Virtual Tailor

Human Body Parameter Determination using a Kinect Sensor

Presented by:
Nevarr Pillay

Prepared for:
F. Nicolls
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics.

November 12, 2017

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:...........................
Nevarr Pillay

Date:...............................

Acknowledgments

Abstract

- Open the **Project Report Template.tex** file and carefully follow the comments (starting with %).
- Process the file with **pdflatex**, using other processors may need you to change some features such as graphics types.
- Note the files included in the **Project Report Template.tex** (with the .tex extension excluded). You can open these files separately and modify their contents or create new ones.
- Contact the latex namual for more features in your document such as equations, subfigures, footnotes, subscripts & superscripts, special characters etc.
- I recommend using the **kile** latex IDE, as it is simple to use.

Contents

Chapter 1

Introduction

1.1 Background to the study

A very brief background to your area of research. Start off with a general introduction to the area and then narrow it down to your focus area. Used to set the scene [1].

1.2 Objectives of this study

1.2.1 Problems to be investigated

Description of the main questions to be investigated in this study.

1.2.2 Purpose of the study

Give the significance of investigating these problems. It must be obvious why you are doing this study and why it is relevant.

1.3 Scope and Limitations

Scope indicates to the reader what has and has not been included in the study. Limitations tell the reader what factors influenced the study such as sample size, time etc. It is not a section for excuses as to why your project may or may not have worked.

1.4 Plan of development

Here you tell the reader how your report has been organised and what is included in each chapter.

I recommend that you write this section last. You can then tailor it to your report.

Chapter 2

Literature Review

Technology background

1. Depth Sensor technology + How Kinect works
2. Point cloud map

Coding references/Getting started

3. Code references and and blog posts?
4. Previous example
5. Hand Example

Mathematics Used

6. Papers on ellipse circumference

Improving Accuracy

7. Skeleton Joints filtering
8. Error Model

Further developments

9. Augmented reality paper

Imaging Processing Background

10. Basics of an image - RGB
11. Matlab

12. Camera Model

Uncertainty Measurements

13. Gaussian
14. Triangular

!!!!!!!!!!!!!!!!!!!!Explain Lit Review Layout

2.1 Literature Guide

2.1.1 Getting Started

Non-Contact Human Body Parameter Measurement Based on Kinect Sensor [3]

Overview This journal article provides an adequate base to understand many techniques used during this project. In the article, a similar project is explored in terms of using a Kinect to determine various personalized measurements. This includes measurements such as height, shoulder length, other key limb lengths and front perimeter measurements for the chest, stomach and waist. All of the work done in this article is similar to the work done in the initial stages of this project. The article also provides useful contextual information regarding the current use of this technology and background knowledge regarding the inner workings of the Kinect.

Relevance The main contribution of this article was the background information it provided, together with the validation of certain aspects of the experiment methodology:

Background Theory This article provides a summary and overview of the process the Kinect utilises to retrieve data about a detected human. It explains the different sensors the Kinect posses, together with a basic understanding of the internal process required to track a skeleton and return information such as "Joint" position etc. This is explained in more detailed in Section (Reference to Middleware). It also provided an explanation of how pixels in the image plane are converted to real world positional points using information from both depth and colour frames. This is also explained in detail in Section (Depth - real conversion).

Methodology This article was found after an initial strategy for calculating measurements was created. However, it served as a validation of the methods used, specifically with regards to using the Pythagorean Theorem in 3D space to calculate distances and for calculating the error between actual measurements and measurements obtained using the Kinect. These methods and the relevant equations are explained in Section (Measurement - Pyth + error). A method was also suggested for the calibration of results to improve their accuracy. However, this technique was not employed in this project. Reasons for the exclusion are given in Section (Design - No calibration or correction)

Real-Time Hands Detection in Depth Image by Using Distance with Kinect Camera [4]

Overview This journal article explored various techniques to improve hand detection using a Kinect. The main focuses were in the areas of background removal, noise removal and feature extraction. Traditional image processing techniques require a large amount of processing power and often produced ambiguous results, with regards to feature extraction, and were prone to noise. Introduction of the Kinect allowed for the use of depth data to reduce the computational power needed and to improve the accuracy of the results. A large section of the paper was dedicated to "Shadow Removal", which is the process of removing depth data unavailable due to an object obstructing the emitted Infrared Rays. "Shadow" was determined to be a significant source of noise in the depth data and its removal improved the accuracy of the system.

Relevance This article was very specific to the detection of hands and therefore its use was limited. However, valuable insights were gained regarding the "Shadow Removal" and Background Removal processes implemented (Classifier Maybe):

Shadow Removal The intricacies of Kinect in creating a depth image using its IR Emitter and IR Camera were explored in more detail. This further expanded on the inner workings of the Kinect and contributed to Section (Depth Image Creation - Kinect). Additionally, this brought attention to a significant source of noise in the depth image and provided a method to improve the accuracy of measurement results. This method is explored in Section (Rec - Noise Removal)

Background Removal This article detailed a method of Background Removal that did not utilise the BackgroundRemoval API available in the Kinect for Windows SDK. This method provides an alternative method for background that could be investigated if the BackgroundRemoval API does not provide the desired results. This is explored further in Section (Rec - Background Removal)

A Real Time Virtual Dressing Room Application using Kinect [5]

Overview This report details and explains an approach used to create a virtual dressing room. The end result was an application that allowed a user to see what clothing would look like by augmenting it onto their body. This was executed in a three part process of user extraction, user tracking and clothing mapping. A Kinect sensor was used to extract the user from the background using depth images and user information. The Kinect was also used to track the skeleton of the user and create a coordinate system on which the virtual clothing could be mapped. Lastly, information including joint orientation and key body lengths were used to augment the clothing model onto the body.

Relevance This report details and explains a possible extension on this project that would be very useful in a production level solution.

Augmented Reality The virtual dressing room proposed in the report is foundational solution using augmented reality and requires further work and research in itself. However, once it is improved to a level sufficient for implementation, it would significantly improve the customer experience. This is envisioned to be used after the customer uses the system, proposed in this report, to determine their full body parameters. The customer would be able to use the virtual dressing room to get an accurate understanding of how different clothes would look on them, without needing to physically try them on. This is expanded on in Section(Rec - Virtual Dressing Room)

Performance Evaluation of the 1st and 2nd Generation Kinect For Multimedia Applications [6]

Overview This report evaluates the performance of the Xbox 360 Kinect (First Generation - v1) and the Xbox One Kinect (Second Generation - v2). The launch of the Kinect v1 in 2010 was revolutionary as it made depth acquisition more affordable and accessible to a wide spectrum of users. Further excitement was created when Microsoft launched the Kinect v2 only three years later. However, the technology used for depth acquisition in the second generation Kinect was completely different to that of the first. The first generation Kinect utilised structured light which is a well known technology used by industry grade laser scanners for 3D construction. The second generation Kinect, on the other hand, uses a Time-Of-Flight (ToF) camera. This drastic changed piqued interest about the performance difference between the sensors and their ideal applications. Consequently, this report is among many written to quantify the difference.

Relevance This report provided in depth knowledge about the principle of operation of the Kinect used in this project, together with its successor. This information contributed both to the background theory for the Kinect's operation and to component selection.

Background Theory The first generation Kinect which is utilised in this project, operates on a principle known as structured light. This knowledge is provides an essential base of understanding that allows for the effect manipulation of data retrieved from the sensor. Also, it provides more information regarding the limitations of the technology. As such, ideal operating conditions can be determined which greatly aids in experimental design. More detail is given in Section (Sensor Background).

Component Selection This report aided in the selection of a component for the design. The results yielded that the second generation Kinect on average performs better than the first generation Kinect. However, a large amount of performance testing was done in extreme conditions such as in direct sunlight, in darkness, at far distances etc. Consequently, significant disparity in the performance occurred mostly in these special conditions. For the purposes and operating conditions of the project, the first generation Kinect is sufficient. Elaboration into the evaluation between the two sensors is done in Section Component Selection

Kinect for Windows Sensor Components and Specifications [7]

Overview This technical documentation provided by Microsoft details the components that comprise the Kinect sensor and its specifications. This provides background information regarding the types of information it provides and a high-level description of how the sensor operates.

Relevance This high-level description of the Kinect provides the foundation of background information that is required before development with the Kinect can take place.

Background Theory The essential information retrieved from this document include a description of each component segment of the Kinect and their respective specifications. The key details retrieved are as follows:

- The resolution of the RGB camera
- The operation of the depth sensor
- Frame rates for the depth and colour stream
- Viewing angle or field of view of the Kinect
- Vertical tilt range
- The accelerometer used to detect to determine the orientation of the Kinect

Depth Camera [8]

Overview This technical documentation provided by Microsoft Robotics details more specific operations and interfaces to interact with the depth camera. This is a generic document for all depth camera devices that are supported by Microsoft.

Relevance This document does not provide specific information regarding how to interact with the Kinect's API to interface with the depth sensor and manipulate depth data. However, it provides essential information regarding how depth data is stored in all Microsoft devices with this functionality.

Background Theory The key details extracted from this document that inform the use of depth data retrieved from the Kinect are as follows:

- A brief introduction to the depth image space.
- The information stored in depth data.
- The bit size of the depth data and the relevance of each bit.

Coordinate Spaces [9]

Overview This technical documentation provided by Microsoft details the coordinate spaces for the three streams of data available - Colour, Depth and Skeleton. This provides background information explains how to interpret the data explains the manner in which it is stored. It also provides a brief introduction to how to use the API to transform data between the different spaces.

Relevance This description expands on the introductory theory provided on the component composition of the Kinect. This forms the reference used for digital representation of real world information captured by the relevant sensing components. It delves deeper into the intricacies of each of the data types and provides the base for interpreting and manipulating the data algorithmically. The detailed explanation of this information is included in Section (Coordinate Spaces Ref). A brief expansion on the most significant information utilised from this document is seen below:

Colour Space All the real world data available in the field of view of the RGB camera is captured and available as a frame of data. The frame is made of a number of pixels depending on the resolution specified. Each pixel at a particular (x, y) coordinate on the colour frame contain values for red, green and blue colour.

Depth Space A depth frame is available that is a grayscale representation of everything in the field of view of the camera. Like the Colour Space, the frame consists of a number of pixels depending on the resolution. However, these pixels hold the distance of the nearest object in millimetres at a given (x, y) coordinate. The (x, y) coordinate correspond to the position of the pixel on the depth frame and have no direct correlation with dimensions in physical space.

Skeleton Space Depth image data is processed to produce a frame of skeleton data. All points in skeleton space are in 3D. The Kinect acts as the point of origin with a positive x-axis extending to the left, a positive y-axis extending upwards and a positive z-axis extending in the direction the Kinect is pointing. Skeleton data can also be used to track human skeletons and provide positional data of the skeleton.

Skeletal Tracking [10]

Overview This technical documentation provided by Microsoft explains the high-level functioning of the Kinect's skeletal tracking capabilities. It provides an overview of how the Kinect recognises people and various operating conditions for effective tracking.

Relevance Understanding how the skeletal tracking system operates is essential in understanding the strengths and weaknesses of using the Kinect for human body detection. Additionally, the information regarding operating conditions advises various aspects of the experiment design procedure.

Background Theory The skeletal tracking capability is a complex and multifaceted system. Before it can be successfully incorporated into another system or application, it is vital to obtain thorough knowledge of each level of functionality and operation. This document provides the starting point for this journey as it provides a condensation of the capability. More detail is given in Section (Skeletal Tracking - Ref)

Design This document also provides important insights into the operating conditions required for skeletal tracking. These are refereed to when designing the experiment and overall solution in Sections (Insert 2 refs to skeletal conditions). Examples of these operating conditions are listed below:

- The orientations of the human body for which the Kinect operates successfully as well as those with which it falters.
- The field of view restrictions of the Kinect

Tracking Users with Kinect Skeletal Tracking [11]

Overview This technical document provided by Microsoft provides a comprehensive walk-through and explanation on how to use the Kinect API to access skeletal information. It provides C++ and C# code snippets to successfully enable user tracking, identify user skeletons and access positional information.

Relevance The skeleton tracking capability of the Kinect forms the major foundation of this project. This document provides information that makes navigating the API seamless and provides foundational understanding for unpacking code samples available by Microsoft. It is therefore a crucial piece of information to familiarise oneself with the Kinect API before application development can commence.

Background Theory The skeletal information provided by the Kinect API is intuitive to understand and utilise. However, there are important nuances that must be understood to prevent unwanted errors during development. These details are discussed thoroughly in Section (Insert Skeletal Ref - Joint states etc.).

Design This document proved critical in algorithm design of the solution. The nature of certain API interactions advised certain decisions regarding the application's class architecture and integration with other parts of the solution. Further explanations are given in Section (Algorithm Design - Skel Tracking).

Formula Derived Mathematically for Computation of Perimeter of Ellipse [12]

Overview The ellipse is a widely used shape that forms a crucial part of mathematics. Some of its uses include, but are not limited to, conic sections and Keplar's law od Planetary Motion. Despite, its prevalence in many applications, there does not exist a set formula that allows the perimeter to be calculated from the axes of the ellipse. Instead, all methods used to date for calculation, are approximations. This journal article, explores these existing methods, proposes a new formula that relates the radius to the perimeter and evaluates their performances.The proposed method performed very close to or better than many existing models. As such, the formula relationship can be utilised and allows for efficient perimeter determination using the major and minor axes.

Relevance The second major part of this project is the modelling of 3D circumferences using measurements obtained by the system. A model to determine these circumferences must be developed and as such, various mathematical shapes or parametric models should be tested. The ellipse is one such shape.

Design It is theorised in this project that a body part can be modelled within a certain error margin using the ellipse as a model. As such, the formula proposed in this article was utilised forms a fundamental part of the modelling stage of the algorithm. This is expanded on in Section (Ellipse Ref - Design).

Programming Guide + Documentation
Uncertainty
Clothing
Parametric Model

Definitions!!!!!!!!!!!!!!!!
Accuracy
Precision
Kinect
Volunteer
Operator
IR - Infrared Rays
API
RGB
SDK

When writing your review start of with the general concepts and move to the more specific aspects explaining the necessary theory as you go. This section is NOT a copy and paste from others work or a rewrite-but-change-one-word section. I suggest you read all your material, and then put it down and write this section, referring back to the work only when you need to check something.

See your PCS textbook for more details on how to write a literature review.

If you include a figure or a table in your text please see the example in Fig. 2.1 as to how to caption it. Please make sure that all text in your figures is readable and that you reference your figures if they are from another source.

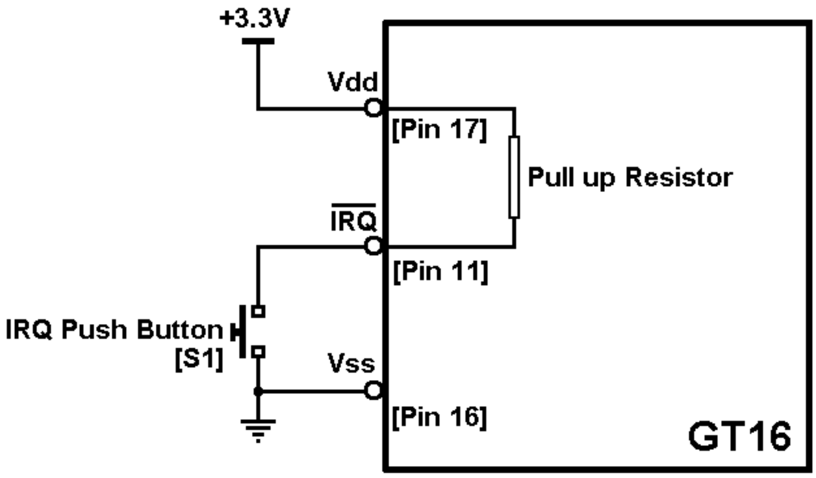

Figure 2.1: A block diagram illustrating the connections to the IRQ pin on the MCS08GT16A microcontroller (Please note that your headings should be short descriptions of what is in the diagram not simply the figure title)

2.2 Background Information

2.2.1 Kinect

Overview

The Kinect was developed and launched in 2010 by Microsoft. At the time of its release, it revolutionised depth acquisition technology as it was the first depth sensing device that was available to the consumer market. This low-cost, freely available technology made depth sensing accessible and caused an explosion in the use of of depth data for various applications. [6]

Versions

Kinect for Xbox 360 The first generation Kinect was launched in 2010. It was released for the Xbox 360 gaming console. It is officially called the Kinect for Xbox 360. [6] A picture can be seen in Figure 2.2. However, it is referred to by various names in different literature. Below are different names used to refer to the Kinect for Xbox 360:

Figure 2.2: Kinect for Xbox 360 [6]

- First Generation Kinect
- Kinect Version 1 (v1)
- Xbox Kinect 360

Kinect for Xbox One The second generation Kinect was launched in 2013. It was released with the Xbox One gaming console and is, thus, officially called the Kinect for Xbox One. [6] A picture can be seen in Figure 2.3. It is also referred to by different names, which are listed below for completeness:

Figure 2.3: Kinect for Xbox One [6]

- Second Generation Kinect
- Kinect Version 2 (v2)
- Xbox Kinect One

Kinect for Windows For each generation of Kinect for Xbox released, a Kinect for Windows was released. The version numbers correlate (I.e. The Kinect for Windows also has a v1 and v2). Kinect for Windows was created to be used by a computer whereas the Kinect for Xbox to be used by the console. They are, however, functionally identical and thus, a Kinect for Xbox can be ”converted” to a Kinect for Windows by using a USB adapter. The only difference with this is that the Kinect for Windows supports ”Near Mode” whereas a Kinect for Xbox used with an adapter does not. (”Near Mode” will be explained further in Section 2.2.1)

Kinect Used In this project, the Kinect for Xbox 360 with a USB adapter was used for development. Further reasons for this component choice is given in Section (Component Selection)

Components and Specifications

The Kinect for Windows (Kinect for Xbox 360 with a USB adapter) consists of a stand and a housing for the sensing components. Between the housing and a stand is a motor that is able to adjust the vertical tilt of the housing. The Kinect is able to achieve a maximum viewing angle of 43° vertical by 57° horizontal. It also has a vertical tilt range of ±27°. [7]

The housing contains the following main components:

- An RGB Camera or Colour Sensor
- An IR Emitter
- An IR Depth Sensor
- A Microphone Array
- A 3-axis Accelerometer

The configuration of the stand, housing and internal components can be seen in Figure 2.4. A more detailed explanation of each of the components necessary for this project is included below:

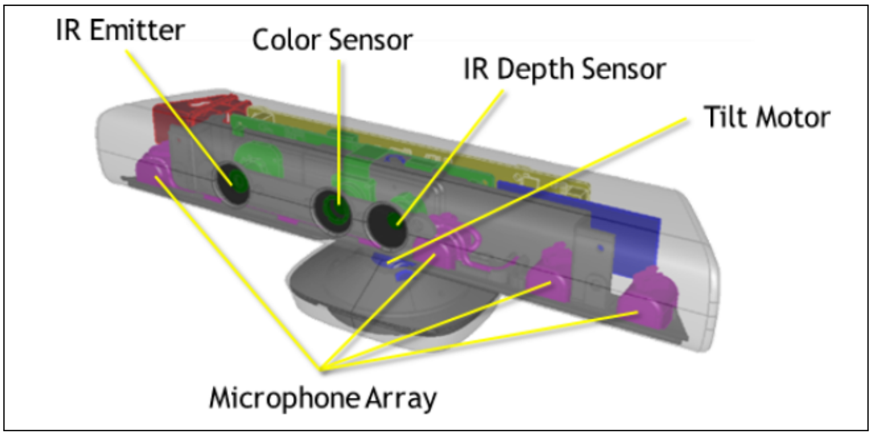

Figure 2.4: Internal component layout of the Kinect for Xbox 360 [7]

Colour Sensor The RGB camera has a maximum resolution of 1280 x 960 pixels. Each pixel is able to store red, green and blue data, allowing the camera to capture a colour image. [7]

Depth Sensor The depth sensor is made up of an infrared (IR) Emitter and an IR depth sensor. The emitter projects infrared light beams across the entire field of view of the Kinect. These beams hit various surfaces across the field of view and are reflected back at the Kinect. The IR depth sensor detects these reflected beams and converts them into depth data indicating the distance between the Kinect and a particular object. The collection and processing of all the reflected beams enables the Kinect to produce depth images with a maximum resolution of 640 x 480 depth image pixels. [7] An extension on the depth sensor is provided in the section below if more information on the intricacies of the depth sensor is required:

Extension on Depth Sensor The depth sensor relies on a principle known as structured light. The IR emitter projects a light pattern at a wavelength of 830 nm that form an image of pseudo-random located dots. The IR depth sensor, which is an IR video-camera, also runs on a wavelength of 830 nm and can detect the light pattern. The Kinect also possesses a reference pattern that displays the light pattern at a fixed distance. This pattern is used for calibration and is used to determine depth by comparing the reflected light pattern detected with the reference pattern. [6] This is illustrated in Figure 2.5.

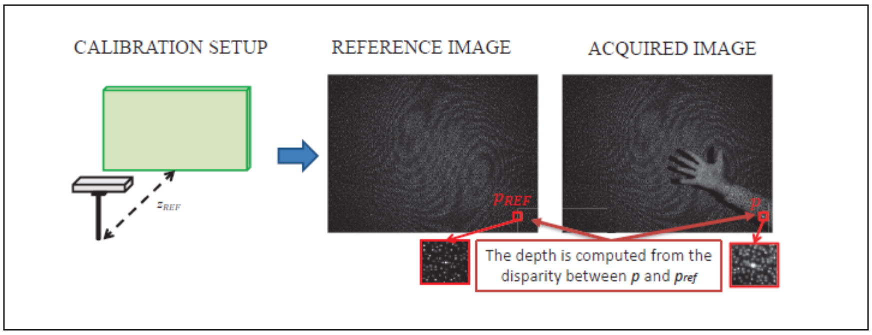

Figure 2.5: An illustration of the process used by the depth sensor to determine depth by comparing the reference image to the detected one. [6]

Coordinate Spaces

The Kinect is able to stream 3 types of data - colour, depth and skeleton data. This streamed data is available in the form of data frames that occur one at a time. The Kinect API also provides functionality that converts data between the different spaces. [9] The sections below will elaborate on each of the spaces and will conclude with a synopsis of the conversion functionality available:

Colour Space The colour sensor is able to produce frames at a rate of 30 frames per second (fps). [7] Each frame captured consists of the colour image of everything inside the field of view of the colour sensor. The frame is made of pixels. As mentioned in 2.2.1, these pixels hold red, blue and green data. The number of pixels in the image depends on the resolution selected and each pixel is uniquely identified by an (x, y) coordinate. These (x, y) coordinates form the colour space. [9]

Depth Space Similarly to the the colour sensor, the depth sensor is also able to produce frames at a rate of 30 fps. [7] Each frame captured also retains the information of everything in the field of view of the depth sensor and is in the form of pixels, whose size is determined by the resolution selected. However, the difference with the depth sensor is that the data captured is a grayscale image and each pixel at a particular (x, y) coordinate, contains the Cartesian distance in millimetres from the camera plane to the nearest object, instead of red, blue and green data. The mapping of the distances to the depth space can be seen in Figure Much like the colour space, the collection of (x, y) coordinates form the depth space (I.e. The location of the depth pixels in the depth frame). The (x, y) coordinates, therefore, do not correspond to physical or Cartesian coordinates in the real world. [9]

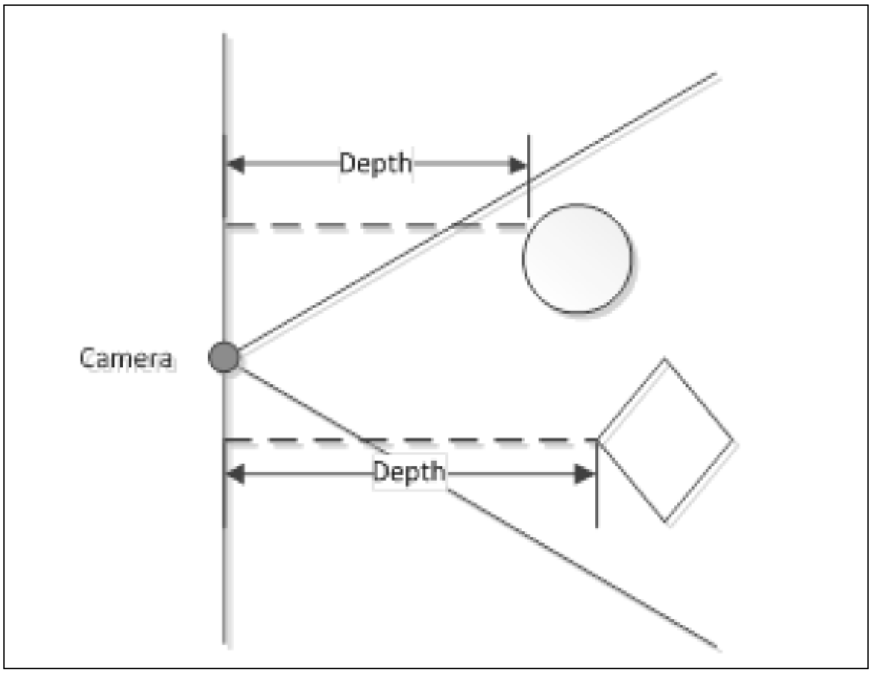

Figure 2.6: An illustration of the distances that stored in each depth pixel, at a particular (x, y) coordinate. [8]

Range The depth sensor has two ranges - default and near range. The default range is available for both the Kinect for Windows and the Kinect for Xbox. However, as mentioned above, the near range is not available in the Kinect for Xbox. Therefore, the Kinect for Xbox use with a USB adapter will still still not have near range capabilities. The distances associated with the two ranges can be seen in Figure 2.7.

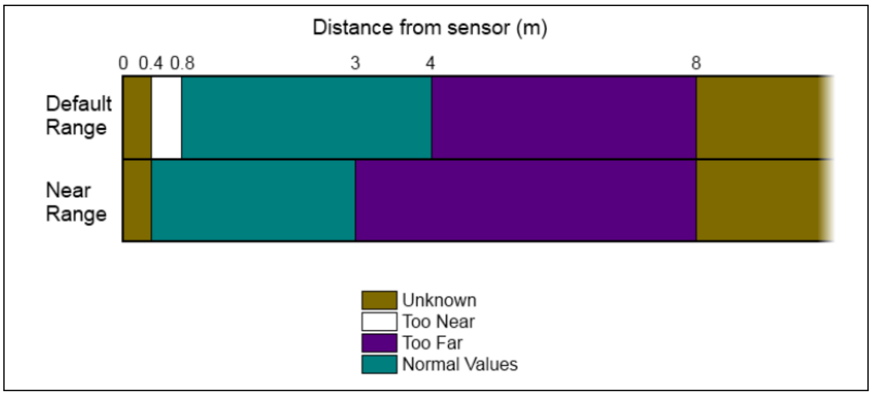

Figure 2.7: An illustration of the distances ranges for the default range and near range modes of the Kinect. [9]

Depth Data Storage As mentioned above, the depth data stored in each pixel contains the distance from the depth sensor to the nearest object in millimetres. Each pixel is made up of 11 bits of information. The 3 least significant bits contain the player index and the remaining 8 bits contain the distance. [8] The player index will be explained in Section 2.2.1.

Also, as seen in Figure 2.7, there are regions of depth data in which they are classified as out of range. If a piece of depth data falls into these ranges, they are given a special value. [9] These values are listed below:

- Too Near - 0x0000
- Too Far - 0x0FFF
- Unknown - 0x1FFF

Skeleton Space A frame of skeleton data is produced by processing a depth frame at runtime. If a person is visible in the depth sensor, skeleton data will be generated that contains 3D position data for their skeleton. Up to two human skeletons can be tracked. (Tracking is further explained in Section 2.2.1). For a tracked skeleton, all position data for their skeleton and joints are represented as 3D coordinates - (x, y, z). These coordinates are expressed in metres. The axes used to define this coordinate system is seen in Figure 2.8. In this coordinate system, the position of the Kinect defines the origin with the positive z-axis extending in the direction the Kinect is pointing. The positive x-axis extends to the left of the Kinect and the positive y-axis extends upwards. [9]

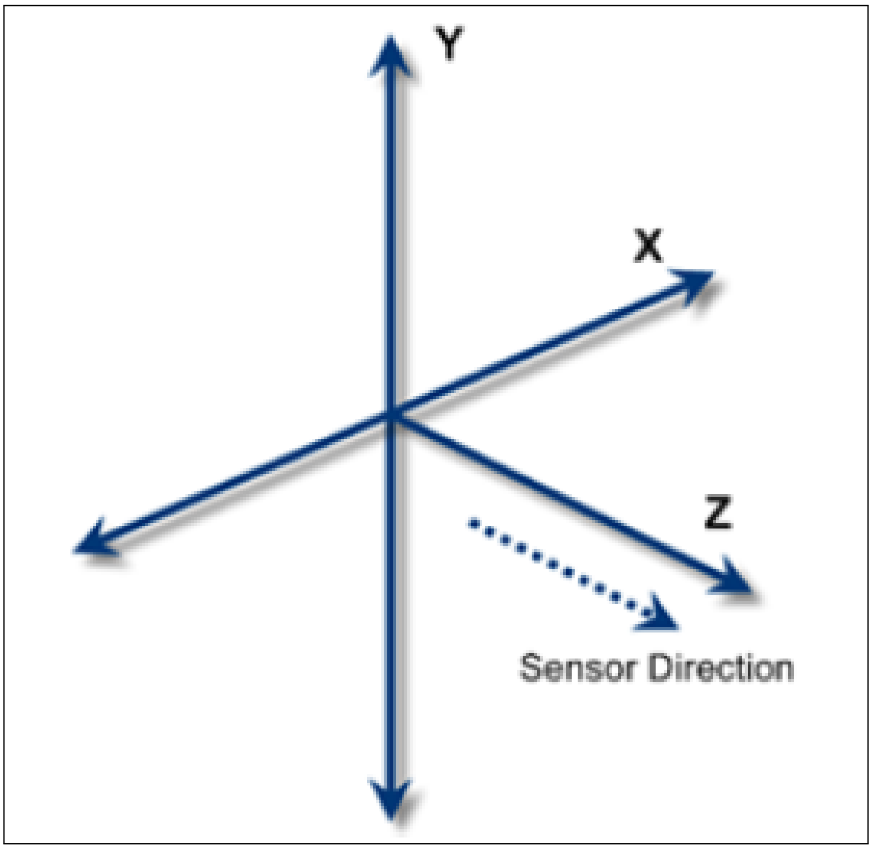

Figure 2.8: An illustration of the axes used for skeleton space. [9]

Mirroring During generation of a human skeleton, the skeleton system mirrors the tracked user. This means that the left and right sides of the skeleton are inverted. This also implies that the direction of a person facing the Kinect is considered to be the -z direction. [9]

(Clipping Paragraph - Maybe)

Conversion Skeleton position data holds 3D positions and as such, can be thought of as real world positions. For colour and depth data, these real world positions are mapped to 2D planes. Therefore, there is a relation between the different spaces. [3] This can be seen in Figure 2.9. However, depth data and colour data do not always map to the same (x, y) coordinate and thus a conversion must also occur between them. [8] Fortunately, these functionality for conversion between the different spaces is provided in the Kinect for Windows SDK. A link to the documentation can be seen in Section (Code documentation).

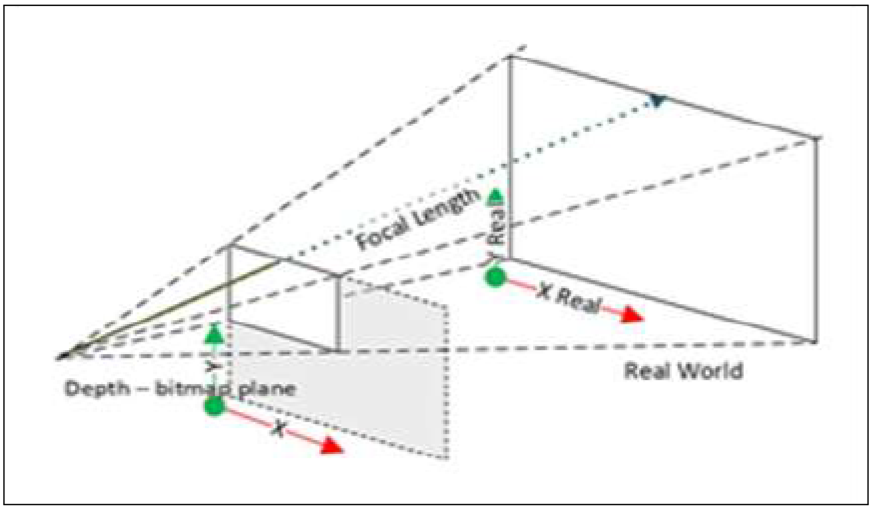

Figure 2.9: An illustration of the relation between the coordinate spaces for colour, depth and skeleton data. [3]

Skeleton Tracking

An inbuilt functionality of the Kinect is skeletal tracking. The skeleton system uses the depth sensor and allows the Kinect to recognise different people in its field of view and follow their actions. The Kinect can recognise up to six people in its field of view at a given time and track up to two of these users in detail. For these two tracked human skeleton, their joints are also which yields information regarding their position and movement at different time intervals. [10]

User Recognition In order to active the recognition of the Kinect's skeletal tracking system, a person needs to be in front of the sensor with their head and upper body visible. Recognition begins automatically without any specific calibration required. In terms of the orientation of the person's body, the system functions most effectively when a person is standing or sitting, and directly facing the Kinect. Sideways poses are difficult for the system to process as the there are parts of the body not visible. [10]

Field of View In default range mode, the skeletal tacking system has the same range as the depth sensor (Data is clearly detectable between 0.8 and 4 metres). However, users will often have to use their arms and thus, the suggested practical range for effective tracking is between 1.2 and 3.5 metres. An illustration of the field of view can be seen in Figure 2.10.

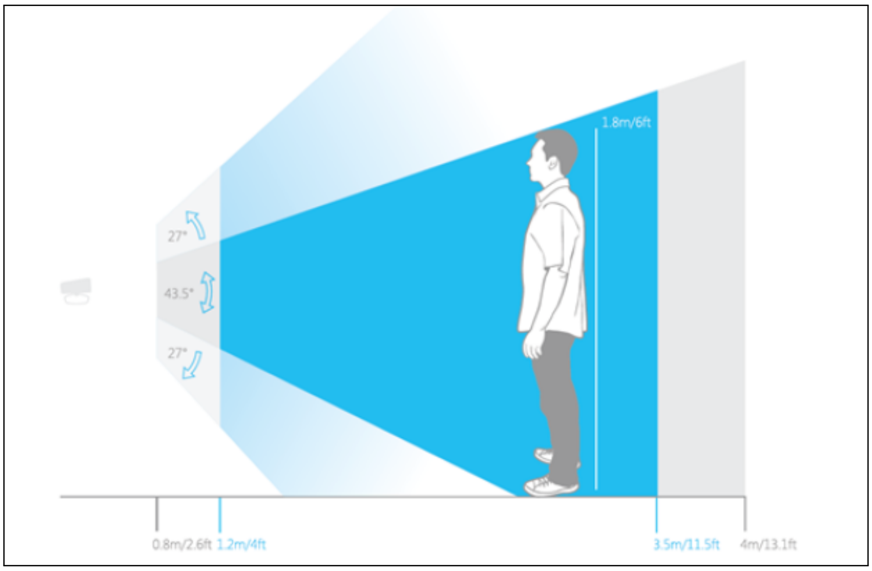

Figure 2.10: An illustration of the suggested practical field of view for effective skeletal tracking using the Kinect. [10]

API Support When using the skeletal tracking system, and users are present, a skeleton frame of data is produced. This frame holds the data of up to six people in the field of view. In the frame, all skeletons of detected users can only have two states - "Tracked" or "Position Only". A "Position Only" skeleton provides information about the position of the user, but no joint information. A "Tracked" skeleton on the other hand, provides detailed information about the position, in the field of view of the sensor, of the user and twenty joints in his/her body. [11] All the joints available and their positions, can be seen in Figure 2.11. Each joint can also have one of three states listed below:

- "Tracked" - A clearly visible joint.
- "Inferred" - A joint that is not clearly visible
- "Non-tracked" - A joint that is excluded (For example, lower joints in seated mode)

As mentioned in Section 2.1.1, full snippets of code required to track users with the Kinect skeletal tracking system, using the Kinect for Windows SDK, are provided in [11]. However, a summary of the process used in C# is given below:

1. Enable skeletal tracking in an active Kinect
2. Create an event handler to fire when a skeleton frame is ready

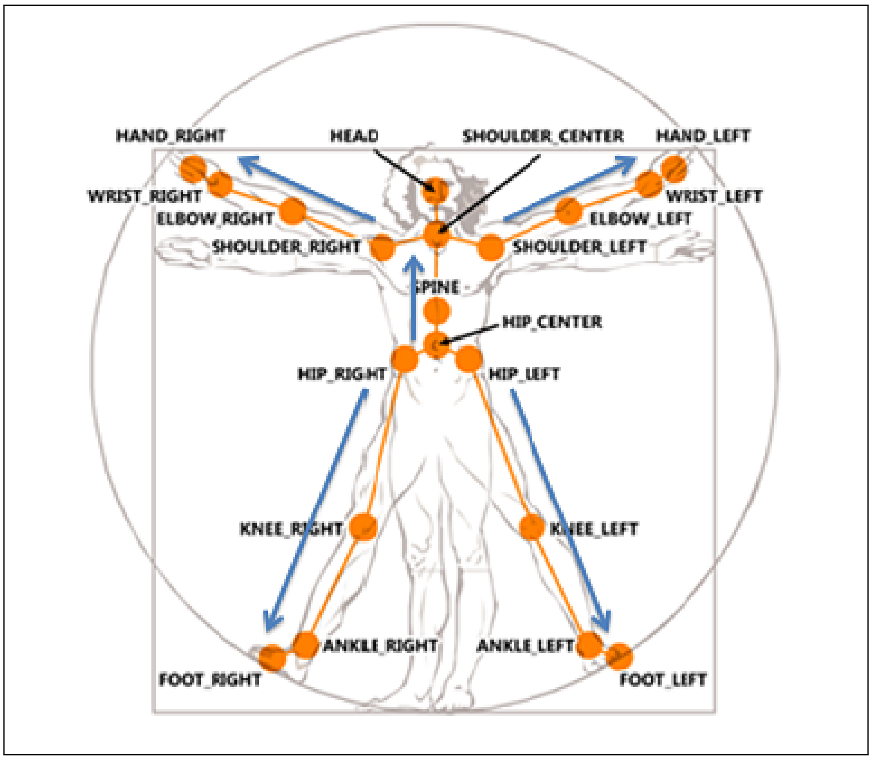

Figure 2.11: An illustration of joints available for a "Tracked" skeleton retrieved from a skeleton frame of data. [11]

3. When a skeleton frame is ready, copy the array of skeletons objects available from the frame to a new array for manipulation

4. Loop through the array to identify the "Tracked" skeletons

5. For a given tracked skeleton, loop through the array of joints provided for information of a single joint

6. Utilise the information of states and/or positions of the joints and skeleton for a given application

Known Errors Due to the structured light technology underpinning the depth sensor, there are certain environmental conditions that would affect its accuracy. Consequently, this would also affect the functioning of the skeletal tracking system. The following environmental conditions should be avoided as they significantly impair the Kinect's functioning: [6]

- The Kinect should not be used in a dark room
- The Kinect cannot operate with direct light shining in its direction (Should also not be used outdoors)

- Very reflective surfaces in the field of view of the Kinect can compromise readings

!!!!!!!!!!!!!!Guidelines for measurements

Internal Process + Noise

Middleware [3]
Joint Filtering

2.2.2 Measurement

3D Space

There are various theorems and axioms that allow for analysis of geometrical in Euclidean or three-dimensional space. One such theorem used for determining the three-dimensional distance between two points in space is the Pythagorean Theorem. Using the Cartesian coordinate system, this formula can be seen below: [3]

Let Point 1 = (x_1, y_1, z_1), Point 2 = (x_2, y_2, z_2) and the distance between Point 1 and Point 2 = d, then:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{2.1}$$

Accuracy

When recording measurements of a physical quantity, it is often useful to understand the difference between the measured value and the actual value of the physical quantity. This difference is refereed as the measurement error or accuracy. The formula used to represent this as a percentage is seen below: [3]

Let the Measured Value = $V_{Measured}$, the Actual Value = V_{Actual} and the Error = $E_{\%}$, then:

$$E_{\%} = \frac{V_{Measured} - V_{Actual}}{V_{Actual}} \times 100 \tag{2.2}$$

2.2.3 Microsoft SDK

Digital Information

Kinect Toolkit

BackgroundRemoval Class

Colour, Depth Class

2.2.4 Clothing

2.2.5 Modelling

Ellipse

2.2.6 Augmented Reality

2.2.7 Accuracy/Other Improvement

Chapter 3

Solution Design

3.1 Implementation Design

1) Online profile of people - Used for online shopping and retail shops
2) Take measurements at a retailer - Virtual Dressing room - A part of the shopping experience and will reduce hassle of trying on clothes
3) Amount wasted in trying on clothes or online returns?
4) Could be used for personalised tailoring 5) Example of UI - Explanation of how it works

3.2 Component Selection

1) Choice of Kinect - Compare to other depth sensors
- Kinect v1 vs v2 - Winddows vs Xbox 2) Choice of Windows SDK

3.3 Algorithm Design

1) Windows examples used - Background Removal, Colour Stream and Skeleton Tracking
- NB - Why BackgroundRemoval instead of own method 3D Points - No calibration

2) Run through of algorithm - Background Removed frame - Send image to separate class for processing - Create array with background removed pixels - Draw skeleton on image - Create axes for measurement - Perpendicular or straight depending on particular measurement

3.4 Experimental Design

- Constraints - Men, distance from Kinect, height of Kinect, Number of views, 3D Modelling, control distance - box of 0.5m - UI to run simulated dressing room - Volunteer to pose as instructed by person controlling UI - Take measurement of front - Take left - Take back - Take right - At each point, take actual readings with uncertainty - Note: Did not use correction in [3] - For one volunteer, take 5 readings in relatively the same pose - Determine uncertainty

Chapter 4

Methodology

This is what I did to test and confirm my hypothesis.

You may want to split this chapter into sub chapters depending on your design. I suggest you change the title to something more specific to your project.

This is where you describe your design process in detail, from component/device selection to actual design implementation, to how you tested your system. Remember detail is important in technical writing. Do not just write I used a computer give the computer specifications or the oscilloscopes part number. Describe the system in enough detail so that someone else can replicate your design as well as your testing methodology.

If you use or design code for your system, represent it as flow diagrams in text.

1. This is a bullet point test
2. I hope this works

4.1 Aim

The aim of this project was to create a system that enabled the measurement and 3D modelling of different parts of a human body through a well designed user interface, to aid in the choosing of clothing and/or custom tailoring.

To determine if the overall aim is met, the following three aims, each with their own unique sub-aims, will be tested:

1. The extremities and/or specific lengths of a human body can be measured and perform within the following criteria:
   (a) The overall measurement of the extremities of a human should perform within an average accuracy tolerance of 25%.
   (b) The measurement of lengths necessary for clothing choices should perform within an average accuracy tolerance of 20%.
   (c) Each measurement "view" should perform within an average accuracy tolerance of 25%.
   (d) Each individual limb measurement should perform within an average accuracy tolerance of 25%.
   (e) The presence of loose clothing should not affect the above performance criteria by more than 30%.
   (f) Each reading should have an uncertainty of less than 10%.
   (g) The system should be robust and able to accurately detect the human body and take all necessary measurements.
2. The measured extremities of the human body can be used to create 3D models of relevant body parts:
   (a) The circumference of modelled body parts should perform within an average accuracy tolerance of 30%.
   (b) Using an ellipse to model human body parts will yield more accurate results than using a rectangle model.
3. The user interface should allow for the successful processing of a person:
   (a) The user interface should be robust and limit the affect that user mistakes have on the measurement process.
   (b) The user interface should be easy to use and as autonomous as possible.

(Add BackgroundRemoval as test or aim)

Chapter 5

Results

These are the results obtained from the investigation outlined in 4. Seven volunteers were used in determining the accuracy of the system. They were first measured by the system and then their physical measurements were obtained for comparison. This chapter explores the results obtained for the measurement of the extremities of a human body and the modelling of 3D body parts, together with the observed performance of the user interface respectively.

5.1 Length and Extremity Measurement

This section begins with a presentation of the overall results of the system and a comparison with the aim of the investigation. Subsequent subsections present further performance of specific areas of the system that form the basis of analysis presented in Chapter 6. These sections focus on the following major areas of the system:

- The accuracy of measuring key lengths.
- The accuracy of each view of measurement (Front, Left, Back and Right) for measuring extremities.
- The accuracy of measuring the extremity of each individual limb.
- The impact that clothing worn by a person being measured has on the system's accuracy.
- Other empirical insights obtained through use and observation of the system.

5.1.1 Overall Results

Below is a summary of the aggregate accuracy of the system and the accuracy per volunteer, together with their personal characteristics. (Table 5.1)

Table 5.1: Overall results of accuracy of system per volunteer

Volunteer Number	***Average Error***	***Build***	***Height***	***Clothing***
1	28.44%	Athletic	Tall	Loose
2	17.27%	Athletic	Average	Tight
3	16.45%	Athletic	Tall	Vest
4	40.75%	Slim	Average	Loose
5	20.72%	Big	Average	Loose
6	19.60%	Slim	Short	Tight
7	18.83%	Big	Average	Vest
Total Average Error			23.15%	

As seen in Table 5.1, despite the average error of some individual volunteers being outside the desired range of 25% accuracy, the total average error of the system is 23.15%. Therefore, the system performed such that it met the requirements of achieving an accuracy better than 25%, as stipulated in section 4.1.

5.1.2 Length Performance

Seen below in Table 5.2 are the results of the average accuracy of key lengths obtained after measuring the volunteers in the "Front" view. These lengths are used using joint locations determined by the Kinect's Skeleton Tracking. Insert reference

As seen in Table 5.2, all length measurements fell well within the desired accuracy range of 20%. The most accurate length on average was the "Left Leg" with an accuracy of 3.12% and the least accurate length on average was the Torso with an accuracy of 8.53%.

Table 5.2: Results of the average accuracy of key lengths per volunteer

Volunteer Number	***Left Arm***	***Right Arm***	***Left Leg***	***Right Leg***	***Torso***
1	6.59%	4.35%	5.64%	3.38%	19.52%
2	11.88%	7.66%	3.22%	10.53%	9.55%
3	12.75%	19.49%	0.54%	0.99%	12.15%
4	8.76%	10.64%	1.43%	0.07%	0.51%
5	2.25%	10.17%	8.35%	7.79%	5.64%
6	4.57%	3.08%	2.36%	3.17%	9.38%
7	2.29%	2.54%	0.27%	2.47%	2.98%
Total Avg Error	***7.01%***	***8.27%***	***3.12%***	***4.06%***	***8.53%***

5.1.3 View Performance

Each "view" (Front, Left, Back or Right) of the system has a unique set of characteristics. It is useful to investigate the performance of each of them to better understand their effectiveness in the system as a whole.

Seen below in Table 5.3 are the results of the average accuracy of each view obtained after measuring the volunteers in each of the respective views.

Table 5.3: Results of the average accuracy of each view per volunteer

Volunteer Number	***Front***	***Left***	***Back***	***Right***
1	21.82%	29.90%	32.61%	31.04%
2	16.58%	15.81%	20.45%	14.60%
3	17.27%	19.01%	15.53%	13.37%
4	35.58%	58.15%	33.58%	44.77%
5	21.06%	10.39%	24.92%	23.47%
6	11.65%	44.58%	14.09%	17.07%
7	20.72%	19.51%	20.60%	12.06%
Total Avg Error	***20.67%***	***28.19%***	***23.11%***	***22.34%***

As seen in Table 5.3, the accuracy of the views in descending order are as follows:

1. Front
2. Right
3. Back
4. Left

The only view that performed outside the desired accuracy range of 25% was the "Left" view with an average accuracy of 28.19%. The Front view performed the best with an accuracy of 20.67%.

5.1.4 Limb Performance

Each extremity being measured is subtly different due to various factors such as position, orientation and size. Therefore, understanding how the system performs in these different cases is useful.

Due to the large number of measurements being taken, the results have been split up in terms of upper and lower body measurements.

Below is a summary of the aggregate accuracy of the system in measuring upper body limbs per volunteer: (Table 5.4)

Table 5.4: Results of the average accuracy of Upper Body Limbs

Volunteer Number	***Chest***	***Upper Left Arm***	***Lower Left Arm***	***Upper Right Arm***	***Lower Right Arm***
1	49.50%	29.90%	10.10%	26.35%	11.02%
2	27.84%	23.16%	9.84%	17.74%	13.24%
3	14.58%	10.99%	24.14%	7.33%	12.27%
4	79.55%	12.20%	26.74%	33.79%	5.47%
5	45.85%	21.22%	8.41%	15.16%	14.82%
6	23.39%	6.64%	15.91%	8.46%	8.63%
7	30.70%	12.34%	12.71%	12.02%	7.95%
Total Avg Error	***38.77%***	***16.64%***	***15.41%***	***17.26%***	***10.48%***

As seen in Table 5.4, measurements of the lower and upper portions of both arms fell well within the desired accuracy of 25%. However, the chest measurements seemed to be significantly more inaccurate with an average of 38.77%, which is far outside the desired range.

Below is a summary of the aggregate accuracy of the system in measuring lower body limbs per volunteer: (Table 5.5)

Table 5.5: Results of the average accuracy of Lower Body Limbs

Volunteer Number	***Waist***	***Upper Left Leg***	***Lower Left Leg***	***Upper Right Leg***	***Lower Right Leg***
1	32.09%	31.47%	13.58%	28.07%	44.04%
2	25.36%	15.32%	14.18%	13.21%	6.61%
3	20.51%	13.09%	24.62%	20.01%	10.12%
4	80.55%	13.36%	27.94%	25.40%	48.15%
5	26.74%	16.08%	8.56%	24.17%	15.81%
6	35.04%	20.78%	43.78%	9.58%	17.38%
7	19.53%	23.78%	22.77%	26.17%	16.17%
Total Avg Error	***34.26%***	***19.13%***	***22.20%***	***20.95%***	***22.61%***

As seen in Table 5.5, measurements of the lower and upper portions of both legs fell within the desired accuracy of 25%. This was similar to the behaviour of the arm measurements, mentioned above. However, they seemed to be slightly more inaccurate with all of the leg measurements having an accuracy tolerance of more than 19%, whereas the arms all fell within 17.5%. The waist measurement also followed the behaviour of the chest measurement above, where seemed to be significantly more inaccurate than the leg measurements. It had an average accuracy tolerance of 34.26%, which is also outside the desired range.

5.1.5 Impact of Clothing

The nature of the system is such that it is sensitive to clothing worn by the measured party. One aim of the system, as stipulated, in section 4.1, was to make the system more robust to clothing, such that the results are less prone to error.

As such, each of the above measurement results tables have been adjusted to remove the effects of clothing. This has been done by analysing the data set after volunteers with observed "loose" clothing were removed.

Below is the clothing adjusted version of Table 5.1.

Table 5.6: Overall results of accuracy of system per volunteer after adjustments for clothing

Volunteer Number	***Average Error***	***Build***	***Height***	***Clothing***
2	17.27%	Athletic	Average	Tight
3	16.45%	Athletic	Tall	Vest
6	19.60%	Slim	Short	Tight
7	18.83%	Big	Average	Vest
Total Error Average			***18.04%***	

It is clear from Table 5.6 that the system performs significantly better when loose clothing is not worn by the volunteer. The new overall average accuracy of the system improved to 18.04%, which is well within the desired range of 25%. This improvement, from 23.15%, due to the removal of 3 Volunteers with loose clothing means the average accuracy of these Volunteers is 29.96%. Therefore, clothing worsened the overall accuracy of an individuals reading by approximately 66.08%. This falls outside the desired range of 30%.

The same behaviour can be observed for the accuracy of the views and the different limbs.

In the adjusted version of Table 5.3 (Table 5.7), the performance of all the views improve. Additionally, the "Left" view now also falls within the desired 25% range and all the other views fall within 18%.

As for the adjusted upper and lower body measurements (Table 5.8 and Table 5.9 respectively), all of the measurements, except for the "Lower Left Leg" measurement (increased to 26.34% which is outside the desired range) either improved in accuracy or remained within 1% of its previous accuracy. The "Chest" measurement is now within the desired accuracy range of 25% with a value of 24.13%. The "Waist" measurement also greatly improved, however, is still narrowly outside the desired range with an accuracy of 25.11%

Table 5.7: Results of the average accuracy of each view per volunteer after adjustments for clothing

Volunteer Number	***Front***	***Left***	***Back***	***Right***
2	16.58%	15.81%	20.45%	14.60%
3	17.27%	19.01%	15.53%	13.37%
6	11.65%	44.58%	14.09%	17.07%
7	20.72%	19.51%	20.60%	12.06%
Total Avg Error	***16.55%***	***24.73%***	***17.67%***	***14.27%***

Table 5.8: Results of the average accuracy of Upper Body Limbs after adjustments for clothing

Volunteer Number	***Chest***	***Upper Left Arm***	***Lower Left Arm***	***Upper Right Arm***	***Lower Right Arm***
2	27.84%	23.16%	9.84%	17.74%	13.24%
3	14.58%	10.99%	24.14%	7.33%	12.27%
6	23.39%	6.64%	15.91%	8.46%	8.63%
7	30.70%	12.34%	12.71%	12.02%	7.95%
Total Avg Error	***24.13%***	***13.28%***	***15.65%***	***11.39%***	***10.52%***

Table 5.9: Results of the average accuracy of Lower Body Limbs after adjustments for clothing

Volunteer Number	***Waist***	***Upper Left Leg***	***Lower Left Leg***	***Upper Right Leg***	***Lower Right Leg***
2	25.36%	15.32%	14.18%	13.21%	6.61%
3	20.51%	13.09%	24.62%	20.01%	10.12%
6	35.04%	20.78%	43.78%	9.58%	17.38%
7	19.53%	23.78%	22.77%	26.17%	16.17%
Total Avg Error	***25.11%***	***18.24%***	***26.34%***	***17.24%***	***12.57%***

5.1.6 Uncertainty

The uncertainty of each limb measurement in each view can be seen in Table 5.10 and Table 5.11. As mentioned in (Insert Reference), these were calculated assuming a Gaussian distribution of measured values.

Note: The values of #N/A represent values that are not available in the respective view - I.e. The left hand side of the body cannot be seen when the volunteer is orientated right and as such are not available for measurement in that view

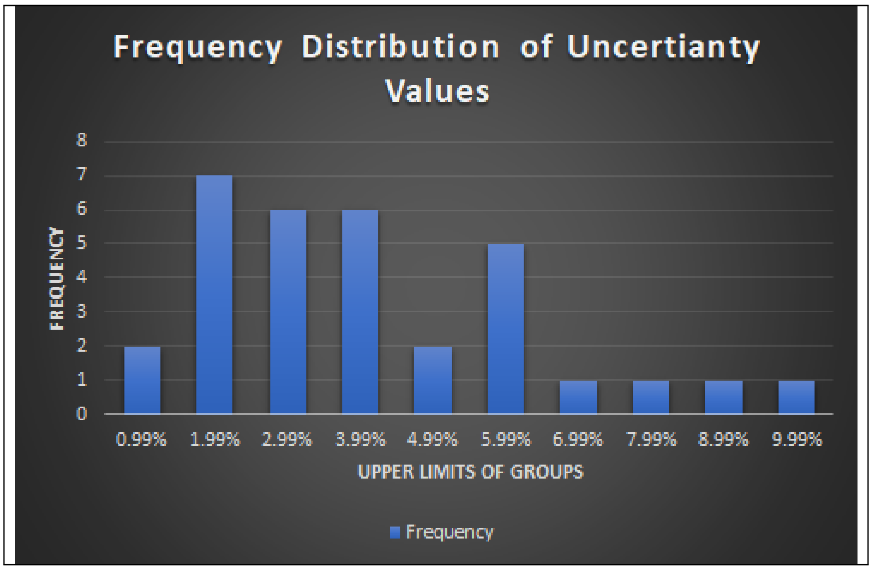

Figure 5.1: A histogram depicting the frequency distribution of the uncertainty values displayed in Table 5.10 and Table 5.11

As seen in Figure 5.1 below, all the uncertainty values calculated for each limb in the different views fell within the desired range of < 10%. Additionally, majority of the values actually fell within the range of 0-6%.

Table 5.10: Results of the average uncertainty (U_n) of Upper Body Limbs per view

View	*Chest*	*Upper Left Arm*	*Lower Left Arm*	*Upper Right Arm*	*Lower Right Arm*
Front	1.19%	3.84%	9.37%	3.48%	5.28%
Left	2.03%	3.50%	6.82%	#N/A	#N/A
Back	1.64%	3.29%	5.41%	3.02%	4.47%
Right	0.91%	#N/A	#N/A	5.00%	5.74%
Total Avg (U_n)	***1.44%***	***3.54%***	***7.20%***	***3.83%***	***5.16%***

Table 5.11: Results of the average uncertainty (U_n) of Lower Body Limbs per view

View	***Waist***	***Upper Left Leg***	***Lower Left Leg***	***Upper Right Leg***	***Lower Right Leg***
Front	1.25%	1.26%	2.86%	3.41%	5.72%
Left	2.36%	1.34%	8.01%	#N/A	#N/A
Back	0.91%	2.42%	2.70%	1.16%	1.08%
Right	2.03%	#N/A	#N/A	4.62%	7.79%
Total Avg (U_n)	***1.64%***	***1.67%***	***4.52%***	***3.06%***	***4.86%***

Upon closer inspection of Table 5.10 and Table 5.11, it is clear that all the individual and average uncertainty values fell within the desired range of 10%. The most reliable reading (One with the smallest uncertainty) was the "Chest" measurement with a 1.44% uncertainty. As a group, the leg measurements seemed to be more reliable than the arm measurements as all of the average leg uncertainties fell within 5%. On the other end of the spectrum, the measurement with the greatest average uncertainty was the "Left Arm" with an uncertainty of 7.20%.

5.1.7 Other Empirical Observations

Apart from the above presented tables of results, other observations were made of the system during its use. These observations are explained in each of the following subsections:

Background Removal Error

There were instances where the Background Removal functionality of the Kinect performed with error. These errors presented themselves in two manners:

Extra Padding During early and final testing of the system, it was observed that the output background removed picture had extra padding around the detected human in some instances. This extra padding consisted of background pixels that should have been removed.

An example of an early test with the above error can be seen in Figure 5.2. In this image,

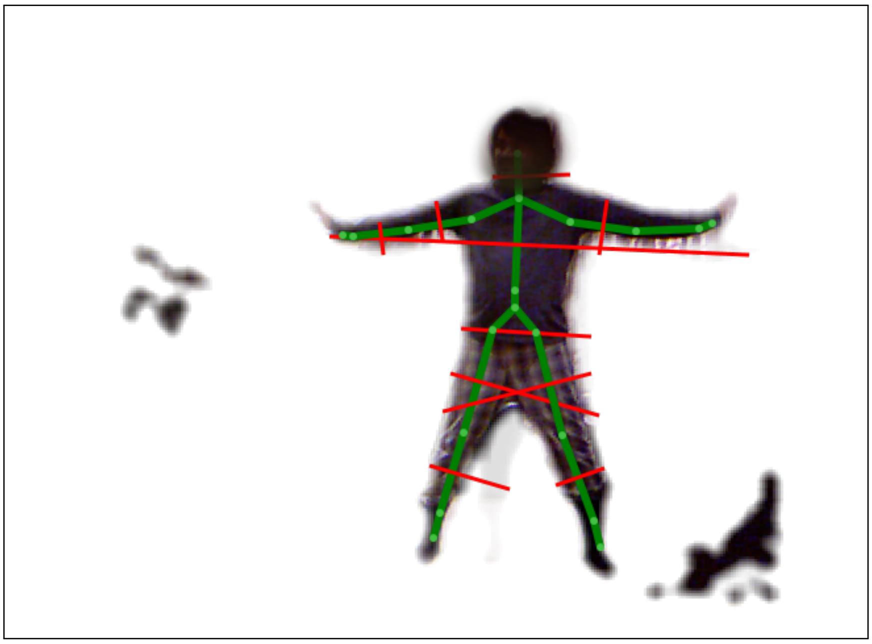

Figure 5.2: An example of failures of extremity measurements due to background removal and padding errors detected in early testing

it can be clearly seen that extra background pixels have been included in regions under the outstretched arms, on the sides of the torso and underneath the groin.

An additional caveat to the above error was also detected. It was observed that when the detected human was processed while moving, trailing pixels from the previous position would be included in the image and form part of the padding.

This was present in both early testing and final testing. In Figure 5.2, mentioned above, this can be clearly seen as the left leg has a "ghost" or "trailing" leg present. An example of a final test with trailing pixels can be seen in Figure 5.3. In this image, careful inspection of the extended hand reveals that trailing pixels are present due to movement of the detected human.

Incorrect Pixel Removal The other error present as a result of the Background Removal functionality is the incorrect pixel removal of pixels that form part of the detected human. The parts of the body that suffered the most frequently from this error are the parts at the extremes (E.g. Head, hands, "feet, etc.). Additionally, the view in which incorrect pixel removal from the "Head" was the most prevalent, was the "Back" view.

An example of a final test with incorrect pixel removal can be seen in Figure 5.4. Careful

Figure 5.3: An example of trailing pixels detected by background removal that forms part of padding errors

inspection of this image reveals that pixels that form part of the head and hands of the detected human have been incorrectly removed.

Overlap Error

It was observed that occasionally, the system failed to detect a correct extremity. As mentioned in Insert Reference, measuring extremities assumes that an outline of a limb will be against a background and thus, after the background is removed, will be clearly distinguishable. However, it was observed that in instances where an outline overlapped with another part of the body or the extra background padding mentioned in section 5.1.7 above, the outline would not be detected by the system.

A clear example can be seen in Figure 5.2. The "Chest", "Upper Right Leg", "Upper Left Leg" and "Lower Left Leg" in this image have been erroneously extended due to overlap errors. The "Upper Leg" measurements were extended due to overlap of the detected human's thighs, thus making their division indistinguishable. The "Chest" and the "Lower Left Leg measurement were both extended due to overlap caused by

Figure 5.4: An example of the presence of a background removal error relating to incorrect pixel removal during final testing

background padding. Respectively, they were caused by the padding underneath the arms and trailing pixels of the left leg.

Examples of this error during final testing can be seen in Figure 5.5 and Figure 5.6. In Figure 5.5, the overlap error is present in the "Chest" measurement as its plane of measurement intersected with the right arm, which was too low in this case. In Figure 5.6, the overlap error is present in the "Lower Right Leg" as its plane of measurement intersected with the "Lower Left Leg". The volunteer in this case was facing right for a "Right" view measurement. However, the volunteer's legs were not together and slightly ajar, thus creating an unwanted overlap.

Missing Measurements

During final testing, it was observed that occasionally, certain limb measurements were not processed properly and as a result, were not available. This manifested in certain limb measurements returning a value of "0cm", in its corresponding location on the user interface, when a non-zero value was expected.

Figure 5.5: An example of failures of extremity measurements due to background removal and padding errors during final testing

This error was observed in four individual limb readings belonging to two separate volunteers. The corresponding volunteer, view, limb details of the missing measurement are listed below, together with the figure in which it can be seen:

1. Volunteer 3 - Back View - Upper Right Arm - Figure 5.7
2. Volunteer 3 - Right View - Upper Right Arm - Figure 5.8
3. Volunteer 4 - Back View - Lower Right Arm - Figure 5.9
4. Volunteer 4 - Left View - Lower Left Arm - Figure 5.10

Incorrect Waist Plane

The last error that was observed during final testing was the occurrence of an incorrect waist plane when taking a measurement from either the "Left" or "Right" view.

Figure 5.6: An example of overlap error due to orientation during final testing

In Figure 5.11, the "Front" view of the volunteer is shown on the left side of the image and the "Right" view on the right side. It is evident that the waist plane in the "Front" view is correct as it is drawn across the hips in an almost horizontal fashion. The waist plane in all views should follow this pattern. However, the waist plane in the "Right" view is severely angled and deviates from the horizontal. As such, the measurement obtained would be incorrect and often inflated.

Additionally, the deviation from the horizontal (Or the angle fo the waist plane) varied depending on the rotation of the volunteers body, thus producing unpredictable results. An example of this changing deviation can be seen in Figure 5.12. Here, the "Left" view measurement of the same volunteer that appears in Figure 5.11 is shown. However, it is evident that the deviation of the waist plane in the "Left" and "Right" views is not consistent.

Figure 5.7: An example of a missing measurement - Volunteer 3 - Back View - Upper Right Arm

5.2 3D Modelling

This section details the results of the accuracy of the two models used for calculating the 3D circumference.

5.2.1 Ellipse Model

Table 5.12 and Table 5.13 are summaries of the average accuracy of using the ellipse model to the various upper and lower body limbs.

Volunteers 1, 2, 3 and 4 were used to test the ellipse model. However, the results of Volunteer 1 were erroneous due to a mistake made during recording of the results. (This mistake is dealt with in Section 5.3)

As seen in Table 5.12 and Table 5.13, the ellipse model was not successful in approximating the 3D circumference of the various limbs. All of the average accuracies calculated were above 60% and the worst performing measurement being the "Waist" with an average

Figure 5.8: An example of a missing measurement - Volunteer 3 - Right View - Upper Right Arm

accuracy of 148.70%. Clearly, the aim of modelling the circumferences with an average accuracy of within 30% was not achieved.

5.2.2 Rectangle Model

Table 5.14 and Table 5.15 are summaries of the average accuracy of using the ellipse model to the various upper and lower body limbs.

Volunteers 5, 6, and 7 were used to test the rectangle model.

As seen in Table 5.14 and Table 5.15, the rectangle model was more successful than the ellipse model in approximating the 3D circumference of the various limbs. All of the average accuracies calculated were below 50%, thus outperforming the ellipse model for every limb.

However, the rectangle model still did not completely meet the aim of modelling the circumferences with an average accuracy of within 30%. Only the "Lower Left Arm"

Table 5.12: Results of circumference modelling of upper body limbs using the ellipse model

Volunteer Number	***Chest***	***Upper Left Arm***	***Lower Left Arm***	***Upper Right Arm***	***Lower Right Arm***
2	129.87%	54.58%	104.72%	69.27%	97.32%
3	101.21%	111.53%	48.71%	#N/A	105.13%
4	199.89%	103.06%	#N/A	50.86%	59.77%
Total Avg Error	***143.66%***	***89.72%***	***76.72%***	***60.06%***	***87.41%***

Table 5.13: Results of circumference modelling of lower body limbs using the ellipse model

Volunteer Number	***Waist***	***Upper Left Leg***	***Lower Left Leg***	***Upper Right Leg***	***Lower Right Leg***
2	135.79%	117.74%	72.41%	107.70%	75.23%
3	119.85%	103.90%	110.44%	80.61%	95.57%
4	190.44%	113.02%	171.66%	180.64%	156.81%
Total Avg Error	***148.70%***	***111.55%***	***118.17%***	***122.98%***	***109.20%***

Table 5.14: Results of circumference modelling of upper body limbs using the rectangle model

Volunteer Number	***Chest***	***Upper Left Arm***	***Lower Left Arm***	***Upper Right Arm***	***Lower Right Arm***
5	74.14%	49.58%	30.97%	30.74%	12.07%
6	33.08%	26.03%	36.51%	31.70%	24.64%
7	37.18%	16.77%	14.71%	35.72%	19.96%
Total Avg Error	***48.13%***	***30.79%***	***27.40%***	***32.72%***	***18.89%***

Figure 5.9: An example of a missing measurement - Volunteer 4 - Back View - Lower Right Arm

and "Lower Right Arm" had average modelling accuracies within this range (27.40% and 18.89% respectively). The "Upper Left Arm", "Upper Right Arm" and "Lower Right Leg" closely missed this target with accuracies of 30.79%, 32.72% and 30.51% respectively.

Additionally, it seemed that on average, the lower body limb modelling performed worse than the upper body limb modelling. The worst performing upper body modelling was the "Chest" with an average accuracy of 48.13%. On the lower body side, the "Lower Left Leg" performed the worst with an average accuracy of 49.28%.

5.3 User Interface Observations

This section details observations made about the user interface of the system during final testing.

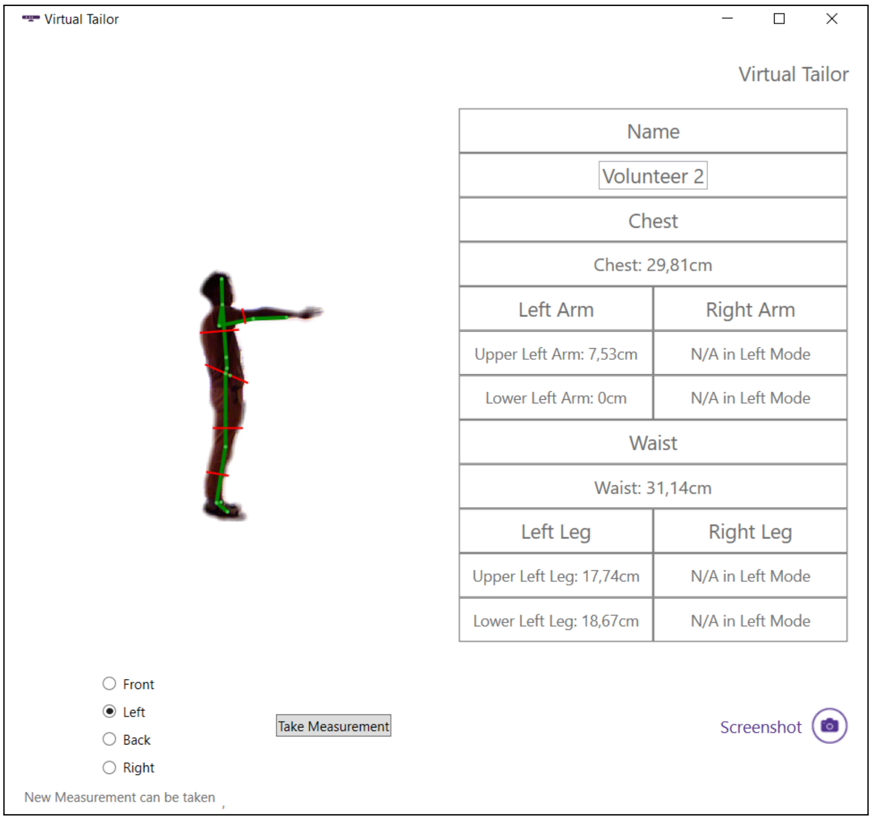

Figure 5.10: An example of a missing measurement - Volunteer 3 - Back View - Lower Left Arm

5.3.1 Measurement Processing

As mentioned in Section (Insert Reference), in order to take a measurement of a person standing in front of the Kinect, another person operating the system would need to press the "Measure" button.

Once the button is pressed, a colour image of the person with their skeleton and all relevant measurement planes is shown. Concurrently, all available measurements populate the measurement panel.

However, during final testing, an issue with this method of measurement processing was detected. In order to limit the empirical errors detailed in Section 5.1.7, the operator would have to look at the image obtained and advise the measured person on how to adjust their body for maximum results. The most frequent examples of this were:

1. An overlap error occurred - The operator would need to instruct the measured person how to orientate his/her body to remove the overlap

2. A trailing limb was detected - The operator would retake the measurement while

Figure 5.11: A comparison of the "Front" and "Right" view measurements volunteer 2 to illustrate the waist plane error

Table 5.15: Results of circumference modelling of lower body limbs using the rectangle model

Volunteer Number	***Waist***	***Upper Left Leg***	***Lower Left Leg***	***Upper Right Leg***	***Lower Right Leg***
5	48.53%	20.87%	27.08%	42.90%	29.98%
6	38.71%	58.73%	75.74%	35.62%	34.50%
7	33.14%	41.38%	45.03%	41.05%	27.05%
Total Avg Error	***40.13%***	***40.33%***	***49.28%***	***39.86%***	***30.51%***

the measured person is completely stationary

3. The skeleton was not detected (Specifically in "Left", "Right" or "Back" view) - The operator would ask the measured person to move until their skeleton was tracked

This manual "feedback loop" of image evaluation on the operator side and position correction on the measured person's side, to mitigate observed errors, meant that often multiple measurements would have to be taken before an "ideal one" surfaced. Each measurement would involve waiting for a measurement to be ready, pressing the button, waiting for the measurement to process and then evaluating the result. This repetitive action would sometimes be time consuming and tedious for both the operator and person being measured. As such, the aim of the interface being easy to use and autonomous was not achieved in this instance.

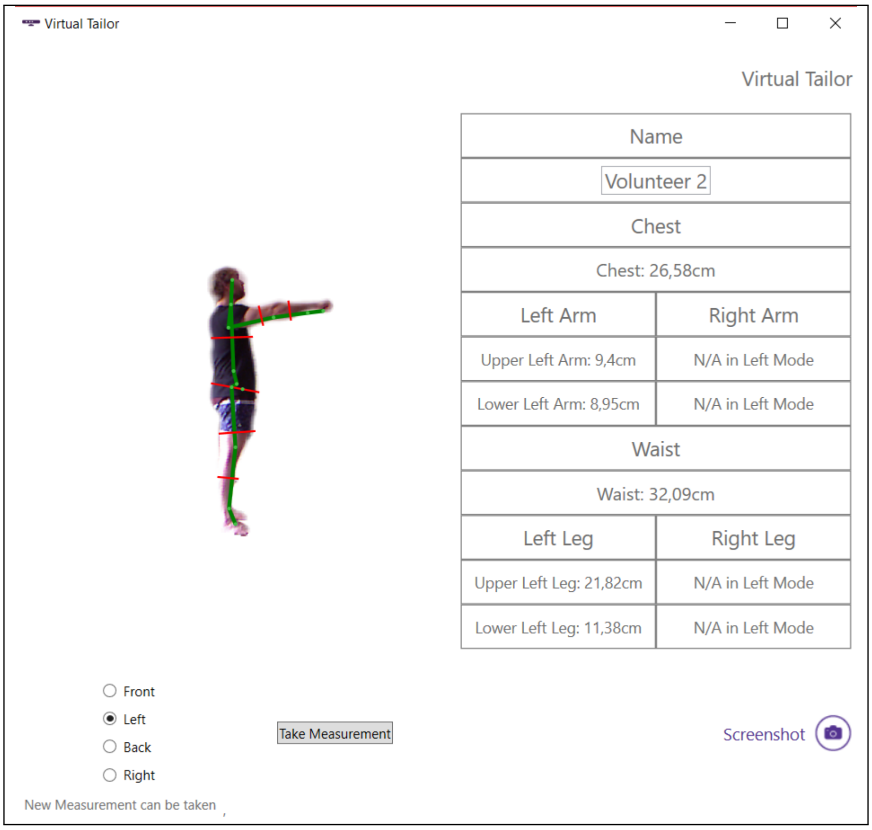

Figure 5.12: A comparison of the "Front" and "Right" view measurements volunteer 2 to illustrate the waist plane error

5.3.2 View Selection

To process a measurement in each view, the operator would first has to select the appropriate view in the radio button group in the UI and then complete the process mentioned in Section (Insert Reference) to record the measurement for that view.

The issue observed with this is that it relies on the operator to always select the correct view. There exists no view validation system that protects against the operator accidentally selecting the incorrect view or forgetting to select the new view before taking a new measurement.

For example, when testing the system on Volunteer 1, after the "Front" measurement was completed, the operator forgot to select the "Left" option before taking the "Left" measurement. As a result, the "Front" measurement was incorrectly overwritten and the circumference readings were unusable.

Therefore, the aim of the system being robust against user mistakes was not realised in this instance.

5.3.3 User Guidance

The operator is given the following signals during the operation of the process:

Status Bar

- Start up message when the program is initialised
- Message when a measurement is complete
- Message when a new view is selected
- Message when a new view can be taken

Buttons

- Measurement Button
- Results Button that appears when all measurements are taken

Results

- Image shown when measurement complete
- Measurement panel populated when measurement complete
- Results pop up window shown when results button pressed

The following observations were made about this guidance:

Status Bar

- Messages seemed to small and not easily noticeable
- "Measurement complete" message lasted for such a short time that it was often missed (Replaced by "new measurement available" message)

Pop Up Windows

- This seemed to be the best method in getting attention from user

Overall

- Little to no guidance was given regarding the quality of the measurement

Chapter 6

Analysis

This Chapter explores the insights gained through analysis of the results detailed in Chapter 5 and Appendix A.

The analysis follows the pattern of Chapter 5 by analysing the system in terms of its three major functional areas in the following sections:

- Length and Extremity Measurement
- 3D Modelling
- User Interface

6.1 Length and Extremity Measurement

This section begins with overall insights gained about this functional area. Afterwards, further analysis is done of each of the major sub-functions:

- Human Detection
- Extremity Detection
- Measurement Validation

6.1.1 Overall

The system performed mostly successfully in terms of the aims stipulated in Section 4.1.

Below is a summary of performance in comparison with the quantitative aims:

- The average accuracy tolerance of the overall system was 23.15% - (Aim: $< 25\%$)
- The length accuracies achieved ranged from 3.12% to 8.53% - (Aim: $< 20\%$)
- 75% of the Views met the "View" aim (The "Left" View achieved the worst accuracy of 28.19%) - (Aim: $< 25\%$)
- 80% of the limb measurements met the Limb aim (The Chest (38.77%) and the Waist (34.26%) performed the worst) - (Aim: $< 25\%$)
- Clothing, on average, increased the inaccuracy of the system by 66.08% - (Aim: $< 30\%$)
- The average uncertainty of each limb measurement ranged from 1.44% to 7.20% - (Aim: $< 10\%$)

Despite using a relatively small data set, the system was deemed to be effective for a variety of body types - The characteristics of Volunteers used for analysis included "Build", "Height" and "Clothing". (A summary of this can be seen in Table 5.1). However, it was noticed that certain characteristics may have been responsible for the systems lack of performance in certain areas and will be explained in subsequent sections.

Additionally, the physical measurements taken for comparison had a certain amount of uncertainty as the apparatus available for measurement was analogue and not perfectly suited for the the experiment. Therefore, the results of the system could be slightly better or worse. However, this increased level of accuracy is not necessary for the purposes of a first attempt at the project to get an overall understanding of the effectiveness of this approach. Methods to evaluate the accuracy to a finer level of detail would be essential for a final implementable solution.

6.1.2 Human Detection

Body Detection

The first major aspect of the system is whether it could adequately detect a human body. As mention in Section (Insert Reference), this was achieved by using the Kinect's inbuilt skeleton tracking capability together with its BackgroundRemoval Class.

The visible effectiveness of this method manifested in the detection of the full human body of each volunteer, together with a rendering of their skeleton, that appeared in the User Interface. (Refer to Appendix A for examples of images taken during final testing).

3D Measurement

The next process in the detection process was to understand if the depth data collected by the Kinect was reliable for 3D measurement. This has been validated by previous projects (Insert Reference), but the measurement of key lengths in the body, performed during this project, also acted as an indirect "acid-test" validation. This is largely due to the fact that the points used to measure the lengths were well established "Joints" in the tracked "Skeleton", where each "Joint" had a clearly identifiable 3D location. Therefore, comparing the measurements obtained through calculation to actual measurements was able to give a rough indication of the accuracy of the Kinect's data. Since all the key lengths had an average accuracy of within 10%, the data collected by the Kinect was deemed appropriate for not just human detection, but for measurement as well.

Kinect Failures

Despite the suitability of Kinect demonstrated above, there were certain instances where its performance was inadequate and negatively impacted the overall performance of the system. These are explained in the following paragraphs.

Range and Resolution The Kinect has a limited operating window and resolution for its depth data as mentioned in Section(Insert reference). This range is further limited in order to ensure the entire human body is detected.

The depth resolution used for this project was 640 x 480, which was the maximum available by the Kinect. The Kinect also has a maximum working range of 2.5m (I.e. Any further diminishes the quality fo the depth data). It was determined that at a height of approximately 1m (101.5 cm), most volunteers had to stand further at 2m or further to be detected. This limited window together with the limited depth resolution, caused a physical limitation on the accuracy of the system and is another reason for the decreased performance.

Detection It was observed that the Kinect's human detection and skeleton tracking was most effective when the volunteer was directly facing it. This is clear from the performance of the measurements in the "Front" view and the lack of miscellaneous errors such as missing measurements. This can be attributed to the fact that the Kinect performs the best when a person faces it head on (Parallel to the image plane). This is due to the Kinect being able to fully track the skeleton of the user. As a result, a more accurate and reliable skeletal coordinate system can be used, which in turn provide more accurate planes of measure.

The Kinect's ability to detect volunteers facing 90° (I.e. "Left" or "Right" view) from itself is limited and detecting volunteers facing 180° (I.e. "Back" view) from itself is even worse. During testing, it was noted that measurements in the different views was only possible if the volunteer's skeleton was already tracked (I.e. facing the Kinect at some point).

Additionally, the Kinect is not built to detect a person if he/she has his/her back to it. If the skeleton is tracked, due to the person facing it at some point, and they now have their back to the Kinect, the Kinect cannot differentiate this from the skeleton facing it directly. To compensate for this in the system, the left and right sides of the body were switched in the algorithm to simulate the turning of the body. However, it was noticed that the skeleton would not map to the body as well as if the person was facing it directly. Also, joints would often be "Inferred" or "Not Tracked". This is the reason for the Back View being prone to errors such as "Incorrect Pixel Removal" or "Missing Measurements" explained in Section 5.1.7.

Inferred Joints As mentioned in (Insert Reference), when the Kinect is unable to track a "Joint" due to it being out of view, the Kinect will "infer" its position. These "Inferred Joints" are useful in understanding characteristics of the detected person such as their 3D orientation. However, they often have a great deal of uncertainty in the modelling of their exact position and thus, for a single position and orientation of a detected person, the position of the ”Inferred Joint” could change drastically. This makes it undesirable for measurement purposes.

The view selection in User Interface is used to compensate for this as the algorithm neglects the body parts out of sight (For example, if the person has the left side of their body facing the camera, the skeleton on right side will not be rendered). The only issue with this is in determining the ”Waist” measurement in either the ”Left” or the ”Right” View. Since only one hip is clearly detected by the ”Kinect”, the other one is often ”Inferred”. The ”Waist” measurement is taken across both hips and due to the inaccuracy of the ”Inferred Joint”, the plane of measurement for the ”Waist” is unpredictable. This is what causes the ”Incorrect Waist Plane” error mentioned in Section 5.1.7.

To compensate for this, the operator of the User Interface often would instruct the volunteer to slightly rotate their body for a more accurate ”Waist” reading. However, this introduces other issues such as changing the axes of measurement for the limbs and occasionally causes ”Overlap Errors” in the ”Lower Leg” measurements. This is therefore not an effective empirical workaround of the problem.

Jitter This refers to the phenomenon observed where the rendered skeleton of the tracked person ”moves around” or ”jitters” when there is live tracking. Even if a person is relatively still, the ”Jittering” of the skeleton is still noticed. This is predominately due to the presence of noise in the measurement process of the person. This causes a slight variation in the skeleton of the tracked person at a given instant. Since the skeleton is used as the coordinate system for all measurements, this is not ideal as it introduces a level of uncertainty.

Additionally, it was observed that ”Inferred Joints” experience an even greater ”Jitter”. This makes them even more unreliable and further explains the occurrence of the ”Incorrect Waist Plane” error.

Skeleton Mapping It was occasionally noticed that the skeleton of detected human often did not map accurately to the colour image. This was caused due to issues mentioned above like ”Jittering” and the Kinect’s inability to detect a human facing away from it. The incorrect mapping causes both an inconsistency in the measurement planes and an error in a measurement due to the incorrect plane being used.

Additionally, it was noticed that due to the limited resolution of the Kinect's depth data and the limited workable distance from the Kinect, the skeleton mapping of "Skinny" individuals lower Arms and legs was poor. As seen in Section 5.1.7 under "Missing Measurements", Volunteer 4's "Lower Left Arm" and "Lower Right Arm" measurements were missing. Upon inspection of Figures 5.9 and 5.10, it is clear that due to the individual's smaller frame and the uncertainty in the skeleton mapping, it is actually mapped to underneath the actual arm (Outside the boundaries of the lower arm).

This meant that the skeleton was mapped to the "background removed" part of the image and caused the algorithm to believe the arm had no extremities. This in turn, caused the incorrect reading of "0cm". This is not a significant problem with larger framed individuals as the large surface area of their limbs ensures that even if their is uncertainty in the skeleton mapping, it is more likely to still map to within the boundaries of the limb.

6.1.3 Extremity Detection

As seen from the overall performance summary above, the system performed relatively well in detecting extremities of the body. As seen in Section (Insert Reference), the detection of extremities combined the ability of the Kinect's internal BackgroundRemoval Class, together with an algorithm that also took depth data into account.

However, there were certain cases mentioned in Chapter 5, where the extremity detection failed or was erroneous. The major contributing factors to this are explored in the following subsections:

Clothing

As mentioned in Section (Insert Reference), clothing worn by the volunteer was expected to have an effect on the system's performance.

However, what was not expected was the dramatic impact it would have on the results. On average, clothing worsened the accuracy of the results by 66.08%. This shows that the system is clearly not designed to be robust against clothing worn by the volunteers.

Fortunately, this can be compensated for by ensuring the person being measured wears either tight clothing or is wearing as little clothing as possible. However, in a fully implementable solution, more work should be done in including this compensation for clothing algorithmically. Possible methods are explained in Chapter 9.

Padding

This addition of pixels around the body (Introduced in Section 5.1.7), was responsible for increased inaccuracy in measurements of various limbs.

In most cases, the "Padding" caused the measurements to be inflated. The part of the algorithm created to compensate for this, effectively detects stray pixels that form part of the background that are too far from the body. The reason why it was also not excluded by the algorithm is because these pixels have approximately the same depth from the camera as the body. As a result, the BackgroundRemoval Class includes them, and the algorithm cannot differentiate between them and the body.

Overlap

The "Overlap" error was also a major contributor to inaccuracies. The system was unable to detect an overlap in the same way it could not detect extra padding - The overlap of two body parts meant that a continuous measurement line was created over both body parts as the expected removed background was not reached (Explained in section 5.1.7).

Additionally, the depth analysis part of the algorithm could not differentiate between the body parts as their average depth is too close. There were no other means to differentiate edges and such, this error would occasionally occur.

To compensate for this, the person being measured can be instructed to hold a specific position that limits the presence of overlap errors. However, this has impacts on the usability of the system and is covered in Section 6.3. Also, this does not always work as is the case with the "Left" and "Right" views. Besides the "Waist Plane" error mentioned above, overlap errors were a major contributor to the inaccuracies of these views.

Kinect Accuracy

The resolution and range limitation of the Kinect has an impact on the accuracy of extremity determination. (The person being detecting having to be a certain distance away from the sensor and the depth data not having a high enough resolution to provide very precise and accurate data). This manifests itself in two predominate manners:

Data Collection The low resolution of the depth sensor means the data collected about the exact position of the person has a certain level of uncertainty. Additionally, for finer measurements such as the precise measurements of the extremities, the lack of precise positional data causes error propagation throughout the system.

Errors The low resolution is also a contributor for many of the errors mentioned in Section 5.1.7. For example, the low resolution means the system has to be less sensitive to depth data. This makes it more difficult to differentiate between body parts (Overlap) and the background (Padding). Additionally, the depth data is prone to noise and contributes to errors in skeleton mapping, background removal and the part of the algorithm that analyses depth data.

6.1.4 Measurement Validity

Uncertainty

As seen above, the average uncertainty of the measurements of the all individual limbs was less than 10%. In fact, even all the individual uncertainties measured in the different views were also less than 10%.

This means that the system is relatively precise and there is minimal variation between multiple readings of a person in the same position. (I.e. a measurement obtained is reliable). That being said, the system currently only takes one reading of a person in each respective view (4 readings in total). Instead, multiple readings could be taken in each view and then aggregated to provide an even more reliable results. This is explored in Chapter 9.

However, the uncertainty may be slightly worse than the values calculated, as the volunteer used to measure the uncertainty was instructed to hold specific positions by the operator. These positions were selected to removed errors such as "Overlap" or the "Incorrect Waist Plane" error. Therefore, without the intervention of the operator, these uncertainty values may be higher. To implement a multiple reading system mentioned in the paragraph above, more stringent error checking must first be employed algorithmically.

Error Checking

At present, the only method of error checking is the operator validating whether a measurement is accurate enough or not (This "manual feedback loop" is explained in Section 5.3.1).

The problem with this method is that it requires a lot of effort on the part of the operator together with specific knowledge of errors for which to look. Additionally, the operator may be able to pick up obvious errors such as large overlaps or a waist plane with significant deviation. However, due to the large amount of information, it would be easy to miss small errors such as small overlaps, slight deviations in the waist plane and/or missing limb measurements.

For an implementable solution, the load on the operator should be reduced and more error checking and validation should be done algorithmically. This will move towards a more autonomous solution, with less reliance on the operator and improve the accuracy of readings.

6.2 3D Modelling

6.2.1 Overall

An effective implementation of 3D circumference modelling from the extremities and lengths determined, was not successfully achieved.

Below is a summary of the performance of the two models used in comparison with the quantitative aims:

- The average accuracy tolerance of the ellipse model was 106.82% - (Aim: $< 30\%$)

- The average accuracy tolerance of the rectangle model was 35.80% - (Aim: $< 30\%$)

It is clear that neither of the models performed within the aim of 30%. However, the rectangle model came very close with an average accuracy of 35.80%. This result was not expected as it was originally theorized that the ellipse model would yield better results.

It should also be noted however, that errors made during the extremity measurements, would propagate through the system and impact the circumference. Additionally, factors such as the presence of clothing would also greatly impact the result. Therefore, if these measurements were improved, this would also result in an improvement in the accuracy of the models.

Further analysis of each individual model is presented below:

6.2.2 Ellipse Model

The ellipse model performed much worse than expected. The accuracy values for the different limbs ranged from 60.06% to 148.70%.

This wide spread of accuracies and large deviation from the actual circumference measurements indicates that the ellipse is not a suitable shape to model the 3D circumference of body parts in this application.

Closer inspection of the measurements of the volunteers obtained together with empirical observations revealed that human limbs are often more elongated and less ovular.

Additionally, the ellipse perimeter is modelled using the length of its minor and major axes. The algorithm was built on the assumption that the 90° rotation in between each measurement would adequately provide measurements for the two axes. However, further analysis of the system revealed that the measurements obtained often did not correspond to the major and minor axes. Instead, they would correspond to diameters or chords slightly off either axis, due to factors such as the slight rotation of the arm during measurement. Frequently, this error in modelling was exacerbated by the algorithm because the minor axis was often given a value that was too big. As a result, the shape being modelled was more along the lines of a circle with a diameter of the longest part of the limb. This is the reason for the large circumference values.

6.2.3 Rectangle Model

The rectangle model performed much better than expected. The accuracy values for the different limbs ranged from 18.89% to 49.28%.

This spread of results and deviation from the actual circumference measurements was significantly better than the results of the ellipse method.

This model can better deal with the elongation of limbs. Additionally, the slight variations in orientation that greatly impacted the ellipse model, do not affect this model as much. This model evidently creates a much tighter circumference approximation than the ellipse method.

This method, however, still does not meet the requirements of have an accuracy of less than 30%. For an implementable solution, this accuracy would have to be further decreased to a value in the region of 5%. Therefore, there is still much work to be done in improving the accuracy of the modelling.

The positive takeaway is that this model can be used to create the maximum boundary or encapsulation of the circumference. With further data points or improved measurement techniques, this modelled can be adjusted down to more accurately fit the circumference (E.g. Add curvature to the corners etc.).

6.3 User Interface

6.3.1 Autonomy

As eluded to in previous sections, the system as it stands relies on the operator or user to a large extent. The system dependencies on the user are listed below:

- User must select the appropriate view measurement
- User must execute the measurement process
- User must error check each "View" measurement
- User must instruct person being measured how to orientate for optimum results

View Selection The selection of "Views" in the user interface is effective in that ensures only one view is chosen at a time. It also has a relevant status update when a new view is chosen. However, the reliance of the user to choose the correct "View" is large. This causes an area of operation that is prone to error as the user can often forget to choose a new view or the correct view. This should be made more autonomous with algorithms that automatically detect the correct view depending on the orientation of the body. If this is not implemented, at least more visible warnings should be shown to guide the user if he/she makes a mistake (E.g. Tries to take a new measurement when the orientation of the measured body has obviously changed).

Measurement Process The measurement process itself relies on the user significantly. All though all the background processing is handled and abstracted from the user, the user still needs to initiate the process, perform validation on the output measurements and advise the measured person how to adjust their body for better readings. This empirical correction and execution would require the use to be train and/or spend a great deal of time and effort during the measurement process. Additionally, it makes the system more prone to error as the user may miss small mistakes in the process. This is not ideal for the user or the person being measured and may cause frustration towards the system, resulting in a lack of adoption. The measurement process should be made far more autonomous before a solution is implementable. Suggestions for this are given in Chapter 9.

6.3.2 Aesthetics

This is the first version of a basic working solution and as such, is missing a few key design elements. Besides the error checking procedures that should be included, mentioned in the above section, the interface itself would have to be redesigned to optimise user ease-of-use, comfort and effectiveness. This would be accomplished by making the interface more intuitive with more obvious controls and more messages. A few required improvements are listed below:

- Improve the colour scheme of the interface to make the experience for the user more enjoyable
- Increase the size and/or improve the locations of buttons to make them more easily accessible

- Make the status bar more visible and/or use more pop-up windows
- Have different screens to guide the user through the process
- Make the data easily exportable (E.g. To a database or Excel spreadsheet)

Chapter 7

Discussion

Here is what the results mean and how they tie to existing literature...

Discuss the relevance of your results and how they fit into the theoretical work you described in your literature review.

Chapter 8

Conclusions

These are the conclusions from the investivation and how the investigation changes things in this field or contributes to current knowledge...

Draw suitable and intelligent conclusions from your results and subsequent discussion.

However, for the purposes of a first attempt at the project to get an overall understanding
- Macro understanding was fine

Chapter 9

Recommendations

1) Statistical Model in taking readings
2) Automatic system to remove the need for a person
3) Clothing model
4) AI for understanding body shape
5) Better modelling of 3D body parts
6) Extension to cellphone
7) Full 3D parameter modelling - Body parts, skin contours, body fat% etc.
8) Filtering skeleton joints
9) Filtering skeleton for better background removal
10) Increase sensitivity of algorithm to discard padding
11) Filter Depth Data - Shadow Removal
12) Use better resolution depth sensor
13) UI - Automatic View Selection, Error Messages for views, if single button - give messages to user on how to instruct person being measured (Or directly to person being measured), process multiple measurements - Stats rec above

Use the IEEE numbered reference style for referencing your work as shown in your thesis guidelines. Please remember that the majority of your referenced work should be from journal articles, technical reports and books not online sources such as Wikipedia.

Bibliography

[1] M. S. Tsoeu and M. Braae, “Control Systems,” *IEEE*, **vol. 34(3)**, pp. 123-129, 2011.

[2] J. C. Tapson, *Instrumentation*, UCT Press, Cape Town, 2010.

[3] A. Adikari, N. Ganegoda and W. Wanniarachchi, ”Non-Contact human body parameter measurement based on Kinect Sensor,” *IOSR Journal of Computer Engineering*, vol. 19, no. 3, pp. 80-85, 2017.

[4] M. Karbasi, Z. Bhatti, P. Nooralishahi, A. Shah and S. Mazloomnezhad. (2015). Real-Time hands detection in depth image by using distance with Kinect Camera. *International Journal of Internet of Things.* [Online]. Available: http://article.sapub.org/10.5923.c.ijit.201501.01.html

[5] F. Isikdogan and G. Kara, ”A real time virtual dressing room application using Kinect,” Bogazici University, Istanbul, 2012.

[6] S. Zennaro, M. Munaro, S. Milani, P. Zanuttigh, A. Bernardi, S. Ghidoni and E. Menegatti. (2015) Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications. *IEEE*, [Online]. Available: https://doi.org/10.1109/ICME.2015.7177380

[7] Microsoft Developer Network, ”Kinect for Windows Sensor Components and Specifications,” [Online]. Available: https://msdn.microsoft.com/en-us/library/jj131033.aspx. [Accessed 31 10 2017].

[8] Microsoft Developer Network, ”Depth Camera,” [Online]. Available: https://msdn.microsoft.com/en-us/library/hh438997.aspx. [Accessed 31 10 2017].

[9] Microsoft Developer Network, ”Coordinate Spaces,” [Online]. Available: https://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges. [Accessed 31 10 2017].

[10] Microsoft Developer Network, ”Skeletal Tracking,” [Online]. Available: https://msdn.microsoft.com/en-us/library/hh973074.aspx. [Accessed 31 10 2017].

[11] Microsoft Developer Network, "Tracking Users with Kinect Skeletal Tracking," [Online]. Available: https://msdn.microsoft.com/en-us/library/jj131025.aspx. [Accessed 31 10 2017].

[12] A. Kalaimaran, "Formula derived mathematically for computation of perimeter of ellipse," *International Journal of Physics and Mathematical Sciences*, vol. 2, no. 1, pp. 56-70, 2012.

[13] Department of Physics, "Guide to Reporting and Measurement", *University of Cape Town*, Cape Town, 2014.

Appendix A

Detailed Results of Volunteers

Add any information here that you would like to have in your project but is not necessary in the main text. Remember to refer to it in the main text. Separate your appendices based on what they are for example. Equation derivations in Appendix A and code in Appendix B etc.

Table A.1: Add caption

	Chest	Waist	Upper Left Arm	Lower Left Arm	Upper Right Arm	Lower Right Arm	Upper Left Leg	Lower Left Leg	Upper Right Leg	Lower Right Leg
Front	38.92	38.27	7.97	9.28	7.09	7.27	24.72	12.55	24.38	12.02
Front	38	31	11.5	9	12	7.5	15.5	10.5	15.5	10.5
Error	2.42%	23.45%	-30.70%	3.11%	-40.92%	-3.07%	59.48%	19.52%	57.29%	14.48%
Left	28.73	31.96	7.39	8.73	#N/A	#N/A	20.05	12.55	#N/A	#N/A
Left	20	20.5	11	7.5	#N/A	#N/A	14.5	11.5	#N/A	#N/A
Error	43.65%	55.90%	-32.82%	16.40%	#N/A	#N/A	38.28%	9.13%	#N/A	#N/A
Back	80.37	34.58	8.12	8.31	7.88	7.37	22.38	12.89	23.54	11.05
Back	35	30	11	7.5	11.5	9.5	15	11.5	14.5	11.5
Error	129.63%	15.27%	-26.18%	10.80%	-31.48%	-22.42%	49.20%	12.09%	62.34%	-3.91%
Right	25.68	29.42	#N/A	#N/A	9.8	8.78	#N/A	#N/A	17.38	24.58
Right	21	22	#N/A	#N/A	10.5	9.5	#N/A	#N/A	17	11.5
Error	22.29%	33.73%	#N/A	#N/A	-6.67%	-7.58%	#N/A	#N/A	2.24%	113.74%

Appendix B

Addenda

B.1 Ethics Forms