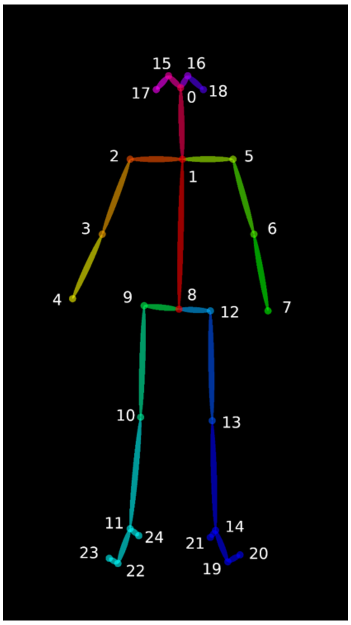


Figure 3.7: OpenPose body25 [14]

Now one of the challenges here is to find the 2D pose of the person of interest if we have many persons in the image. The JSON file only gives an array of objects of person and the corresponding array of pose_keypoints_2D but it doesn't give any information regarding which persons array object corresponds to which person. To find out the corresponding array of pose_keypoints_2D of the person of interest, an algorithm has been designed. From the figure 3.8, we can see that the person of interest is always the first person from the left of the image. For an image, the left-top corner is the point (0,0) of the form (y, x). So, to get the array of pose_keypoint_2D of that particular person, we have to find the minimum value of pose_keypoints_2D of all the values in the x-direction. From the array of pose_keypoints_2D, the one which has this minimum value that a person's pose_keypoint_2D is the required object of person.

Figure 3.8: OpenPose output with 2D skeleton of RGB Images

To achieve this, first, the JSON file is read using the loadjson MATLAB function [40] and is converted to cell array. Now, this cell array is formatted in such a way that all the x and y values of each object of person are in column format. Now from these values, the minimum value of x is found and the whole column of x and y are stored in another variable. These values stored are the pose_keypoint_2D of the required person.

Once the pose_keypoint_2D of the particular person of interest is found, the 2D skeleton is generated using these values.

To generate a 2D skeleton, first, an image of the same size as RGB is created i.e.1980 x 1080 and all the pixels in that is assigned a value zero. Then as we can see from the figure of body pose, specific key points are joined to a specific key point so we create body pairs. The created body pairs are shown below.

bodypairs = [(1,2), (1,5), (2,3), (3,4), (5, 6), (6, 7), (1, 8), (8, 9), (9, 10), (10, 11), (11, 24), (11, 22), (22, 23), (8, 12), (12, 13), (13, 14), (14, 21), (14, 19), (19, 20), (0, 15), (15, 17), (0, 16), (16, 18), (0, 1)]

Now each body pair is taken and is joined using cv2.line function. We do the same thing with all the body pair and the end image results in a 2D pose of the required person.

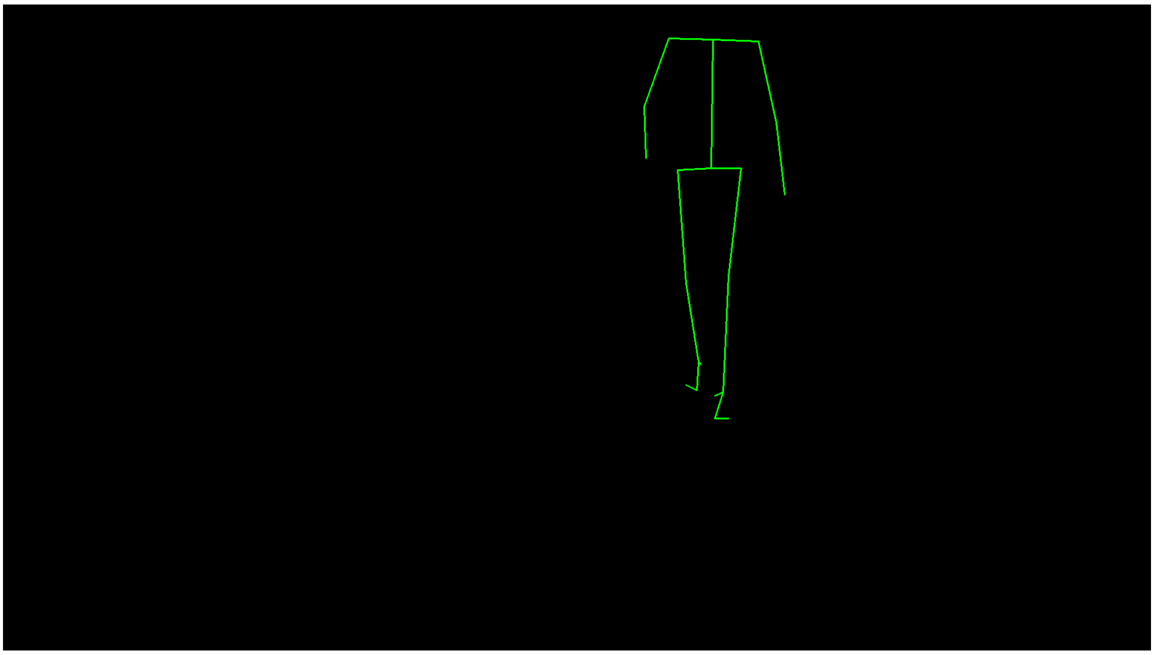


Figure 3.9: 2D skeleton from JSON values generated from OpenPose

3.2.2 3D Pose estimation

The idea of obtaining 3D pose estimation is that we will first we will take RGB images and pass it through OpenPose Library to detect human skeleton in the RGB images. Then we will select the correct skeleton using image processing techniques to form a 2D skeleton. Once we have obtained 2D skeleton, we align this 2D skeleton with the Depth image to obtain 3D pose estimation.
Now as the sizes of RGB images and depth images are different, Nicolas algorithm for Kinect calibration [41] is used for the alignment of the images.
To obtain 3D pose estimation, first the camera calibration parameters are supposed to be calculated. Camera calibration can be defined as relating the ideal camera model to the properties of an actual physical device in this case Kinect and determining the position and orientation of the camera w.r.t. the world coordinate frame. There are various methods to find these parameters.
As there is distance between the Infrared Camera and the RGB camera in the Kinect device, the depth and RGB images are not aligned.

3.2.3 Camera Calibration Methods

MATLAB

- **Step 1:** Check for the App: Camera Calibrator in MATLAB.
- **Step 2:** This opens a new window as shown in fig. Add Images and specify the size of the chessboard squares. Once you put those values, Chessboard detection starts
- **Step 3:** Now Inspect the correctness of detection and choose what to estimate – 2/3 radial distortion coefficient? – Tangential distortion? – Skew?
- **Step 4:** Click on Calibrate
- **Step 5:** Export to a MATLAB variable.

By using this we can get the values of Intrinsic parameters for color camera but it is not successful in giving intrinsic parameters for depth camera.

Drawback:

This cannot be used for Kinect Camera as it does not give intrinsic parameters for Depth camera. We also have stereo Camera Calibration App in MATLAB and even with that we can't achieve the desired result as that app requires both images to be of the same size and for Kinect, we have RGB images and depth images of different sizes.

3.2.4 Camera Calibration Toolbox for MATLAB

The Microsoft Kinect Device is widely used for computer vision applications and research and it has many interesting features like built-in calibration for mapping points of 2D image space and 3D camera space, body-tracking, face capture, 3D reconstruction etc. But all these functionalities are not available for MATLAB. The Kin2 interface for MATLAB toolbox [42] overcomes these limitations by providing many of their Kinect V2 features to MATLAB Users and one of its features is Camera Calibration.
The software requirements for using the Kin2 toolbox are: Microsoft Windows 8 or newer, Visual Studio 2012, MATLAB with image processing toolbox. All these requirements are owned by the Kinect for Windows SDK 2.0

Installation

- **Step 1:** Download the Kin2 toolbox and the add the path in MATLAB [42].
- **Step 2:** Set the compiler using mex -setup C++
- **Step 3:** Open compile_cpp_files and set the include and library paths of Kinect2 SDK.
- **Step 4:** Add to the windows path ,the bin directory containing the Kinect20.Fusion.dll and Kinect20.Face.dll For example: C:/Program Files/Microsoft SDKs/Kinect/v2.0_1409/bin
- **Step 5:** If you modify Windows path, close MATLAB and open it again in order to detect the changes.

- **Step 6:** Run compile_cpp_files.m. With this the set-up is complete
- **Step 7:** Now run calibrationDemo.m and make sure you have set the MATLAB path to Mex folder.
- **Step 8:** Now you will see two new windows, one showing the color image and other showing depth image. Press d to obtain depth camera intrinsic parameters and press c to calibrate color camera.

Camera Calibration Parameter

Camera Calibration Parameters calculated using the Camera Calibration Toolbox for MATLAB, for the Kinect are shown in the Table 4.2 and Table 3.2:

Color Camera Parameters

Parameters	Values
Focal Length X	1058.6819
Focal Length Y	1058.6819
Principal Point X	961.7849
Principal Point Y	535.8286
Rotation wrt Depth camera	0.99999 0.0037317 0.0020157 -0.0037282 0.99999 -0.0017105 -0.0020221 0.001703 1
Translation x, y, z wrt depth camera(meters)	0.052675 3.7884e-05 2.9777e-05
Radial Distortion 2nd order	0.024998
Radial Distortion 4th order	-0.015233
Radial Distortion 6th order	0.0069285

Table 3.1: Color Camera Parameters

Depth Camera Parameters

Parameters	Values
Focal Length X	364.3901
Focal Length Y	364.3901
Principal Point X	256.162
Principal Point Y	209.5862
Radial Distortion 2nd order	209.5862
Radial Distortion 4th order	-0.2663
Radial Distortion 6th order	0.082286

Table 3.2: Depth Camera Parameters

3.2.5 Align depth image and RGB image

For 3D pose estimation, the mapping of the RGB image and depth image is required. As the image size of RGB and depth are not the same, the alignment of both images is required.

By referring to Nicolas Burrus algorithm from University of Madrid on Kinect Calibration [41], the alignment of RGB and depth images are achieved. Nicolas Burrus maintains a software package Kinect RGB Demo. His procedure consists of photogrammetric calibration, correction for image distortions, stereo calibration between IR and RGB-camera and reconstruction of 3D scene as a colored point cloud. The program also includes some methods to reconstruct complex 3D environments.

Here are the steps to transform from the raw disparities of the depth image to the rectified RGB image.

- **Step 1:** Mapping depth pixels from depth image coordinates [xd, yd]T to depth camera coordinates [Xd,Yd,Zd]T. Using the intrinsic parameters of the depth camera, each pixel (x_d,y_d) of the depth camera can be projected to metric 3D space using the following formula:

$$Xd = \frac{(x\_d - cx\_d) * depth(x\_d, y\_d)}{fx\_d} \tag{3.1}$$

$$Yd = \frac{(y\_d - cy\_d) * depth(x\_d, y\_d)}{fy\_d} \tag{3.2}$$

$$Zd = depth(x\_d, y\_d) \tag{3.3}$$

  where fx_d, fy_d, cx_d and cy_d are the intrinsic parameters of the depth camera.

- **Step 2:** Transform point clouds from depth camera coordinates Xd to color camera coordinates XRGB.

$$XRGB = R * Xd + T \tag{3.4}$$

- **Step 3:** Mapping point clouds from color camera coordinates XRGB to color image coordinates [xRGB, yRGB]T.

$$xRGB = (XRGB * fxRGB/ZRGB) + cxRGB \tag{3.5}$$

$$yRGB = (YRGB * fyRGB/ZRGB) + cyRGB \tag{3.6}$$

Note: After projecting to color image coordinates, xRGB and yRGB must be rounded to (1, 512) and (1, 424) respectively. Once the alignment of RGB and depth image are performed, we get the 3D Pose estimation. This 3D Pose estimation results can be saved in the form of point cloud. Also, for the 25 key-joints of interest, the 3D pose-estimation values x, y, z is saved in an excel file which is used for estimation of gait parameters.

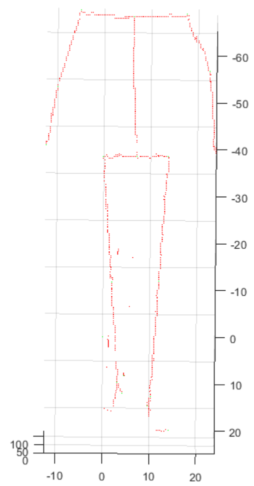


Figure 3.10: Estimation of 3D pose from RGB and Depth images

3.3 Gait parameters estimation

From the 3D pose of the person, the gait parameters are estimated. These gait parameters are important for the analysis of gait. Gait parameters such as mid-hip displacement, displacement of right and left foot, step length, stride length, swing time, stance time, stride time, cadence and velocity are estimated. The important definitions of these gait parameters are given below.

3.3.1 Step Length

It is the distance between the initial point of contact of one foot and the initial point of contact of the opposite foot. Let z1 be the initial point of contact of one foot (right foot) and z2 be the initial point of contact of the opposite foot (left foot) as shown in figure 3.11then step length is calculated as shown below

$$StepLength = |z2 - z1| \tag{3.7}$$

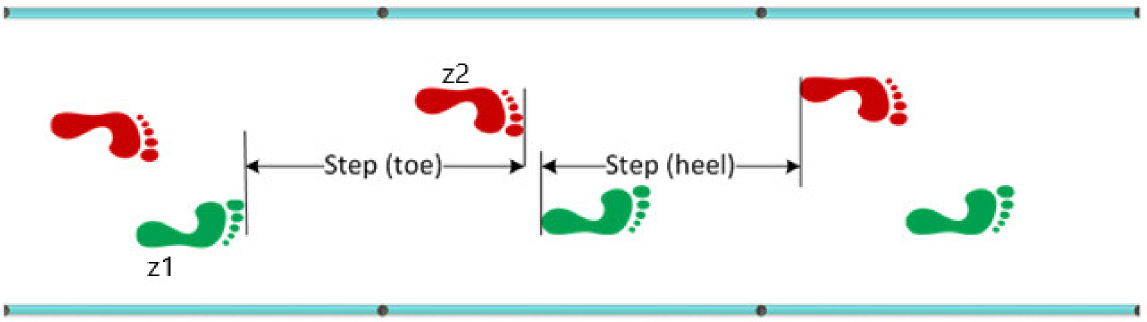


Figure 3.11: Step length [39]

3.3.2 Stride Length

It is the distance between two consecutive heel contacts of the same foot. Let z1 be the initial point of contact of one foot (left foot), z2 be the initial point of contact of the opposite foot (right foot) and z3 be the initial point of contact of the left foot as shown in figure 3.12 then step length is calculated as shown below

$$StrideLength = |z3 - z1| \tag{3.8}$$

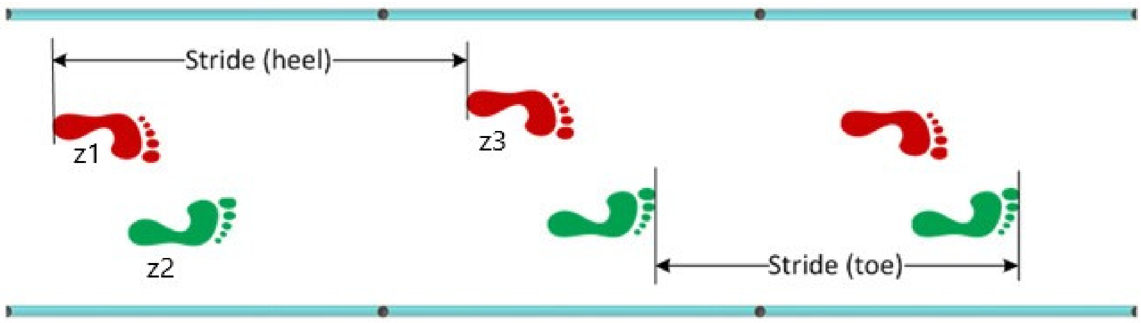


Figure 3.12: Stride length [39]

3.3.3 Estimation of Step Length and Stride Length

As the sequence of body-joint locations of all the frames is known, the feet locations across time is used to perform gait analysis as shown in figure 3.13 . To estimate step length and stride length, we follow the following steps [43]:

- **Step 1:** The first step is to calculate the horizontal displacement from the origin of the left and right foot. This is done using the z-values from 3D Pose estimation. To calculate the horizontal displacement, we calculate the horizontal Euclidean distance between the two feet.
- **Step 2:** Then we calculate the minima and maxima for these horizontal displacements.
- **Step 3:** To remove false extrema, apply convolution smoothing with a uniform kernel to get a smooth curve.
- **Step 4:** Finally, we determine the correct extrema (maxima and minima).
- **Step 5:** To calculate stride length, we take into consideration alternate maxima.

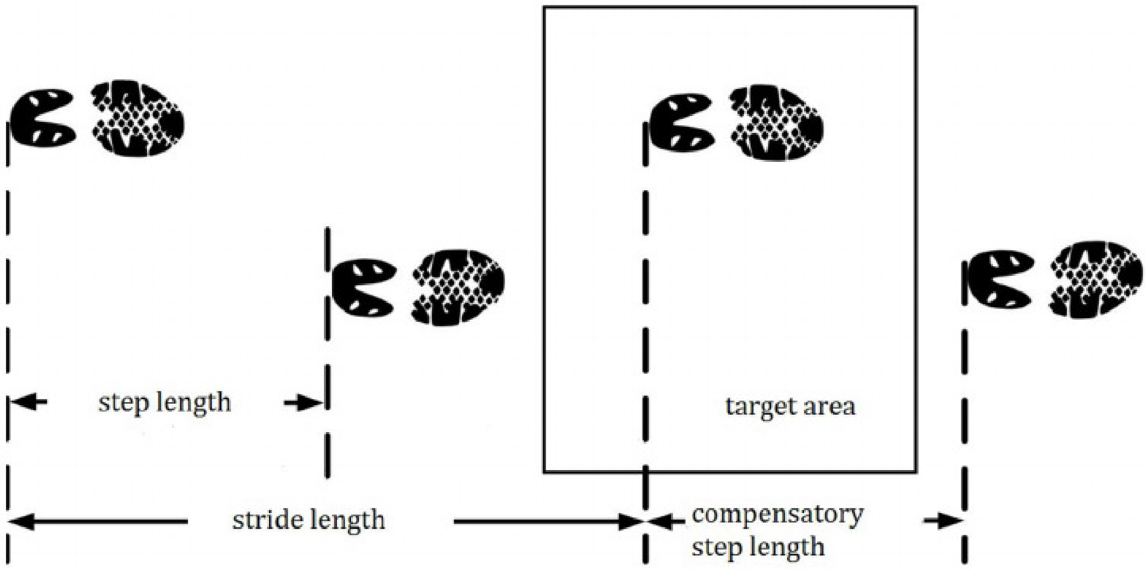


Figure 3.13: Step length and stride length [39]

3.3.4 Step Time

It is the time taken to complete one step i.e. time taken between the initial point of contact of one foot and the initial point of contact of the opposite foot. Let f1 be the frame number of the initial point of contact of one foot, f2 be the frame number of the initial point of contact of the opposite foot as shown in figure 3.14 and as Kinect is set to a frame rate 25, we take fr as 25 then step time can be calculated as follows

$$StepTime = \frac{f2 - f1}{framerate(fr)} \tag{3.9}$$

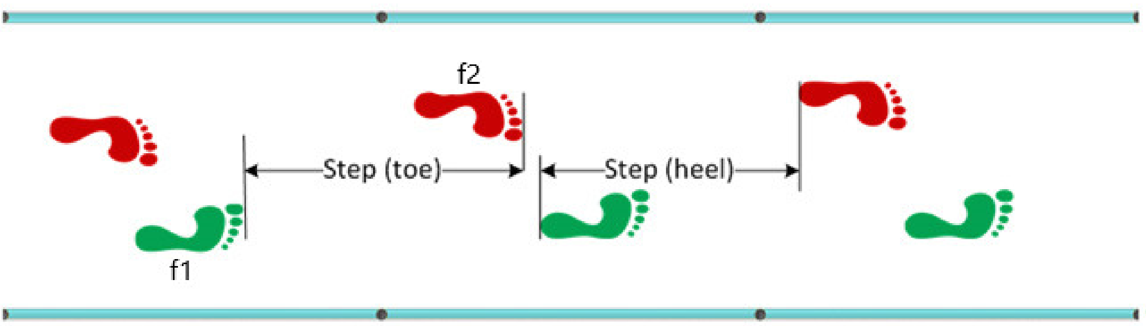


Figure 3.14: StepTime [39]

3.3.5 Stride Time

It is the time taken to complete one stride i.e. the time taken for two consecutive heel contacts of the same foot. Let f1 be the frame number of the initial point of contact of one foot (left foot), f2 be the frame number of the initial point of contact of the opposite foot(right foot) , f3 be the frame number of next initial point of contact of left foot as shown in figure 3.15 and as Kinect is set to a frame rate 25, we take fr as 25 then stride time can be calculated as follows

$$StrideTime = \frac{f3 - f1}{framerate(fr)} \tag{3.10}$$

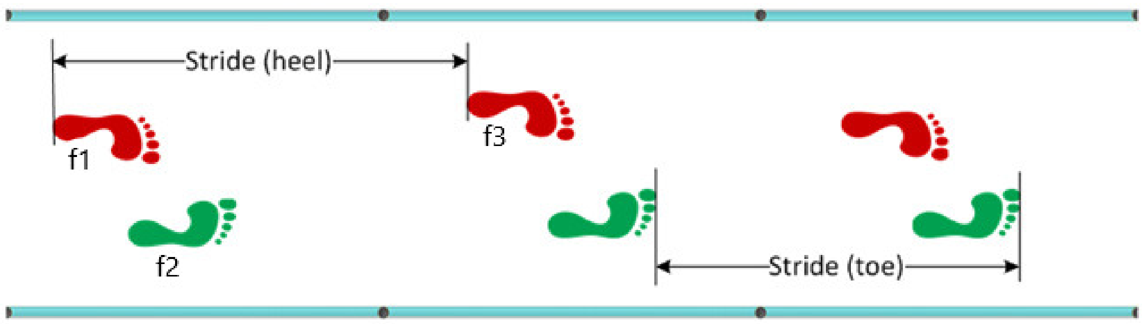


Figure 3.15: Stride time [39]

3.3.6 Cycle Time or Stride Time

It is the time interval between successive instant of foot floor contact of the same foot i.e. duration of one gait cycle.

$$CycleTime = StrideTime \tag{3.11}$$

3.3.7 Cadence

Number of steps taken by unit time. Usually measured as step/min. Let S be the number of steps completed in time T minutes then cadence is given as follows

$$Cadence = \frac{Numberofsteps(S)}{time(Tminutes)} \tag{3.12}$$

3.3.8 Speed

Speed is defined as the gait distance travelled by a person per unit time as shown in figure 3.16. It is usually measured in km/h or m/s.

$$Speed = \frac{|z2 - z1|}{Tc + Tv} \tag{3.13}$$

$$Speed = StepLength * Cadence \tag{3.14}$$

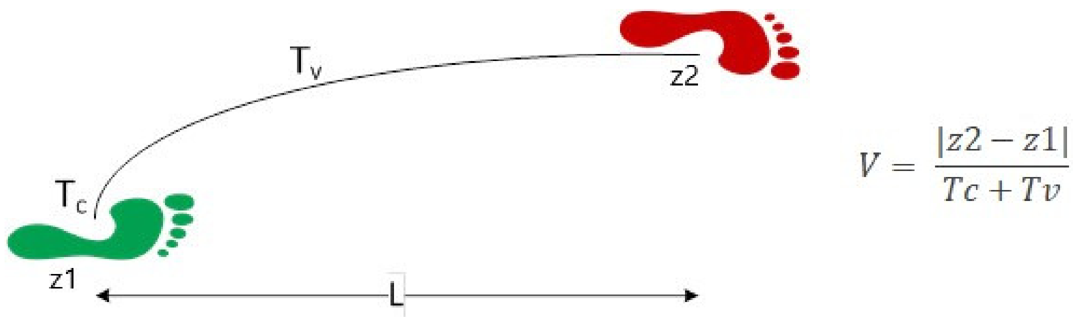


Figure 3.16: Speed [39]

4 Results, evaluation and comparison

As estimation of gait parameters is an important aspect in the analysis of gait, this section gives the obtained results and overview about the estimated gait parameters from different methods.

The table 4.1 given below shows the results of estimated gait parameters using three different methods. The first method is using Optogait device and this result are considered as the ground truth for comparison with other methods. The second method is the proposed approach using 3D pose estimation with OpenPose, RGB images and depth images. And the third method is 3D human pose estimation using Deep Linear Network (DLN) with residual connections.

The estimation of gait parameters are listed in Table 4.1.

Parameters / Method	Step Length (cms)	Stride Length (cms)	Step Time (secs)	Stride Time (secs)	Cadence (steps/min)	Velocity (m/sec)
Optogait Device	38.2	58.53	0.69	1.39	133.51	0.85
Proposed Approach	39.50	61.00	0.72	1.44	138.46	0.91
3D Pose Estimation using DLN	29.43	40.86	0.36	0.72	214.29	0.82

Table 4.1: Estimated Gait Parameters

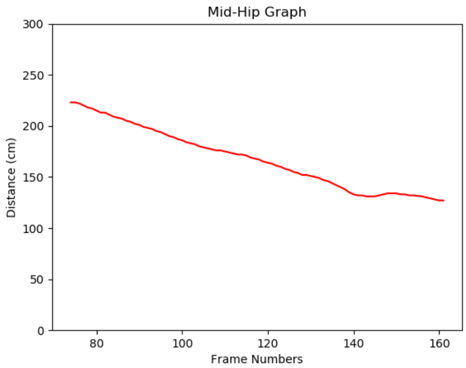
From Table 4.1, we can see that the estimated gait parameters using proposed approach matches with the computed gait parameters using Optogait device. Also, it can be observed that some of the gait parameters obtained from 3D human pose estimation using Deep Linear Network (DLN) approach matches with the computed gait parameters using Optogait device but are still not good enough compared to the proposed approach. This makes the proposed approach more effective in performing gait analysis when compared to 3D human pose estimation using Deep Linear Network (DLN) approach. And with proposed approach, rather than measuring the gait parameters related to feet, we can also analyse other parameters like knee position, hip displacement etc. These additional parameters are further helpful in analysis of gait efficiently.

Table 4.2 below shows the accuracy percentage of different methods. Here the parameters obtained from Optogait Device are considered to be ground truth and so the parameters are assigned with 100 percent accuracy. From the table we can see that the accuracy obtained from proposed approach is far better than 3D human pose estimation using Deep Linear Network (DLN) approach. To see these results in more detail, comparison of proposed approach and 3D human pose estimation using Deep Linear Network (DLN) approach is done graphically.

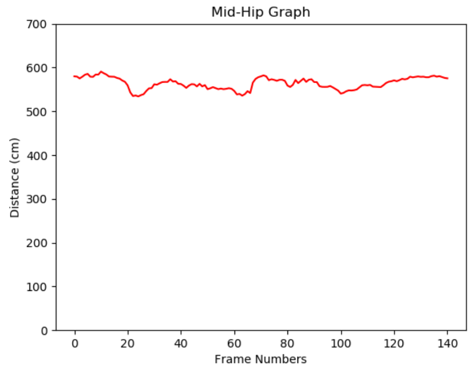
Method \ Parameters	Experiment	Step Length (%)	Stride Length (%)	Step Time (%)	Stride Time (%)	Cadence (%)	Velocity (%)
Optogait Device	RG1	100	100	100	100	100	100
	RG2						
	CB7						
	WSPL						
Proposed Approach	RG1	80.78	80.27	81.60	82.60	74.21	77.76
	RG2	82.84	80.42	78.14	81.28	80.35	82.43
	CB7	78.21	77.65	79.13	81.29	79.67	78.81
	WSPL	82.90	81.95	82.49	79.72	75.57	76.40
3D Pose Estimation using DLN	RG1	69.81	61.84	51.69	55.68	47.28	42.60
	RG2	61.08	58.97	27.28	33.85	39.82	22.67
	CB7	56.95	59.24	33.96	44.65	52.92	49.90
	WSPL	70.62	49.82	38.524	55.21	43.81	44.75

Table 4.2: Accuracy in Percentage of Estimated Gait Parameters

Figure 4.1 below gives the displacement of mid-hip of the person during the gait analysis. On X-axis is the number of frames and on Y-axis is the distance in centimetres. As the person comes towards the camera, the distance of mid-hip joint wrt camera should decrease and it can be observed with the proposed approach. But in the case of 3D human pose estimation using Deep Linear Network (DLN) approach, the distance of the mid-hip joint almost remains the same.



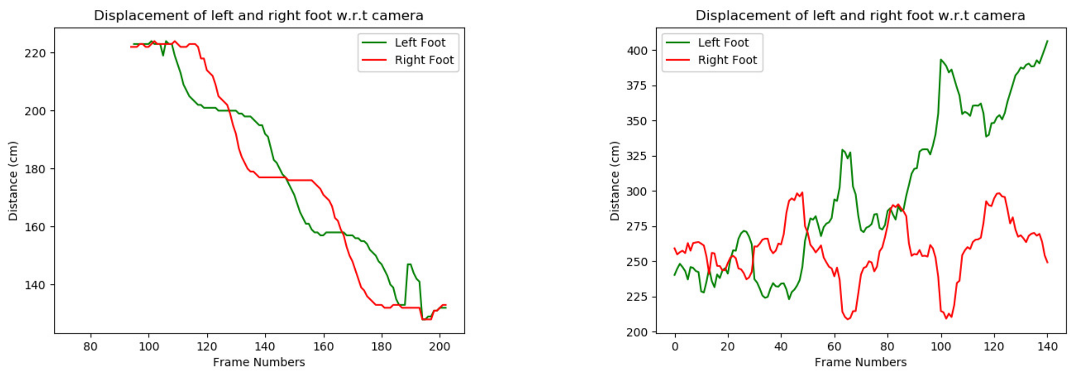
(a) Proposed Approach



(b) 3D Pose Estimation using DLN

Figure 4.1: Mid-Hip estimation of a person walking

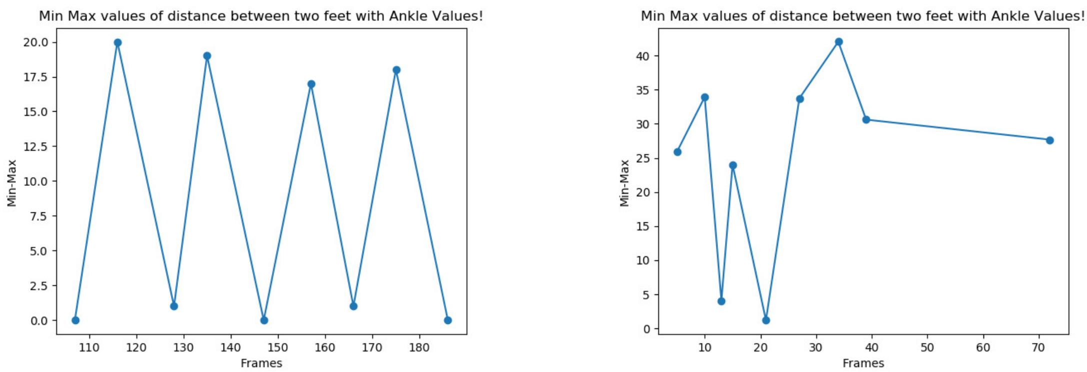
The distance between the person wrt camera reduces as the person walks towards the camera. Hence the distance of the right and left foot decreases continuously as it can be seen from figure 4.2 with the help of proposed approach when compared to the results of 3D human pose estimation using Deep Linear Network (DLN) approach, where the distance of the right and left foot increases asymmetrically. The disturbance in the distance indicate the stance and swing phase of the gait cycle.



(a) Proposed Approach (b) 3D Pose Estimation using DLN

Figure 4.2: Displacement of Left and Right foot of a person walking

Figure 4.3 shows the minima and maxima values of horizontal displacement of left and right foot. The minima and maxima values are very important for the calculation of gait parameters. These values are used to calculate the step length and stride length and with the help of frame numbers we calculate the step time and stride time. The combination of min- max values and frame numbers are used to calculate cadence, velocity. The proposed approach proves to be efficient in obtaining min-max values comparing with the 3D human pose estimation using Deep Linear Network (DLN) approach where the min-max values are observed to the scattered. So, we can conclude that the obtained gait parameter values with 3D human pose estimation using Deep Linear Network (DLN) approach are not accurate.



(a) Proposed Approach (b) 3D Pose Estimation using DLN

Figure 4.3: Minima and maxima values for calculation of gait parameters

5 Conclusion

With increase in age, it is observed that due to several neuro-degenerative or musculoskeletal disorders, there will be an effect on the human gait. Analysis of gait is an important aspect and is important in many clinical applications which include the possibility of detecting dementia in advance. For such scenarios, a long-term gait data of an individual elderly person is necessary. This target is achieved by acquiring the gait data of elderly persons with age above 80 which is done in a span of three years. The gait data is acquired using the Kinect.
3D pose estimation is done using the state of the art 2D pose-estimation algorithm Openpose and mapped the 2D poses to the depth data for the estimation of 3D poses. The necessary key joint for gait parameter estimation are selected and the projection of those key joints into a 3D environment is done. Later, some gait parameters that are useful to analyse the gait of a person are estimated. The estimated gait parameters are compared with the gait parameters computed using commercial gait analysis device Optogait and 3D human pose estimation using Deep Linear Network (DLN) approach and the results look promising.
In future, main focus is on estimating more gait parameters from 3D pose and also to improve the accuray of estimated gait parameters. Performing the estimation of gait parameters in multiple elderly individuals for more times and identification of possible gait abnormalities. Identification of certain gait abnormal patterns for definition of possible health conditions with the associated gait deviations. This approach opens even more options in the field of gait analysis and in future these possibilities are explored more efficiently.

Bibliography

[1] W. M. Van Der Flier and P. Scheltens, “Epidemiology and risk factors of dementia,” Neurol. Pract., vol. 76, no. SUPPL. 5, 2005.

[2] O. Beauchet, G. Allali, G. Berrut, C. Hommet, V. Dubost, and F. Assal, “Gait analysis in demented subjects: Interests and perspectives,” Neuropsychiatr. Dis. Treat., vol. 4, no. 1 A, pp. 155–160, 2008.

[3] World Health Organization, “Dementia,” Fact Sheets, 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/dementia. [Accessed: 29-Sep- 2019].

[4] Oliver Beauchet, Cedric Annweiler, Michele L Callisaya, Anne-Marie De Cock, and Jorunn L Helbostad, “Poor Gait Performance and Prediction of Dementia: Results From a Meta-Analysis,” Physiol. Behav., vol. 176, no. 3, pp. 139–148, 2017.

[5] R. D. Seidler et al., “Motor control and aging: Links to age-related brain structural, functional, and biochemical effects,” Neurosci. Biobehav. Rev., vol. 34, no. 5, pp. 721–733, 2010.

[6] M. D. Joe Verghese, M.D., Richard B. Lipton, M.D., Charles B. Hall, Ph.D., Gail Kuslansky, Ph.D., Mindy J. Katz, M.P.H., and Herman Buschke, “Abnormality of Gait As a Predictor of Non-Alzheimer’S Dementia,” vol. 347, no. 22, pp. 1761–1768, 2002.

[7] J. Perry and Judith M. Burnfield, Gait Analysis: Normal and Pathological Function, 2nd ed. SLACK Incorporated, 2010.

[8] A. Middleton, S. L. Fritz, and M. Lusardi, “Walking speed: The functional vital sign,” J. Aging Phys. Act., vol. 23, no. 2, pp. 314–322, 2015.

[9] F. Wang, K. Kim, S. Wen, Y. Wang, and C. Wu, "Study of gait symmetry quantification and its application to intelligent prosthetic leg development," 2011 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2011, pp. 1361–1366, 2011.

[10] V. Southard and R. Gallagher, "The 6MWT: Will different methods of instruction and measurement affect performance of healthy aging and older adults?," J. Geriatr. Phys. Ther., vol. 36, no. 2, pp. 68–73, 2013.

[11] V. Weerdesteyn, K. L. Hollands, and M. A. Hollands, Gait adaptability, 1st ed., vol. 159. Elsevier B.V., 2018.

[12] A. Leland, K. Tavakol, J. Scholten, D. Mathis, D. Maron, and S. Bakhshi, "The Role of Dual Tasking in the Assessment of Gait, Cognition and Community Reintegration of Veterans with Mild Traumatic Brain Injury," Mater. Socio Medica, vol. 29, no. 4, p. 251, 2017.

[13] Cross-Platform Computer Vision Framework – Develop. Fast. Flexible. (n.d.). Retrieved September 3, 2019, from https://xpcv.dev/

[14] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2D pose estimation using part affinity fields," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, no. Xxx, pp. 1302–1310, 2017.

[15] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE Multimed., vol. 19, no. 2, pp. 4–10, 2012.

[16] C. H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 5759–5767, 2017.

[17] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D Human pose estimation: A review of the literature and analysis of covariates," Comput. Vis. Image Underst., vol. 152, no. October 2017, pp. 1–20, 2016.

[18] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 1263–1272, 2017.

[19] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5137–5146, 2018.

[20] D. Mehta et al., "VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera," Tog, vol. 36, no. 14, pp. 1–13, 2017.

[21] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting multiple views for marker-less 3D human pose annotations," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 1253–1262, 2017.

[22] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3D human pose estimation," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11214 LNCS, pp. 765–782, 2018.

[23] H. Rhodin et al., "Learning Monocular 3D Human Pose Estimation from Multi-view Images," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8437–8446, 2018.

[24] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-October, pp. 398–407, 2017.

[25] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to Estimate 3D Human Pose and Shape from a Single Color Image," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 459–468, 2018.

[26] L. Wang, T. Tan, H. Ning, and W. Hu, “Silhouette Analysis-Based Gait Recognition for Human Identification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pp. 1505–1518, 2003.

[27] W. Gowtham Bhargavas, K. Harshavardhan, G. C. Mohan, A. Nikhil Sharma, and C. Prathap, “Human identification using gait recognition,” Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017, vol. 2018-January, pp. 1510–1513, 2018.

[28] C. Y. Yam, M. S. Nixon, and J. N. Carter, “On the relationship of human walking and running: Automatic person identification by gait,” Proc. - Int. Conf. Pattern Recognit., vol. 16, no. 1, pp. 287–290, 2002.

[29] J. W. Davis and H. Gao, “Gender recognition from walking movements using adaptive three-mode PCA,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2004-January, no. January, 2004.

[30] D. Hogg, “Model-based vision: a program to see a walking person,” Image Vis. Comput., vol. 1, no. 1, pp. 5–20, 1983.

[31] S. Wachtert and H. Nageltt, “Tracking of Persons in Monocular Image Sequences,” IEEE Nonrigid Articul. Motion Work., pp. 2–9, 1997.

[32] K. Rohr, “Incremental recognition of pedestrians from image sequences,” IEEE Comput. Vis. Pattern Recognit., pp. 8–13, 1993.

[33] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, “Tracking loose-limbed people,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, pp. 0– 7, 2004.

[34] D. Ramanan, D. A. Forsyth, and A. Zisserman, “Strike a pose: Tracking people by finding stylized poses,” Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005, vol. I, pp. 271–278, 2005.

[35] Roberts Timothy, Mckenna Stephen, and Ricketts Ian W, Human pose estimation using learnt probabilistic region similarities and partial configurations, European C. Springer, 2004.

[36] R. Ronfard, C. Schmid, and B. Triggs, “Learning to parse pictures of people,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2353, no. 602, pp. 700–714, 2002.

[37] M. W. Lee and R. Nevatia, “Human pose tracking in monocular sequence using multilevel structured models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 1, pp. 27–38, 2009.

[38] A. S. Micilotta, E. J. Ong, and R. Bowden, “Detection and tracking of humans by probabilistic body part assembly,” BMVC 2005 - Proc. Br. Mach. Vis. Conf. 2005, 2005.

[39] M. Srl, “OPTOGAIT-System for Gait Analysis.” [Online]. Available: http://www.optogait.com/. [Accessed: 29-Sep-2019].

[40] Qianqian Fang (2020). JSONLab: a toolbox to encode/decode JSON files (https://www.mathworks.com/matlabcentral/fileexchange/33381-jsonlab-a-toolbox-to-encode-decode-json-files), MATLAB Central File Exchange.

[41] Burrus, Nicolas. “Kinect Calibration.” Nicolas Burrus Homepage - Kinect Calibration. http://nicolas.burrus.name/index.php/Research/KinectCalibration.

[42] Terven Juan, Cordova-Esparza Diana, Kin2. A Kinect 2 Toolbox for MATLAB, Science of Computer Programming, 2016, http://dx.doi.org/10.1016/j.scico.2016.05.009

[43] David, Sayana, Anin, Darke, Evan, Shen, Kelly, et al. “Vision-Based Gait Analysis for Senior Care.” arXiv.org, December 1, 2018. https://arxiv.org/abs/1812.00169.