

University of Denver

Digital Commons @ DU

Electronic Theses and Dissertations

Graduate Studies

2020

Development of a Clinical Marker-less Motion Capture System for Patient Monitoring

Abigail Eustace
University of Denver

Follow this and additional works at: <https://digitalcommons.du.edu/etd>



Part of the [Biomechanics and Biotransport Commons](#), and the [Other Biomedical Engineering and Bioengineering Commons](#)

Recommended Citation

Eustace, Abigail, "Development of a Clinical Marker-less Motion Capture System for Patient Monitoring" (2020). *Electronic Theses and Dissertations*. 1760.
<https://digitalcommons.du.edu/etd/1760>

This Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu, dig-commons@du.edu.

Development of a Clinical Marker-less Motion Capture System for Patient Monitoring

Abstract

The ability to understand human movement is beneficial for deciding surgical procedures, tracking disease progression over time and helping with patient rehabilitation. The current gold-standard for collecting human movement is the use of 3-dimensional marker-based systems. Several studies have presented the many limitations to the current gold-standard that reduces the number of people who are able to benefit from a gait analysis. Those limitations in the current gold-standard include the requirements of large laboratory space, costly equipment, long instrumentation and collection time, and the potential for motion artifact from markers being placed on the skin. The purpose of this study is to create a marker-less motion capture system using the newly-released Kinect Azure cameras from Microsoft. The study aims to validate the new system against the gold-standard. A validation of a four Kinect Azure camera system was conducted with 10 subjects completing over ground walking trials at a self-selected pace, sit-to-stand, lunge, and step up/down while Kinect and 3D marker-based data were collected simultaneously. The data was synchronized and cut to a single activity cycle where joint angles and spatio-temporal measures were compared between the two systems. Walking speed and stride length were highly correlated between the two systems with r -values >0.9 and p -values <0.001 . The average difference in maximum knee flexion angle between the two systems is 2.84° with a $r=0.785$ and p -value <0.001 . A 3D point cloud was generated from the four Kinect Azure camera system to generate a surface mesh. The 3D mesh was used to provide a better understanding of body habitus than current BMI.

Document Type

Thesis

Degree Name

M. S.

Department

Mechanical Engineering

First Advisor

Paul Rullkoetter

Second Advisor

Chadd W. Clary

Third Advisor

Barry Zink

Keywords

Kinect, Marker-less, Motion capture, Validation

Subject Categories

Biomechanics and Biotransport | Biomedical Engineering and Bioengineering | Other Biomedical Engineering and Bioengineering

Publication Statement

Copyright is held by the author. User is responsible for all copyright compliance.

Development of a Clinical Marker-less Motion Capture System for Patient
Monitoring

A Thesis
Presented to
the Faculty of the Daniel Felix Ritchie School of Engineering and Computer
Science
University of Denver

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Abigail Eustace
August 2020
Advisor: Dr. Paul Rullkoetter

Author: Abigail Eustace
Title: Development of a Clinical Marker-less Motion Capture System for Patient Monitoring
Advisor: Dr. Paul Rullkoetter
Degree Date: August 2020

Abstract

The ability to understand human movement is beneficial for deciding surgical procedures, tracking disease progression over time and helping with patient rehabilitation. The current gold-standard for collecting human movement is the use of 3-dimensional marker-based systems. Several studies have presented the many limitations to the current gold-standard that reduces the number of people who are able to benefit from a gait analysis. Those limitations in the current gold-standard include the requirements of large laboratory space, costly equipment, long instrumentation and collection time, and the potential for motion artifact from markers being placed on the skin. The purpose of this study is to create a marker-less motion capture system using the newly-released Kinect Azure cameras from Microsoft. The study aims to validate the new system against the gold-standard. A validation of a four Kinect Azure camera system was conducted with 10 subjects completing over ground walking trials at a self-selected pace, sit-to-stand, lunge, and step up/down while Kinect and 3D marker-based data were collected simultaneously. The data was synchronized and cut to a single activity cycle where joint angles and spatio-temporal measures were compared between the two systems. Walking speed and stride length were highly correlated between the two systems with r -values >0.9 and p -values <0.001 . The average difference in maximum knee flexion angle between the two systems is 2.84° with a $r=0.785$ and p -value <0.001 . A 3D point cloud was

generated from the four Kinect Azure camera system to generate a surface mesh. The 3D mesh was used to provide a better understanding of body habitus than current BMI.

Acknowledgements

First and foremost, I would like to thank my advisor Dr. Paul Rullkoetter for all of his support and mentorship over the last two years. I would also like to thank Dr. Casey Myers for all of his mentorship in helping to successfully develop a framework for the collecting and processing all of data with the Kinect Azure cameras.

I would like to thank Dr. Chadd Clary and Dr. Barry Zink for serving on my defense committee.

I want to also thank all of the members of the Human Dynamics Lab as well as the Experimental Biomechanics Lab for participating in pilot Kinect data collections that helped me develop the framework needed to validate the Kinect devices. I would especially like to thank Brittany Marshall for all the time she spent in the lab walking as I determined the best setup for collecting with multiple Kinect cameras. Also I would like to thank Jeremy Keller and Eddy Rogers for helping me develop the necessary C++ code to run and process the data from the Kinect Azure cameras.

Table of Contents

1	Introduction	1
1.1	Introduction	1
1.2	Study Objectives.....	3
2	Background & Literature Review	4
2.1	Introduction	4
2.2	Background of Gait Analysis for Human Movement.....	4
2.3	Gold-standard & limitations.....	6
2.4	Alternative Methods of Marker-less Motion Capture	7
2.4.1	IMU	7
2.4.2	Fixed installation camera systems.....	11
2.5	Kinect	12
2.5.1	Method for Collecting Gait Data with Kinect.....	14
2.5.2	Current Validation Study of Microsoft Kinect.....	15
3	Materials & Methods	21
3.1	Introduction	21
3.2	Experimental Setup.....	21
3.2.1	Calibration	22
3.2.2	Initial Validation Study	32
3.3	Data Acquisition.....	38
3.4	Validation Study	41
3.5	Data Post-processing	41
3.5.1	Estimation of Gait Metrics.....	42
3.5.2	Musculoskeletal Mechanics.....	44
3.5.3	Subject Surface Geometry	47
3.5.4	Point Cloud Driven Kinematic Model.....	49
4	Results	52
4.1	Initial System Validation	52
4.2	Larger Validation Study	55
4.2.1	Spatio-temporal Measures Compared to Prior Literature.....	57
4.2.2	Kinematic Measures Compared to Prior Literature	57
4.3	3D Point Cloud & Surface Mesh	58
4.3.1	Kinematic Model.....	62
5	Discussion	64
5.1	Limitations	66
5.2	Future Work	69

A	Kinect Users Guide	77
A.1	Hardware requirements	77
A.2	Setting Up Multiple Kinects	77
A.3	Calibrating Kinects.....	79
A.4	Running the Kinects for Collection of Human Movement	83
B	Kinect Processing Guide	85
B.1	Introduction	85
B.2	Basics.....	85
B.3	Calibration.....	86
	B.3.1 Capture.....	87
	B.3.2 Alignment Using CloudCompare	89
B.4	Construct a Configuration File.....	92
B.5	Secondary Calibration (Optional).....	95
B.6	Step 4: Depth Data Processing	101

List of Figures

Figure 2.1	Braune and Fischer’s experimental setup with Geissler tubes attached to subject’s body	5
Figure 3.1	Study Workflow	22
Figure 3.2	Kinect Workflow	24
Figure 3.3	Calibration object setup	25
Figure 3.4	Selecting points in global reference frame (R pts) and subordinate reference frame (A pts) to be used for creating a transformation between the two reference frames.	26
Figure 3.5	Merged point cloud of the two coronal plane Kinect captures	28
Figure 3.6	ICP setup of aligning sagittal plane capture to merged point cloud. Red is the merged point cloud and yellow is the sagittal plane point cloud	29
Figure 3.7	Length of calibration object (left). Transformed point cloud of calibration object used to measure length (right)	31
Figure 3.8	Laser scanner point cloud of phantom shank aligned with Kinect Azure point cloud (blue dots) of phantom shank . . .	31
Figure 3.9	Altered Depth Data from Reflective 3D Motion Capture Markers	32
Figure 3.10	Microsoft Kinect Azure Body Tracking Joint Labels	34
Figure 3.11	Initial data collection setup with single Kinect Azure camera and Optotrak	35
Figure 3.12	Point Cloud of Subject with Smaller Markers (left). Point Cloud of Subject without Markers (right)	36
Figure 3.13	X represent key point locations when markers are present in the point cloud. O represent key point location when markers are not present in point cloud.	37
Figure 3.14	Kinect Azure documentation on coverage of depth fields of view	38
Figure 3.15	Bird-eye view of the Kinect Setup for the validation study (left). Kinect camera setup for validation study(right)	40
Figure 3.16	Microsoft Kinect Azure Body Tracking Joint Labels	43
Figure 3.17	Musculoskeletal model with Kinect markers (left) and Vicon markers (right).	46
Figure 3.18	Points isolated in 3D surface point cloud for calculation of thigh circumference	48
Figure 3.19	Initial alignment of the shank (magenta) and thigh (green) .	50

Figure 4.1	Representative knee flexion angle as a function of gait cycle measured using both systems showing ± 2 standard deviations around the mean at each point in the gait cycle for all subjects	53
Figure 4.2	Correlation plot comparing Optotrak and Kinect max knee flexion angle for each subject	54
Figure 4.3	Representative trial comparing stance time, stride length, and walking speed between the two systems	54
Figure 4.4	Representative subject knee angle for a gait cycle for Kinect and Vicon data	55
Figure 4.5	Representative subject stance time, stride length, and walking speed for Kinect and Vicon data	56
Figure 4.6	Asymmetric gait pattern (left) and symmetric gait pattern (right) as identified by Kinect Azure	56
Figure 4.7	Frontal (left), side (middle), and posterior (right) view of the result of the custom program yielding a 3D point cloud from four Kinect Azure cameras	58
Figure 4.8	3D point cloud subject walking at different time points . . .	59
Figure 4.9	3D surface mesh of subject walking at different time points .	60
Figure 4.10	3D surface mesh location of circumference measurement of pelvis (yellow), thigh (red), and shank (green)	61
Figure 4.11	ICP's attempt at alignment of the two point clouds. Red represents the Kinect target model pose and green represents the cylindrical model pose.	62
Figure 4.12	Final CPD result with both rigid and nonrigid CPD implemented (left). Final iteration of CPD rigid only (right). The red represents the Kinect target model pose and the blue represents the cylindrical model pose.	62
Figure A.1	Bird-eye view of the Kinect Setup for the validation study. .	78
Figure A.2	Bird-eye view of the Kinect Setup for the validation study. .	79
Figure A.3	Example of the command to change the working directory of the command window	80
Figure A.4	Example of the command line argument to run 2 Kinects with bash scripting for calibration	82
Figure A.5	Example of the command used to trigger a master Kinect camera	83
Figure A.6	Example of the command used to trigger a subordinate Kinect camera	83
Figure A.7	Example of the command line argument to run 2 Kinects with bash scripting	84

List of Tables

Table 2.1	Kinect V2 & Kinect Azure Specification Comparison	14
Table 3.1	Probed points for each body segment	33
Table 3.2	Difference between segment lengths between capture with & without markers	36
Table 4.1	Difference between measured and estimated segment circumference for pelvis, thigh and shank	61
Table 4.2	Effect of point cloud density on CPD results	63

Chapter 1

Introduction

1.1 Introduction

Gait analysis has been evolving since Aristotle's first theories on human movement and continued as new theories emerged along with advances in technology [1]. Gait analysis is a clinical technique used to assess a person's ability to walk and to help identify deficits which could cause detrimental consequences of energy expenditure and balance [2]. Due to its' ability to identify deficits, gait analysis can be used to help diagnose and treat a wide range of pathologies that result in gait abnormalities.

Currently, there are few diseases or gait pathologies which are treated using gait clinical analysis due to the limitations of the current gold-standard for collecting data [3][4]. The current gold-standard for gait analysis is the use a 3-dimensional motion capture camera system with reflective markers attached to boney landmarks to track how a person is moving through time and space. This method requires a large experimental laboratory space for capture volume and expensive equipment. In addition, experienced engineers and clinical personnel are required to collect and process the data after, long collection times needed to apply markers to a person that have the potential to alter their natural movement

[5][6][7][8][9]. The cost and complexity are the main reported limitations of this system which have led to a translational lack of gait analysis used in the clinic and reduced the number of people who are able to benefit from gait analysis. These systems are mainly found in research laboratories and select hospitals. If the limitations of the current system were addressed, it would increase number of people who could benefit from a gait analysis. There would be an increase in the number of diseases and injuries that would benefit from a gait analysis by allowing treatment to be aided by identifying deficits in human movement.

Neurological diseases, like Parkinson, ataxia, and dementia, could be detected earlier if a gait analysis were able to be collected in the clinic or in the field. A gait analysis could also help to track the progression of these diseases by being collected every time a patient visits the clinic [10]. Additionally, gait analysis would help in the diagnosis and progression of osteoarthritis (OA). Currently in the field of hip and knee arthroplasty functional scores are used widely to assess changes to a patient's disease progression. However this method is subjective and there is significant disparity between the patient's and doctor's evaluations. Measured obtained with a gait analysis could be integrated along with other clinical and instrumental data to help with treatment and diagnosis of OA [11]. A more compact system could be also used in sports to help decide when an athlete is ready to return to their sport after an ACL injury or concussion by determining if there are deficits in the athlete's movement [12]. It could also be used to help advance the performance of athletes by optimizing their movement. There is a need for advancement in gait analysis technology that does not require markers, has a small capture volume, is easy to use, and cost effective. The aim of

this study is to develop a marker-less motion capture system based on the use of multiple inexpensive depth cameras that could be used to quantify movement quality in a clinical environment.

1.2 Study Objectives

The research objectives of this thesis are:

1. Develop a synchronized marker-less system based using multiple depth cameras to be used in a clinic for movement analysis.
2. Validate the system for clinically relevant spatio-temporal and joint kinematic measures across dynamic activities of daily living.
3. Using synchronized output for multiple depth cameras, develop a 3-dimensional surface mesh for a person performing dynamic activities that can be used to better understand the relationship between body habitus and movement.

Chapter 2

Background & Literature Review

2.1 Introduction

The need for marker-less motion capture is underscored when examining the current gold-standard for collecting human movement along with the system's limitations. This literature review will 1) introduce the history of gait analysis; 2) present the current gold-standard of marker-based motion capture and the system's limitations; 3) present alternative methods for capturing human movement using inertial measurement units and a fixed-installation camera system; 4) the benefits of a marker-less motion capture system; and will conclude with 5) the current systems using Kinect cameras to monitor human movement.

2.2 Background of Gait Analysis for Human Movement

The first theories on human movement began with Aristotle observing how the head moves as a person progresses through a gait cycle. He understood the head would not move in a straight line but in a zig-zag pattern [1]. As the understanding of physics improved, people started to debate whether the movement of a person was due to gravity or forces within the muscles. As scientists started to gather evidence for one theory or the other it lead to

improvements in collecting human movement data. Eventually the first 3-dimensional gait analysis was performed in 1895 by Braune and Fischer [1]. The first gait analysis used cameras with a continuous exposure to collect images of the subject who had Geissler tubes attached to their body (Figure 2.1). The instrumentation time for this system took between 6 to 8 hours due to the bodysuit requirements needed with the Geissler tubes to ensure the electrical current would not come into contact with the subject's body [1].

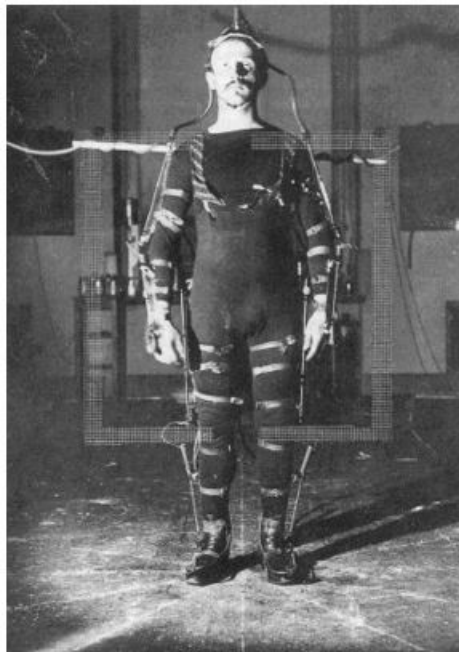


Figure 2.1: Braune and Fischer's experimental setup with Geissler tubes attached to subject's body

Injuries during the First World War lead to doctors wanting a better understanding of how to measure rehabilitation of those injured during the war. The Second World War then lead to the greatest advancement in gait analysis in the United States when Inman and Eberhart founded the first biomechanics lab at the University of California around 1945 [1]. The initial setbacks of a gait analysis collected in the lab at the University of California was the time required to

manually process the data. As computers advanced and processing time was reduced through automation, the availability of a gait analysis increased. While technology has advanced since Braune and Fischer first conducted gait analysis, the general system and processing remain the same. Currently, multiple infrared cameras are used to capture the location in space of reflective markers attached to boney landmarks on a person similar to how the Geissler tubes were used by Braune and Fischer.

2.3 Gold-standard & limitations

The gold-standard for collecting a gait analysis is using a 3D marker-based motion capture system [3][4]. The system uses infrared cameras and reflective markers placed over the whole body at anatomical landmarks to determine a reconstructed skeleton of the subject that can be used to calculate kinematic and spatio-temporal measures. In addition to cameras, it is common to use force plates to determine kinetic quantities. The high-quality cameras and force plates needed to have a lab where a gait analysis is performed with the current system is expensive. There is also a long preparation phase due to the time required to accurately place the markers on a person, along with the number of markers required to track rigid bones. These are both limitations cited by a number of researchers of the current system. Studies have additionally cited requiring skilled technicians to run and process the data, a large capture volume, soft tissue artifact, and markers influencing normal human movement as limitations to traditional marker-based motion capture [5][6][7][8][9]. A majority of these limitations reduce the number of eligible recipients of a gait analysis, as there are few labs with this system due to the equipment and personnel required. However, there has been an increase in the demand for analyzing human

movement to help in the surgical decision process, in diagnosing and monitoring different diseases like Parkinson's [2], with rehabilitation either in a clinic or at home, and to better understand injury and recovery in sports. Due to this increase in demand, and the limitations of the current method for collecting human movement not being able to accommodate the demands, researchers have been trying to determine an alternative method to capture human movement.

2.4 Alternative Methods of Marker-less Motion Capture

2.4.1 IMU

An alternative method to 3D marker-based motion capture to collect human movement is with wearable inertial measurement units (IMUs). IMUs are devices with embedded 3D accelerometers, gyroscopes, and magnetometers that calculate angular velocity and linear acceleration. When the sensors are calibrated on even surfaces with right angles, the axes of the sensor's local coordinate system represent the orthogonal base typically aligned with outer casing of the sensor. Combining the multiple signals from the IMU there are different fusion and filtering algorithms that can provide estimates of a sensor's orientation with respect to a global fixed coordinate system [13]. This orientation from multiple sensors placed along limb segments can be used to calculate measures of human movement. One limitation of IMUs is the presence of magnetic disturbances, which can limit the accuracy of the orientation estimates [13]. IMUs are attached to rigid body segments for tracking the location of the thigh, shank, and foot usually for lower body studies. The leg is a challenging spot to attach a sensor as it is not an even surface with right angles. Therefore, the local coordinate axes do not coincide with the joint axis. A transformation is required to align the axes in a

joint-related coordinate system. Calculating kinematics from IMUs these methods are only accurate for the primary rotation DOF, and the 2nd and 3rd rotations get progressively worse in terms of accuracy. However, even with these device limitations, researchers have still been exploring using IMUs to capture human movement because they have been able to address some of the current limitations of 3D marker-based motion capture. The sensors are small, allowing people to wear them in special clothing or IMUs can be attached to the body with tape or elastic straps. IMU sensors are low-cost and easy to use [13][14][15] [16]. IMU sensors can be used as a multi-sensor platform or as a single IMU to collect movement data. IMUs are wireless and are not confined to a laboratory as they can be used outside or in a clinic. IMUs are a system that could be used to assist in the rehabilitation process as patients can use them in their own home. Studies have evaluated the use of IMUs in detecting gait phases, measuring joint angle, and estimating stride length.

Spatio-temporal parameters consisting of stride time, stride length, cadence, and step length are measures used in the clinic to assess how well a person is moving. These are simple metrics that offer clinical value for diagnosis and monitoring disease progression by comparing them to patient normals. The collection of spatio-temporal parameters does not required an expensive gait analysis. Clinician's use simple tools like tape measures and stop watches to collect spatio-temporal parameters, but using a system like IMUs to collect these parameters would improve inter-rater reliability. Several studies have attempted to validated IMUs to both the gold-standard system along with the current clinical methods [16]. IMUs showed excellent validity when measuring speed, gait cycle time, stride length, and cadence when compared to methods used by

clinicians. Another study found the mean difference between IMU and an optical system for stride time, stride length, cadence, and step length to be 0.02 (sec), -0.04 (m), -2.92 (step/min), and -0.02 (m) respectively [17]. All of spatio-temporal parameters of interest were within 95% limits of agreement. There is no significant difference between the current gold-standard and IMUs when determining spatio-temporal measures. IMUs also showed excellent validity compared to current methods used in the clinic to determine spatio-temporal parameters [15][17].

Joint angles from IMUs are calculated by placing an IMU on either side of the joint of interest. Most work by researchers has focused on determining the knee joint angle and validating it against the current gold-standard, but the ankle and hip angles have also been compared. A majority of published work considers only the flexion and extension angle at the knee reducing the degrees of freedom at the knee to one. In order to validate IMUs as a method for a gait analysis, joint angles from both systems are simultaneously recorded. The study design consists of placing an IMU on the trunk, right and left thigh, shank, and foot. In most studies, the reflective markers for the optical system are placed on the IMUs. There are a few studies in which the reflective markers are placed on anatomical locations instead of the IMU. When validating between the two systems, it is important to understand what is being compared. If markers are placed on the IMUs, the study compares the accuracy of the optical system to the inertial system, versus comparing the results of an optical gait analysis to a inertial gait analysis [13]. Studies that placed markers on anatomical locations reported higher root-mean-squared error (RMSE) values for joint angles. Studies with markers placed on IMU sensors report RMSE values around 4° and studies with anatomical markers report

RMSE values around 8° [13]. The accuracy of the hip and ankle angle are also of interest. Watanabe et al. [15] reported, for all walking speeds, the ankle angle had an RMSE value smaller than 4° . The accuracy of the hip joint angle has been reported to be greater than that of both the knee and ankle. The hip joint angle has the same trajectory between the two systems but the magnitude is different [17].

When patients use IMUs at home for rehabilitation there is potential for the sensor to be incorrectly placed on the body. Leardini et al. [14] investigated how the malposition of an IMU affected joint angle calculations. In this study, the IMU sensor was malpositioned in the frontal plane by $\pm 15^\circ$ and in the medio-lateral direction by ± 7 cm. The sensitivity of the system to malpositioning of the sensors has a RMSD value less than 4° in either plane of alignment. The medio-lateral direction was the most sensitive to the malposition with a RMSD value of $3.1 \pm 1.8^\circ$. Seel et al. [13] validated the IMU sensors for use in a gait analysis with a unilateral lower limb amputee and noticed the agreement between the two systems was better on the prosthetic side than the unaffected side. From this, researchers concluded that there is still skin and soft tissue artifact with the IMU sensors being attached to body segments [17].

IMUs address many of the limitations present with the current gold-standard for collecting data on human movement. IMUs are low-cost, easy to use, and can be used anywhere. There is no significant difference between IMUs and the current gold-standard for measuring spatio-temporal parameters [15][16][17]. Additionally the accuracy of measuring joint angles is around 4° . IMUs still have the limitation of requiring a subject or patient to have sensors attached to their body.

2.4.2 Fixed installation camera systems

Another low-cost and portable system some are using for a gait analysis is a fixed camera system use RGB video cameras. There is increased variability in fixed cameras systems in relation to the setup and processing of the data. Fixed RGB video camera system use a variable number of cameras to collect the data. Yang et al. [18] used a single camera to determine kinematic values at the knee. The thigh and shank were tracked in time with black and white targets attached at the joints. The location of the markers were determined using a search area constraint and an image quality assessment algorithm from an image formation point of view. Only the knee angle was compared between the fixed camera system and the gold-standard with a max error of 5° and 9° respectively for normal and slow walking conditions. Verlekar et al. [19] used a two-camera system to classify gait impairments. It was an appearance-based system that did not rely on markers. They had successful acquisition and evaluation of an individuals' gait. Krishnan et al. [20] used a webcam and a custom algorithm processed in Labview to track real-time human movement. They cited a need for a quantitative, low-cost system for a gait analysis to eliminate the limitations of a qualitative gait analysis. Qualitative gait analyzes have minimal instrumentation and are inexpensive but they require a considerable amount of training and practice to be effective and even then the outcome is still subjective and lacks reliability [20]. Krishnan et al. [20] validated their marker-less system to the Lokomat® system. Lokomat® is a robotic rehabilitation device that provides highly repetitive and the most physiological movement training [21]. This device is used to retrain a person's gait and can be set to a specific angle or given a trajectory to match. The kinematic data between the webcam and those of the

potentiometers on the Lokomat® were similar. There was a mean absolute error of less than 2° for all speeds tested. The measures were also repeatable, as data was collected on two separate days and the mean difference between the two days was $1.56 \pm 0.13^\circ$ and $2.56 \pm 0.53^\circ$ for the hip and knee, respectively. The number of strides and the stride duration was the same between the two systems. This method of measuring real-time human movement with a low-cost webcam is reliable and accurate. The limitation of the system is the study was only in a single plane and therefore any movement out of the plane would cause errors in estimated joint angles. The system also cannot handle occlusions. Fixed installation camera systems are low cost and are easy to use but some setups still require markers to track body segments.

2.5 Kinect

The complexity of acquiring accurate 3-dimensional kinematics has limited the methods for using marker-less motion capture [6]. Technology has advanced in the recent years creating better options for marker-less motion capture. A depth-based camera has been researched as a potential method of marker-less motion capture. One depth based camera many researchers are considering is the Kinect from Microsoft. Microsoft developed the Kinect for gaming purposes but due to the inexpensive nature of the device and the combination of a depth camera and a RGB camera, researchers have been interested in investigating it as method for marker-less motion capture. Microsoft has released four different versions of the Kinect. The original version was the Kinect Xbox 360, released in 2010. It uses an infrared (IR) laser projector combined with monochrome CMOS sensor to capture 3D data [22]. In 2013 the next version of the Kinect was released, the Kinect for Xbox One, with a new

method of capturing depth data with a time-of-flight (TOF) camera. The TOF increased the accuracy of the depth data and can track visible light using an active IR sensor. A year later in 2014 the Kinect 2 for Windows was released. This version of the hardware was similar to the Kinect Xbox One, but also included a Windows Software Development Kit (SDK) platform to encourage wide use and development across applications [23]. The Kinect 2 for Window is called Kinect V2 in many research studies and the Kinect Xbox 360 is called Kinect V1. In 2019 the latest version of the Kinect was released, Kinect Azure, with the intent for the sensor to be used in research and industry and no longer compatible with the Xbox. The TOF sensor has been updated and uses a HoloLens 2 [24]. There are many improvements made to the new Kinect Azure. The Kinect Azure has a higher resolution RGB camera than the Kinect V2. The greatest improvement of the Kinect Azure is the multiple resolutions available for collection of depth data. The Kinect Azure has three depth resolutions at two different collection rates. Additionally, the Kinect Azure has the capability to sync multiple Kinect Azure devices together. The Kinect V2 does not have the capability to sync devices together without external hardware. Finally, the Kinect Azure has a seven microphone circular array compared to the four microphone linear phased array of the Kinect V2. A summary of the specification differences between the Kinect V2 and the Kinect Azure can found in Table 2.1

Table 2.1: Kinect V2 & Kinect Azure Specification Comparison

	Kinect V2	Kinect Azure
RGB Camera	1920 x 1080 px @ 30 fps	3840 x 2160 px @ 30 fps
Depth Camera	512 x 424 px @ 30 fps	640 x 576 px @ 30 fps 512 x 512 px @ 30 fps 1024 x 1024 px @ 15 fps
Synchronization	RGB & Depth internal only	RGB & Depth internal, external device-to-device
Audio	4-mic linear phased array	7-mic Circular array

Studies have explored validating the Kinect V1 and the Kinect V2 against the current gold-standard for both spatio-temporal parameters, kinematics, and kinetics. In the SDKs released with both the Kinect V1 and Kinect V2 there is a multi-segment skeleton that uses the depth data from the Kinect to determine joint locations. Most studies use these SDKs to determine joint locations to calculate spatio-temporal parameters and kinematics.

2.5.1 Method for Collecting Gait Data with Kinect

Studies have used both the Kinect V1 and the Kinect V2 for collecting human movement to validate with the current gold-standard. A systemic review by Springer et al. [25] examined the findings of 12 papers using the Kinect for gait assessment and found 10 of the 12 papers used the Kinect V1 and 2 papers used the Kinect V2. All but one paper used a single Kinect to collect data for the gait assessment. Of those 11 papers, 10 had the Kinect placed in front of the subject at a distance varying between 2 and 8 meters from the subject when the subject was walking on the ground. Generally, when a treadmill was used in the study, the Kinect was placed 1 meter in front of the treadmill. In one study, the Kinect was

placed to the subject's left and at a 45° angle from the treadmill. In another study, the Kinect was placed in front of the subject but at an off-centered position. From this review by Springer et al. [25] most studies use a single Kinect since the Kinect V1 and Kinect V2 are not designed to be synchronized together with multiple sensors. There are studies with multiple Kinect sensors capturing data, but external syncs and additional hardware is needed. The Kinect sampling rate was 30 Hz for all the studies reviewed. Most of the studies used the Microsoft Kinect skeleton tracking algorithm for calculating gait metrics, however two studies used the raw depth data and one study used the RGB data.

2.5.2 Current Validation Study of Microsoft Kinect

Studies have attempted to validate Kinect V1 and Kinect V2 for spatio-temporal parameters and kinematic measures with the current gold-standard. Many studies have validated the Kinect V1 and V2 to be used for determining spatio-temporal parameters. Clark et al. [26] found excellent agreement for gait speed, step length, and stride length compared to 3D marker-based gait analysis with r-values of 0.95, 0.99, and 0.99 respectively and p-values all <0.001. Behrens et al. [27] found only moderate agreement for gait speed between the Kinect and a clinical measured timed walking test. Vernon et al. [28] showed excellent agreement between a stopwatch and Kinect in time-up-go (TUG) test. Geerse et al. [29] found good to excellent agreement for gait speed, cadence, step length, stride length, step time, and stride time with ICC values of 0.995, 0.974, 0.994, 0.999, 0.888, and 0.962 respectively. Step width, step

length, and stride length are parameters that show excellent relative agreement. These studies showed there is good potential for the Kinect to be used to determine spatio-temporal parameters.

The systemic review showed a larger variation in the results for kinematic measures between the Kinect and current gold-standard. Studies showed some kinematic measures with excellent agreement while some kinematic measures presented low agreement and large error. The agreement between the two methods is not consistent enough for the Kinect to be used for a clinical assessment. When validating kinematic measures, Skals et al. [5] found the strongest correlation in knee flexion/extension angle (0.81), hip flexion/extension angle (0.82), and hip abduction/adduction angle (0.81). There was also strong correlation for the ankle (0.80), knee (0.78), and hip (0.71) resultant joint reaction forces between the two systems. Pfister et al. [30] found the Kinect peak angular displacements were smaller than those measured by the gold-standard ($p < 0.004$) in every case but the right hip. Additionally, they found the Kinect peak extension measurements to be consistently greater than that of gold-standard. Anderson et al. [31] found the hip joint angles to have a similar pattern between the two systems, but the Kinect predicts larger peak values. They also found the knee joint angle measured by the Kinect to display larger knee flexion during the first 40% of the gait cycle and slightly delayed peak knee flexion during swing.

In order to increase the accuracy of tracking human movement with the Kinect, several studies have explored the effects of using multiple Kinects to collect data. Studies have used between two and six Kinects to capture a subject walking. A study by Müller et al. [4] used six Kinects to collect joint position data.

They used three Kinects on each side of the walkway 2.5 meters from each other to create a tracking volume 9 meters in length. Müller et al. [4] found that having two-sided data from the Kinect increases the accuracy of spatio-temporal measures. They determined the skeleton fitting algorithm of the Kinect depends heavily on the angle of view from which the camera tracks the body. Therefore, when the Kinect was located only on the right or left side of the subject there were self-occlusions that disrupted the skeleton fitting algorithm. The joint positions are biased towards the surface area that is visible to the respective sensor (i.e. right side or left side) [4]. Another study used two Kinect sensors to track a subject walking and linked the sensors using iPi motion capture software [5]. They looked at validating kinematic measures compared to the gold-standard and found promising results, however these results were not highly correlated like the spatio-temporal measures in the Müller et al. [4] study.

From the studies conducted in validating the Kinect against the gold-standard, there have been some errors identified with the Kinect body tracking SDK. Researchers stated that, if these errors were addressed, it could help to improve the validation results. Using the Kinect to collect data around the ankle has been difficult for researchers. Skals et al. [5] provided some of the errors that currently exist with the Kinect at the ankle. Tracking the ankle could be compromised because of the light reflected on the ground during data collection. Light sensitivity has been proposed as a limitation of the Kinect. Additionally, tracking errors are considerably larger near the edges of the Kinect sensor's field of view and therefore could cause poor tracking of the feet. The final error Skals et

al. [5] proposed is due to the feet being in contact with the ground during stance phase of gait where clear distinction between the feet and ground does not exist in the depth map. There may be errors when fitting the stick figure to the depth data.

Researchers have also explored methods for predicting or estimating ground reaction forces (GRF) without the need for force plates to further extend the utility of gait analysis outside of the laboratory. Embedded force plates are common in the current gold-standard system because they yield the GRF data required to calculate kinetic quantities associated with movement. Kinetic quantities can be the primary determinants of understanding what is causing gait dysfunction in some diseases or conditions. For example, with Parkinson's Disease, understanding changes in a person's kinetics can help determine the reason for an unexplained fall [7]. Force plates are expensive, and for the best results they need to be embedded in the floor, making them an impractical device for a clinical setting. Therefore, researchers have been trying to predict GRF data from a person's kinematics. Ren et al. [32] has been successful in predicting GRF data during the single support phase of gait but GRF data cannot be determined during the double support phase of gait as the problem is closed looped [33]. The double support phase of gait is valuable for the diagnosing of disease and assessment of gait. Most attempts at estimating GRF data during double support phase are limited by the assumptions made about human movement. For example, one method assumed that the right and left limb were symmetrical but this is not the case for many people, especially with some diseases. Oh et al. [33] used a neural network model to predict GRF during double support phase. Their results for GRF data during double support phase matched magnitude and

overall patterns compared to measured values. All the GRF planes had a correlation coefficient greater than 0.8. They used methods similar to Ren et al. [32] for determining GRF during the single support phase and combined the two phase together with a spline. The combined GRF data for all planes had correlation coefficients greater than 0.84 [33]. Using a neural network to predict GRF data had significant improvements compared to previous methods and models.

Using Kinect data to drive musculoskeletal models is another potential method to estimate GRF data. This method uses artificial muscle-like actuators at contact points under each foot to calculate GRF data. This leads to a muscle recruitment problem which must be solved. Eltoukhy et al. [7] and Skals et al. [5] both used AnyBody modeling software with the GaitFullBody model which was scaled to a subject's specific size to estimate GRF from the skeletal joint locations from the Kinect V2. Eltoukhy et al. [7] solved the muscle recruitment problem during the double support phase using the Damsgaard et al. [34] method. Skals et al. [5] used polynomial optimization to solve the muscle recruitment problem allowing the muscles to only pull. Strong correlations were found between the gold-standard and the estimated GRF for the vertical GRF. However, peak values were significantly different and the time of occurrence in the gait cycle deviated slightly between the two system for the vertical GRF [5]. Eltoukhy et al. [7] found the calculated propulsive vertical force, braking horizontal force, and propulsive horizontal force to be significantly greater than that of measured values. The GRF_{ML} and GRF_{AP} agreement ranged from good to excellent between the two systems. Kinect driven musculoskeletal models showed promise for estimation of

unilateral GRF components but are highly dependent on the component of force being evaluated. Using more Kinect cameras could improve the tracking accuracy and therefore the accuracy of the estimated GRF [7].

Research has shown the Kinect to address the limitations of the current gold-standard by eliminating the need for markers, reducing the capture volume, and being cost effective. The Kinect V1 and V2 have been validated for use in determining spatio-temporal measures but not reliable for kinematic measures [26][29][30]. There is still more research needed to be able to create a marker-less motion capture system with the Kinect that can be used in a clinic for understanding a person's kinematic and kinetic quantities.

Chapter 3

Materials & Methods

3.1 Introduction

The methods for this study consisted of creating a configuration of Kinect Azure cameras, calibrating the Kinect Azure cameras to a global coordinate system, and collecting human movement to be validated against the current gold-standard. Additionally, surface model data was generated to allow for information about a person's body habitus.

3.2 Experimental Setup

An overview of the workflow for obtaining relevant data from the Kinect Azure cameras can be seen in Figure 3.2. The following sections explain each step of the workflow in more detail.

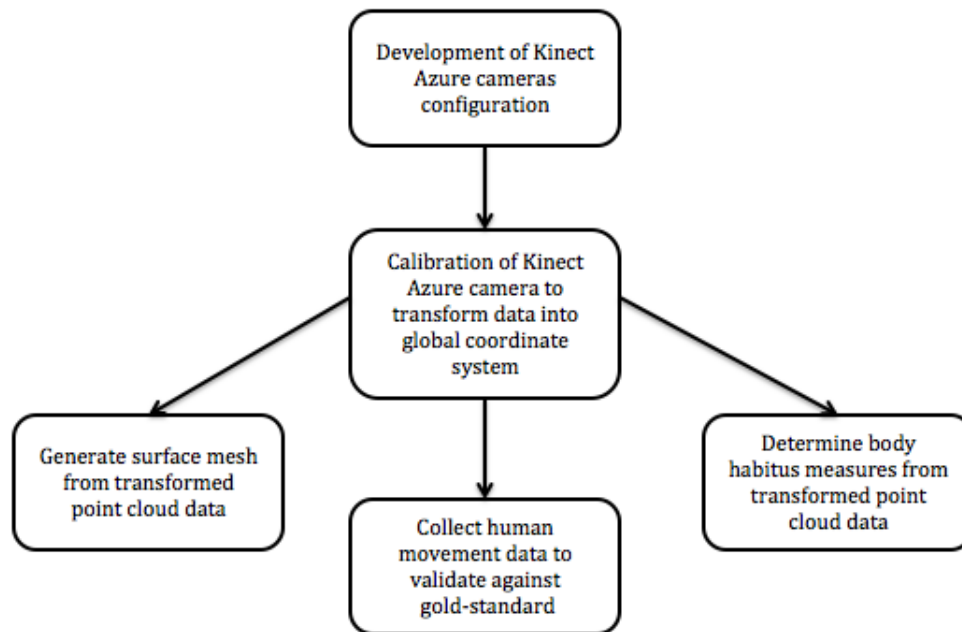


Figure 3.1: Study Workflow

3.2.1 Calibration

When collecting data with Kinect cameras that were not initially designed to be used for these purposes, it is important to have an accurate calibration aligning the output of all the cameras together in a global space. The Kinect Azure cameras can be time synced together with a pulse sent from the master camera to all of the other cameras connected in either a daisy-chain or star configuration. This study used a daisy-chain configuration to sync multiple cameras. A custom two-phase calibration process was designed to calibrate the Kinect Azure cameras. The first phase aligned the cameras close to each other in a global space using a point pair alignment and the second phase used iterative closest point (ICP) to refine and optimize the alignment. Three square tiles were

placed vertically spread throughout capture volume with the corners visible to all the cameras at varying heights and locations in the capture volume, as seen in Figure 3.3.

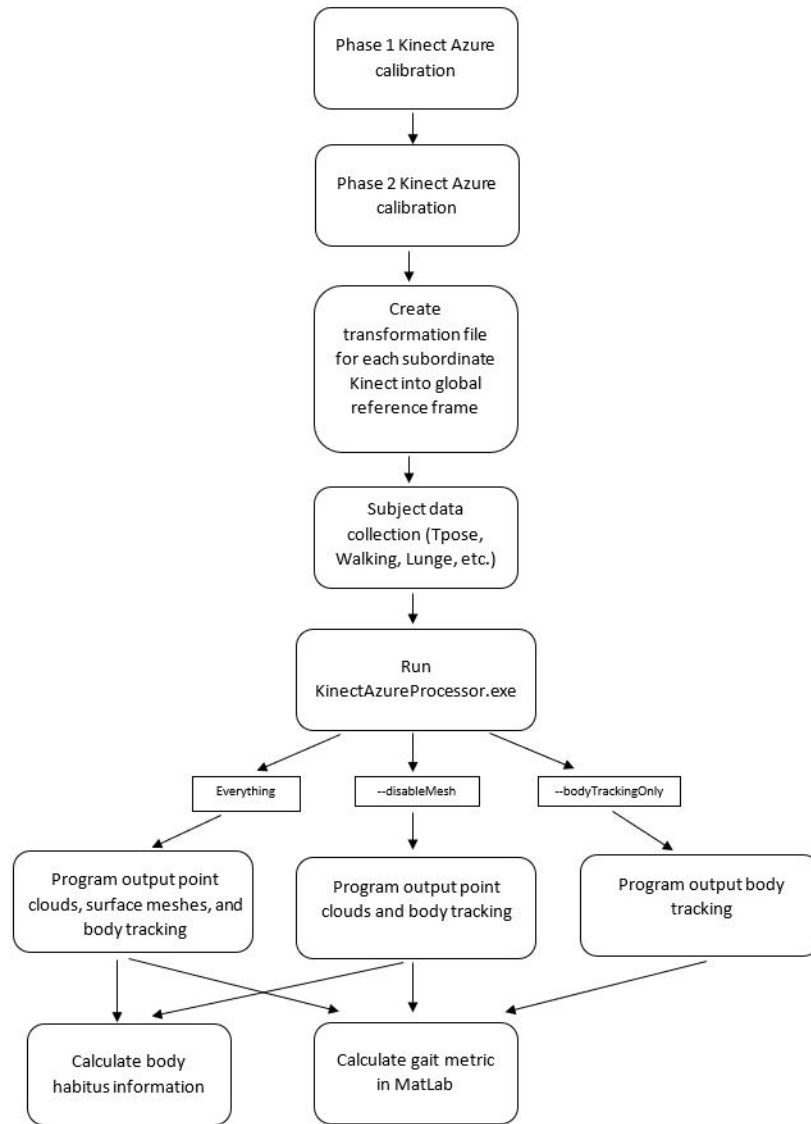


Figure 3.2: Kinect Workflow



Figure 3.3: Calibration object setup

The corresponding corners are selected in the master camera (global frame) and in each subordinate camera local reference frame. The 12 corner points are used to create a transformation for each subordinate camera to the global reference frame (Figure 3.4). This transformation is the first phase of the calibration and aligns the two reference frames with each other.

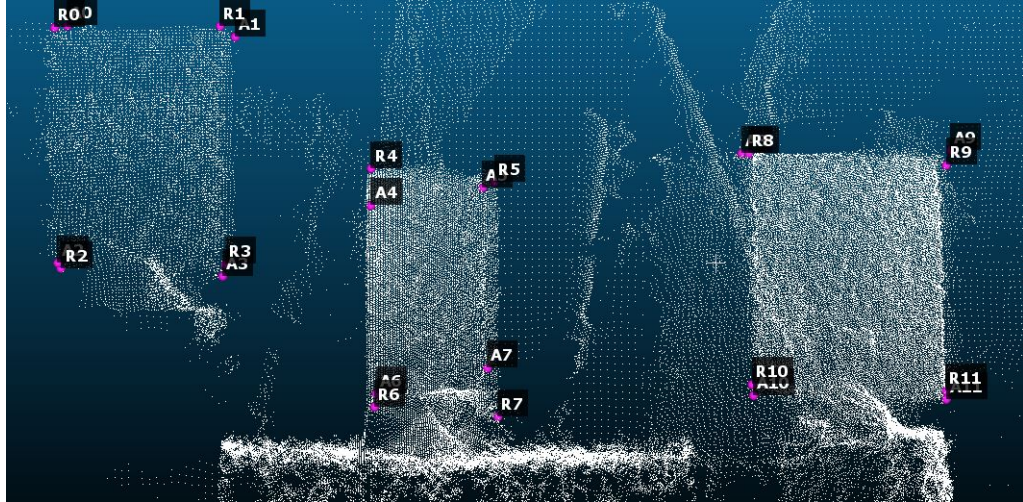


Figure 3.4: Selecting points in global reference frame (R pts) and subordinate reference frame (A pts) to be used for creating a transformation between the two reference frames.

For the second phase of the calibration, the initial transformation was applied to the point clouds of a subject in a static t-pose. The point cloud of the front and back of the subject were merged from the first phase transformation. Then ICP was applied to each of the two point clouds of the side of the subject to finish aligning the point clouds together. Each ICP generates another transformation for each of the subordinate cameras, perpendicular to the direction the subject is walking, into the global reference frame. The second phase transformation was combined with the first phase transformation to create a global transformation that is applied to all trials collected with the cameras in the current configuration. The second phase calibration is done only once per camera configuration and not completed for each subject collected with that camera configuration. If the camera(s) are bumped or moved then the two-phase calibration is repeated for the remain trials. The first phase of this calibration yields reasonable results of an error of $<3\text{mm}$ but the second phase improves the accuracy due to some of the properties of TOF cameras with an error of $<1\text{mm}$. It

has been reported that a TOF camera is sensitive to the color and material of an object when determining depth. For most colors and materials, the farther away from the camera an object is, the error in the reported depth data is increased [35]. The distance the calibration objects were away from each camera could account for some of the error due to this effects of TOF. It has also been reported that the edges of an object in a TOF frame are the nosiest and a place where the greatest distortion occurs [35]. This demonstrates the need to place the calibration tiles throughout the capture volume to generate the most robust transformation possible. The points being used to determine the transformation from each subordinate camera to the global frame were at the edges of the calibration object and therefore the point cloud accuracy of the object is reduced with added noise, making selecting accurate points challenging. Both the color, material, and the point of interest being at the edge of the calibration object could account for the RMS errors currently seen when attempting to align the 12 points in each subordinate camera reference frame relative to the master or global reference frame. Therefore, the second phase is need to overcome the known issues with the TOF and improve the accuracy of the calibration. The second phase of the calibration improves anatomical accuracy for the purpose of collecting human movement by merging the coronal plane point clouds (i.e. front and back of the body, Figure 3.5) and applies an ICP to align the merged point cloud with each of the point clouds from the sagittal plane (i.e. side of the body) one at a time as in Figure 3.6.

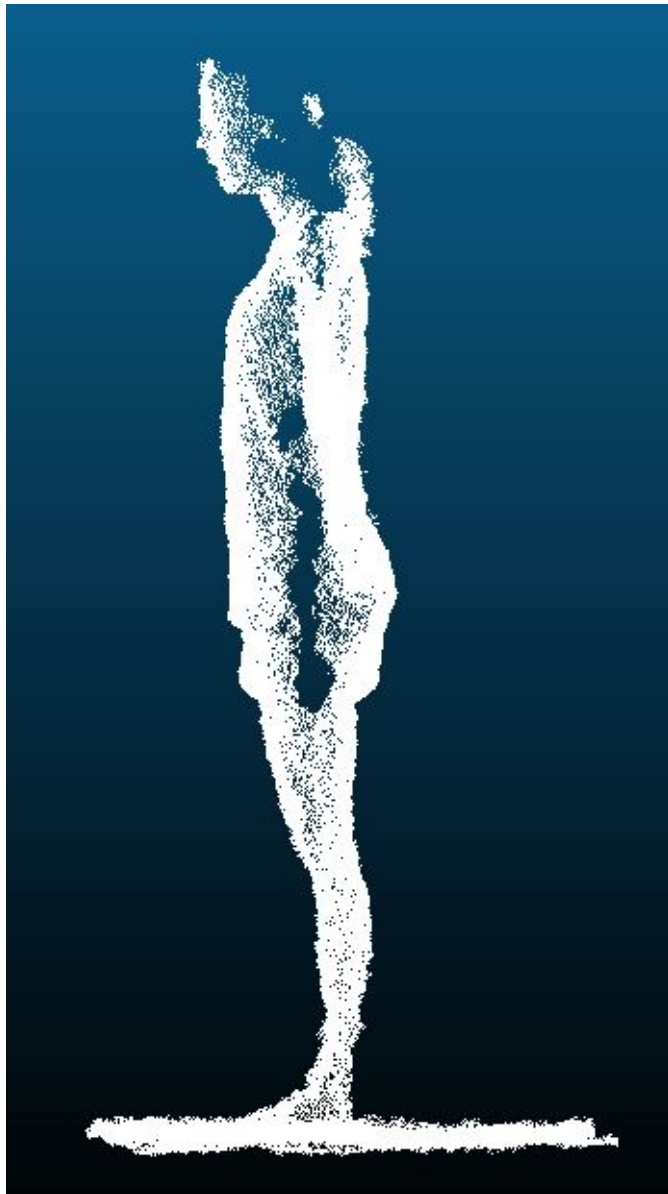


Figure 3.5: Merged point cloud of the two coronal plane Kinect captures

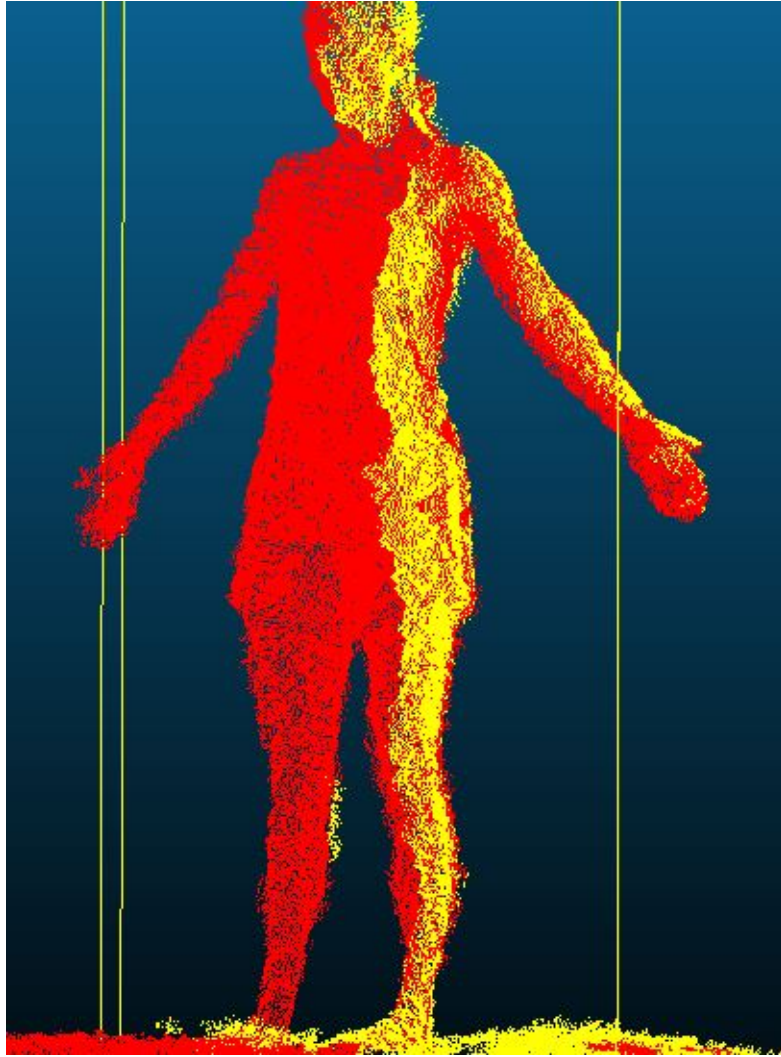


Figure 3.6: ICP setup of aligning sagittal plane capture to merged point cloud. Red is the merged point cloud and yellow is the sagittal plane point cloud

Once the ICP is complete, there is a new transformation for each Kinect Azure camera relative to the global frame. The second phase transformation is multiplied by the first phase transformation to calculate the overall transformation from the subordinate camera to the master camera equation 3.1 .

$$T_{Final} = T_{SecondPhase} * T_{FirstPhase} \quad (3.1)$$

The accuracy of the calibration process was determined by calculating the error between the known length of one of the calibration objects (Figure 3.7 left) and the corresponding length determined in the point cloud (Figure 3.7 right). The percent error between the actual length and the length after the point clouds are transformed into one global coordinate system is 0.396%. The error of only doing the first phase of the calibration is 1.04%. By adding the ICP for the side cameras, the accuracy of the calibration is improved and there is a reduction in the noise of the point clouds. This error after the two phase calibration is within the acceptable tolerance for our system as the Kinect depth data is measured in mm. This calibration process is used every time the orientation of the Kinect Azure cameras changes as well as each time a Kinect Azure camera is bumped.

The accuracy of depth measurement from the Kinect Azure was determine by comparing the points clouds of phantom of a shank with a high resolution laser scanner to the transformed Kinect Azure point cloud. Figure 3.8 showed the laser scanner point cloud aligned with the transformed point cloud from the Kinect Azure. The root-mean-squared error between the laser scan and the Kinect Azure point cloud is 4.2 mm.

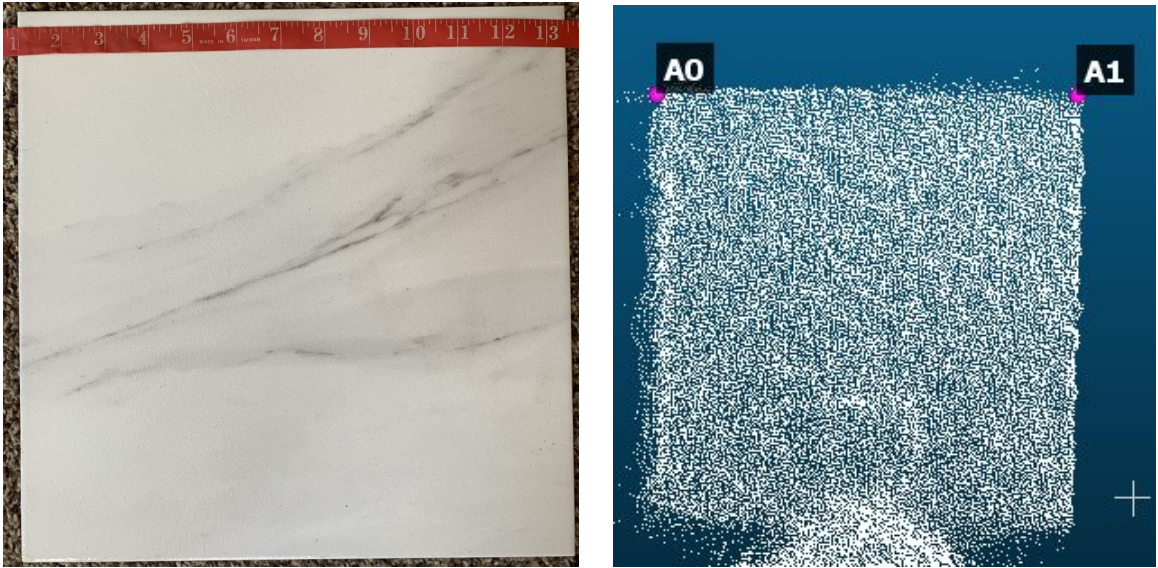


Figure 3.7: Length of calibration object (left). Transformed point cloud of calibration object used to measure length (right)



Figure 3.8: Laser scanner point cloud of phantom shank aligned with Kinect Azure point cloud (blue dots) of phantom shank

3.2.2 Initial Validation Study

An initial validation study was performed when the Kinect Azure was first released to determine if kinematic quantities could be measured with greater accuracy than the previous version. If the results of the small scale study showed promise, then a larger scale study would be conducted to validate the Kinect Azure as a method for marker-less motion capture instead of the gold-standard. Due to interference of reflective markers on the depth data as seen in Figure 3.9 an active marker system was used as the gold-standard.

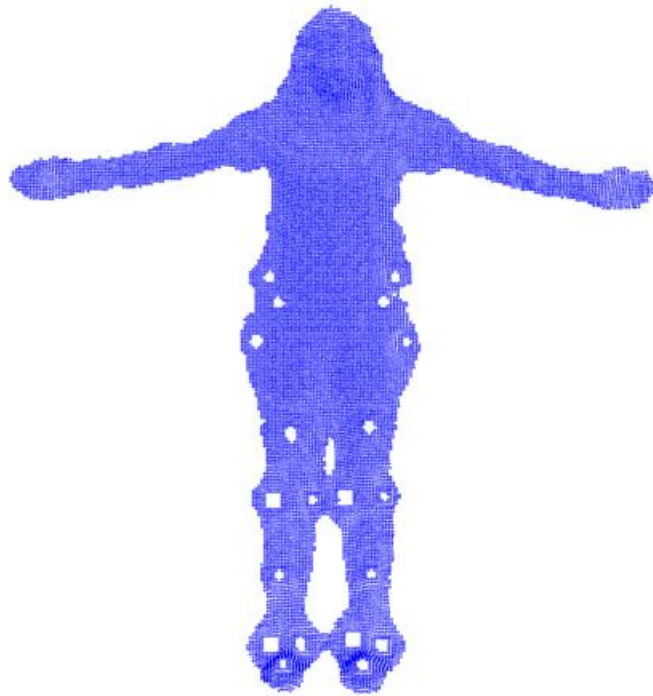


Figure 3.9: Altered Depth Data from Reflective 3D Motion Capture Markers

10 subjects performed a total of 10 over-ground barefoot walking trials and human movement data was collected simultaneously using a single Kinect Azure (30Hz) and Optotrak (100Hz) active kinematic tracking system. The subjects wore everyday clothes as a clinical measurement, and had three rigid bodies attached on the right thigh, shank, and foot. The experimental setup can be seen in Figure 3.11. Landmark points were collected on the pelvis, femur, tibia, and foot to create coordinate systems at the hip, knee, and ankle joints according to the International Society of Biomechanics (ISB) standard for calculation of relative joint angles. Table 3.1 shows the probed points for each of the rigid bodies used to create coordinate systems. The depth data from the Kinect camera was post-processed using the body tracking SDK to estimate key points on each subject (Figure 3.10).

The hip, knee, and ankle joint centers were used to determine the relative knee angle and other metrics during gait. The two systems were synced at initial heel strike and data was cut to a single gait cycle. The methods for determining when heel strike occurred can be found in section 3.5.1. Data were normalized to 100 points to represent the gait cycle from 0-100%. The maximum flexion angle was determined for each subject and compared between the two systems. Clinical metrics, including stride length, walking speed, and stance time were also compared.

Table 3.1: Probed points for each body segment

Segment	Probe Pts
Thigh	LASI, RASI, RKneeLat, RKneeMed, RTRO
Shank	RKneeLat, RKneeMed, RAnkleLat, RAnkleMed
Foot	RKneeLat, RKneeMed, RHeel, RMET5, RMET1, RToeTip

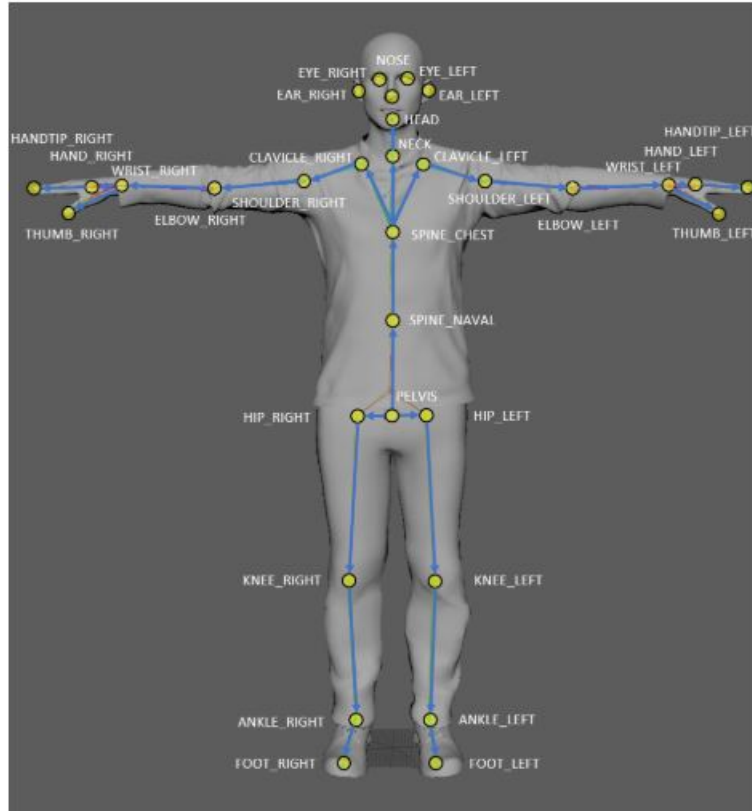


Figure 3.10: Microsoft Kinect Azure Body Tracking Joint Labels

The results from the initial system showed promise that the new Kinect Azure could be used as a method for measuring human movement instead of the gold-standard. The average difference between the maximum knee flexion angle between the two systems was 3.8° and r-value of 0.85. The correlation coefficient for walking speed and stride length were $r=0.986$ and $r=0.982$ respectively [36].

Determining Active Gold-standard Experimental Setup

A larger validation study would be conducted with a different setup. It was determined that using a passive marker system with reflective markers could be done but with modification to the lab's normal setup. Smaller reflective markers were placed on the pelvis landmarks to track the pelvis while a rigid cluster of

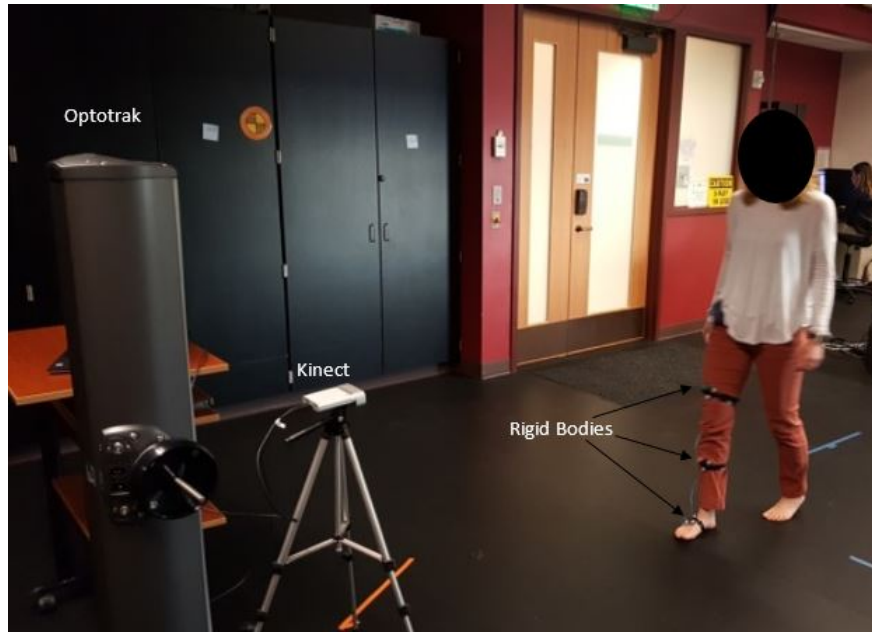


Figure 3.11: Initial data collection setup with single Kinect Azure camera and Optotrak

three markers was placed on the femur, tibia, and foot to track the respective bones. The rigid body used smaller markers as well and was placed on the front of the segment not on the side as usually done. Smaller reflective markers were placed on the medial and lateral side of the knee and ankle for calibration of the segments and removed during dynamic trials. The small reflective markers on the pelvis still create a small amount of interference in the depth data as seen in the left image of Figure 3.15.

The effect the interference had on the body tracking SDK key point locations was investigated. The segment lengths of the femur, tibia and foot were compared between a subject with markers and the same subject without markers in a static Tpose. The point cloud of each case can be seen in Figure 3.15. The relative body segments varied on average by about 5 mm for the pose with markers compared to the pose without markers. The exact values for the



Figure 3.12: Point Cloud of Subject with Smaller Markers (left). Point Cloud of Subject without Markers (right)

differences in length of the femur, tibia, and foot are in Table 3.2. The error is not clinically significant allowing for the smaller markers and rigid bodies to be used in a larger validation study. The location of the key points from the depth data with the markers and the depth data without the markers can be seen in Figure 3.13.

Table 3.2: Difference between segment lengths between capture with & without markers

	Thigh Difference (mm)	Shank Difference (mm)	Foot Difference (mm)
Left	5.7797	5.5238	2.7048
Right	5.7736	5.5819	5.5819

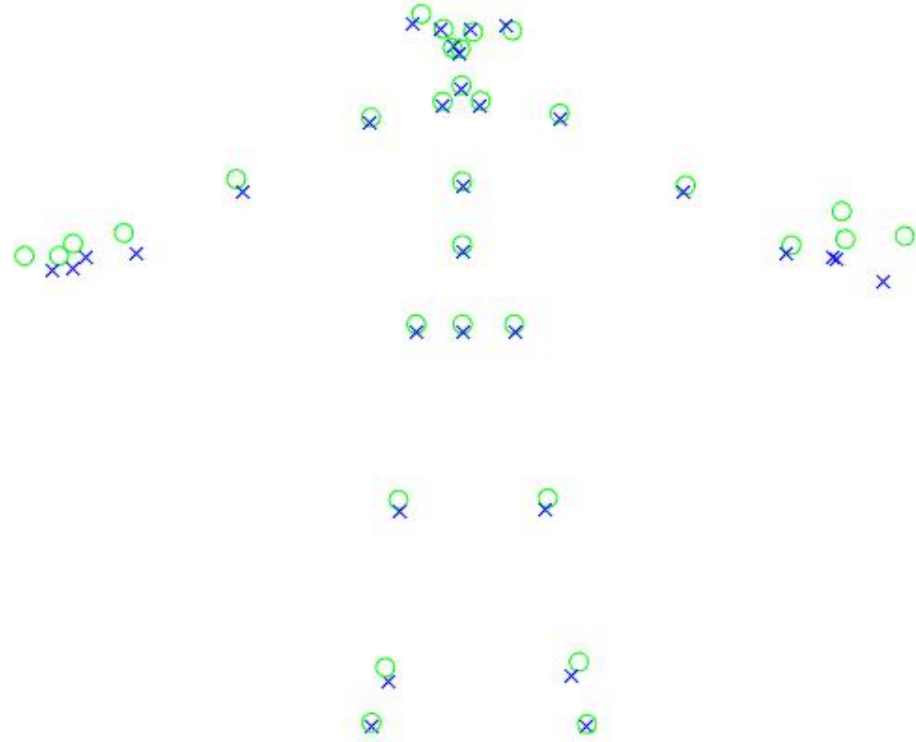


Figure 3.13: X represent key point locations when markers are present in the point cloud. O represent key point location when markers are not present in point cloud.

3.3 Data Acquisition

Once the Kinect cameras are calibrated, human movement data can be collected. The Kinect Azure cameras have different settings that allow for customizing of the configuration to best suit the needs of the user. There are four different depth modes that can be used to capture data. There are two narrow fields of view (NFOV) which are beneficial if a farther distance from the camera is needed and lower point cloud density is acceptable. In addition, there are two wide fields of view (WFOV) which is beneficial if point cloud density is important and distance from the camera is not important. Figure 3.14 shows the coverage difference between the two fields of view.

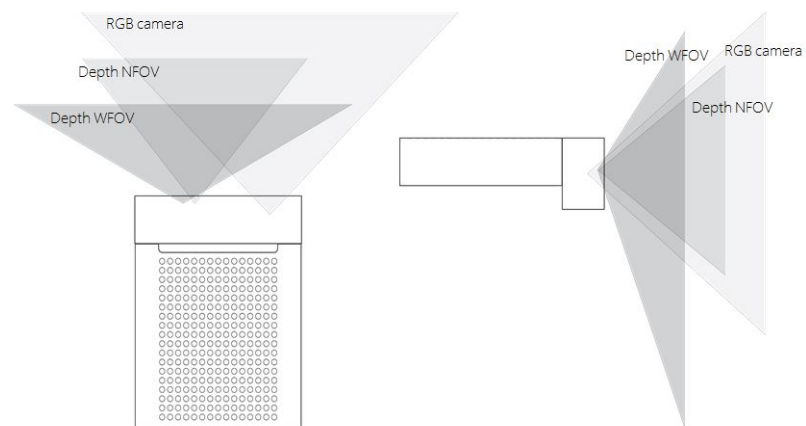


Figure 3.14: Kinect Azure documentation on coverage of depth fields of view

The WFOV is more appropriate when a comprehensive 3D surface mesh of the subject is needed because the higher point cloud density reduces the number of holes present in the surface mesh. NFOV provides the most accurate quantification of spatio-temporal and kinematics measures due to being able to collect more steps. The validation study used a four Kinect Azure camera setup in a configuration as seen in Figure ???. The study attempts to maximize the depth

mode for obtaining surface data in WFOV with three cameras, while placing the one frontal view camera in NFOV for obtaining joint tracking by collecting data from the four camera setup with different depth modes. Three cameras in a triangular configuration were used to obtain surface data for the subject and were collect in the WFOV_UNBINNED (15Hz) and the frontal camera used for body tracking was collected with the NFOV_UNBINNED (30Hz). Figure ?? displays the collection configuration. The master, Sub1, and Sub3 camera were used for generating subject surface point clouds and meshes. Sub2 was used for generating body tracking. The Kinect Azure cameras were synced together in a daisy-chain configuration using a 3.5mm mono cable less then 10 m in length as recommended by the documentation for the Kinect Azure. Each of the Kinect cameras was mounted on a tripod and set to the same height from the ground. Two laptops were used to collect the data from the four Kinects. Each laptop collected the data from two Kinect cameras. Due to the size of the data being collected, only two Kinects could be connected to a single laptop before there was a lag in data or even data streams being dropped from collection. Each Kinect was run through a separate command window to execute the k4arecorder.exe from Microsoft. The data from each Kinect Azure was output in a Matroska Video file (mkv). A more detailed step-by-step collection guide can be found in Appendix A.

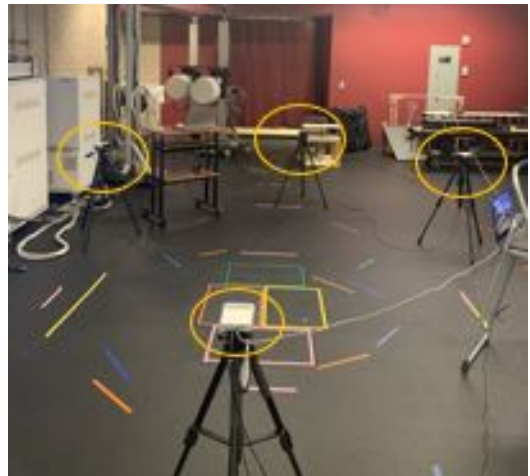
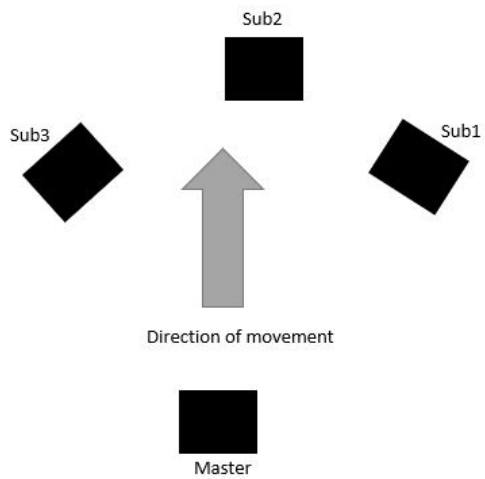


Figure 3.15: Bird-eye view of the Kinect Setup for the validation study (left). Kinect camera setup for validation study(right)

3.4 Validation Study

A validation study was conducted using a four Kinect camera setup synced with a passive marker optical system as the gold-standard. 10 subjects performed overground walking trials at a self-selected pace, a lunge, a step-up, and sit-to-stand while data was simultaneously collected from four Kinect cameras (3 @ 15Hz & 1 @ 30Hz) and 11 Vicon cameras (100 Hz). The Kinect camera collected at 30 Hz was used to generate the body tracking for determining spatio-temporal and kinematic measures. The output files from the Kinect camera were post-processed with a custom program to determine body tracking and generate surface meshes. The Vicon data was post-processed to create a labeled skeleton that would be used as an input to OpenSim to determine kinematic measures. A static trial was collected with both systems to be used to scale a musculoskeletal model as described in section 3.5.2. A representative trial for each subject was used to compare kinematic measures of hip and knee angle and spatio-temporal measures of step length, stance time, and walking speed. For kinematic measures each trial was cut to a single gait cycle and normalized to 100 points to represent 0-100% of the gait cycle. Maximum knee and hip flexion were compared between the two systems. The following sections describe in more detail the post-processing of the data collected.

3.5 Data Post-processing

The .mkv file allows for storage of many separate video files in a single file. Post processing of the .mkv file is required to access the necessary data to calculate spatio-temporal measures, kinematics, and generate a surface mesh. A custom program was created using the sensor API and body tracking API found

in the Kinect Azure documentation to generate point clouds and joint locations for each trial. The input for the program is a folder containing all the .mkv files from the trial (i.e. 4 .mkv file for a 4 Kinect Azure camera capture) and the path to the calibration text file with the transformation matrix for each subordinate Kinect Azure camera relative to the global reference frame. More details on the custom program can be found in Appendix B. Once the point cloud data and joint locations were known, further post-processing steps were taken in order to determine the desired gait metrics and body surface models. These steps are outlined in the following section.

3.5.1 Estimation of Gait Metrics

The program uses the depth data from each time point to determine the location of each of the joints, according to the Microsoft skeleton template, as seen in Figure 3.16. The program outputs a JavaScript Object Notation (json) file which contains all the information from the body tracking capabilities of the Kinect. The only information from the json file used to estimate gait metrics were the x,y,z coordinates of the joint centers. Spatio-temporal measures for walking speed, stride length, and stance time were calculated. Heel strike for each subject was required to calculate these parameters. Heel strike is defined as a maximum horizontal distance between the center of the pelvis and the heel [37]. The two joints used from the Kinect skeleton to determine the horizontal distance were the pelvis and the ankle on the right and left (Figure 3.16), respectively, for right heel strike and left heel strike. Toe-off is defined as the minimum horizontal distance from the pelvis to the toe [37]. The pelvis and foot joints were used from the Kinect skeleton (Figure 3.16) to calculate the horizontal distance. A threshold

value around the global maximum and minimum was used to determine all the heel strikes and toe-offs during a given trial. Once the heel strikes and toe-offs on the right and left side were known, spatio-temporal measures were calculated. Stride length was calculated as the horizontal distance from one heel strike to the following heel strike on the same leg. Stance time was calculated as the time between heel strike and toe-off on the same leg. Walking speed was calculated as stride length divided by corresponding time of the stride length. Once the occurrence of heel strike and toe-off are known other spatio-temporal measured could be calculated besides the three outlined above.

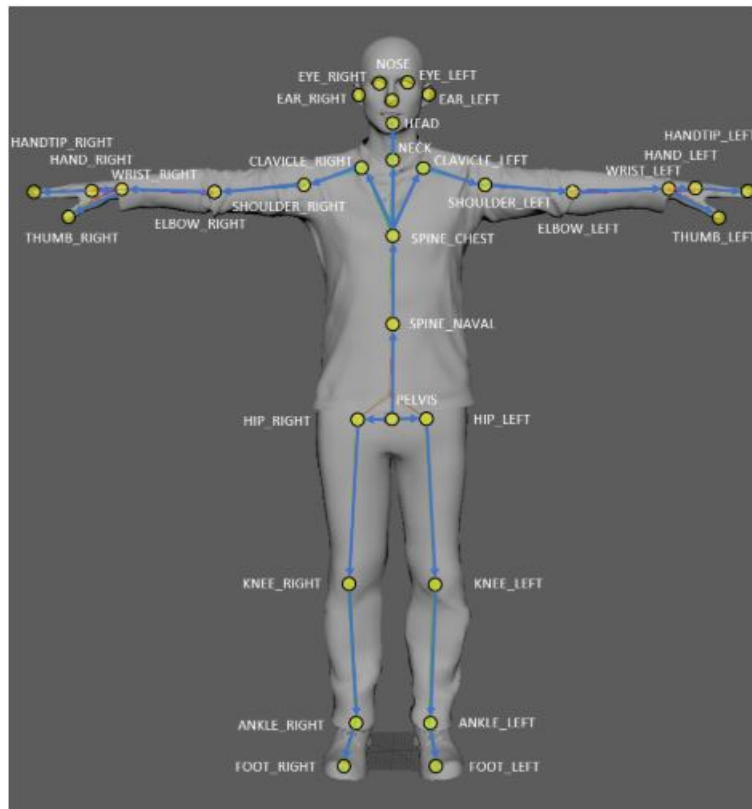


Figure 3.16: Microsoft Kinect Azure Body Tracking Joint Labels

3.5.2 Musculoskeletal Mechanics

Along with spatio-temporal measures kinematic measures can also be calculated from the Kinect and gold-standard data. The initial validation study performed on the new Kinect Azure focused on knee angles. The relative knee angle from the gold-standard was determined by creating ISB coordinate systems for the pelvis, femur, tibia, and foot. Then the data for each rigid body (femur, tibia, foot) was transform into the global coordinate system. Once all the rigid bodies were transformed into global space, the knee angle was determined by solving a system of linear equations. The result of solving the system of linear equations yields direction cosines. The direction cosines were converted into direction angles. The knee angle from the Kinect was determined by creating two vectors one from the knee joint center to the hip joint center, and the second from the knee joint center to the ankle joint center. The angle between the two vectors is the relative knee angle. The joint centers at the hip, knee, and ankle were determined by applying the body tracking SDK to the depth data for the trial.

Another method for determining kinematic quantities is to use a musculoskeletal modeling software. OpenSim was used in this study to determine kinematic measures and provide the initial inputs into the larger musculoskeletal modeling workflow that can be expanded upon in the future. The process for calculating joint angles in OpenSim started with creating a model with representative markers based on the experimental setup (Figure 3.17). Two models were created one with the corresponding Vicon markers and one with the lower extremity Kinect key points from Figure 3.16. A static trial of a subject standing in a T-pose was used to scale both models to the size of the subject. Once

the bones were scaled mass scaling was applied for better kinetic results. Additionally, for the Vicon model, the rigid bodies were moved to match the correct location of the subject after the bones were scaled based on markers placed on boney landmarks. Since the rigid bodies were not placed on boney landmarks, there was variation between subjects and the model which was corrected to improve accuracy of tracking kinematics in dynamic trials. Once there was an appropriately scaled model for the subject, the inverse kinematic tool was used in OpenSim to determine joint angles. The inverse kinematics tool required input of a model, a file of the weighting for each marker, and the file with the marker positions throughout time. OpenSim uses a least squared minimization to solve to for correct pose at each time step and determine the corresponding joint angles.

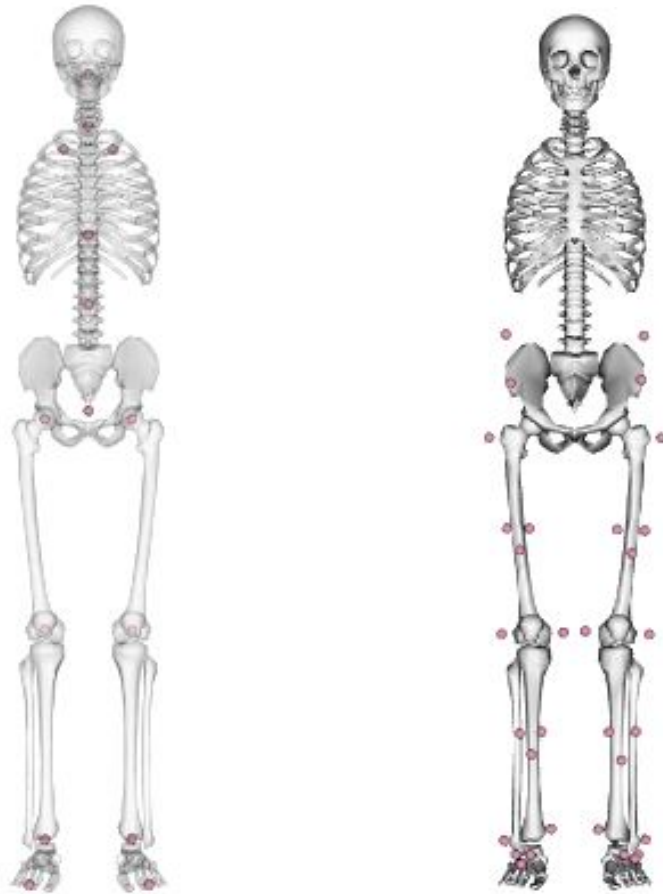


Figure 3.17: Musculoskeletal model with Kinect markers (left) and Vicon markers (right).

3.5.3 Subject Surface Geometry

The depth data from three Kinect Azure cameras is used to generate a 3-dimensional surface model. The transformation determined from calibration is used to transform all four Kinect Azure depth data points into a global reference frame. The body is isolated from the depth data using a bounding box around the joint locations from the Kinect that the subject is walking towards. Once the body is isolated, filtering is applied to the point cloud before a mesh is generated from the point cloud. A mesh is generated at each time point where depth data exists for all Kinects in the capture and is animated through time. The point clouds are used to calculate the circumference of body segments. These measures allow for a more complete understanding of a person's body habitus.

Currently BMI is used as a metric for determining the health of a person by calculating a ratio of a person's height and weight. However, this a coarse metric that can be misleading when describing a patient, and a more detailed method for understanding a person's true body shape is needed. With the 3D surface model circumference measurements can be made at any location and the full 3D representation is used to better identify their shape. This study looked at circumference measurements made on the shank about 6 in below the knee, on the thigh about 8 in above the knee and at the pelvis or waist. Physical measurements were taken of the subject using a flexible tape measure and compared to those taken from the 3D surface mesh. The circumference was calculated in the 3D mesh by isolating a small section of points at the desired location as shown in Figure 3.18 with the red points isolated from the rest of the point cloud for the thigh measurement. The pelvis circumference was calculated

by determining the maximum and minimum value in the isolated point cloud in the coronal plane to determine the diameter in this direction and then the maximum and minimum value of the isolated point cloud in the sagittal plane to determine the diameter in this direction. The pelvis was considered to be an oval and equation 3.2 was used to calculate the circumference of the pelvis. The thigh and shank circumference were calculated in a similar way, but used the isolated point cloud of the thigh and shank to determine the diameter in the coronal and sagittal directions to be used to equations 3.3 and 3.4 respectively.

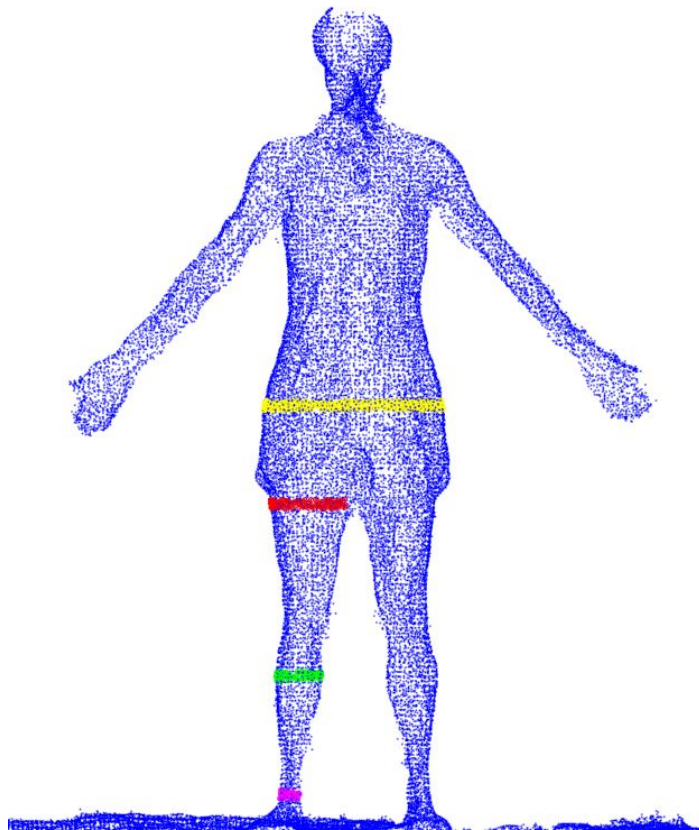


Figure 3.18: Points isolated in 3D surface point cloud for calculation of thigh circumference

$$C_{Pelvis} = \frac{2 * \pi * \sqrt{\frac{(\frac{d_{coronal}}{2})^2 + (\frac{d_{sagittal}}{2})^2}{2}} + \pi * (\frac{d_{coronal}}{2} + \frac{d_{sagittal}}{2})}{2} \quad (3.2)$$

$$C_{Thigh} = \frac{2 * \pi * \sqrt{\frac{(\frac{d_{coronal}}{2})^2 + (\frac{d_{sagittal}}{2})^2}{2}} + \pi * (\frac{d_{coronal}}{2} + \frac{d_{sagittal}}{2})}{2} \quad (3.3)$$

$$C_{Shank} = \frac{2 * \pi * \sqrt{\frac{(\frac{d_{coronal}}{2})^2 + (\frac{d_{sagittal}}{2})^2}{2}} + \pi * (\frac{d_{coronal}}{2} + \frac{d_{sagittal}}{2})}{2} \quad (3.4)$$

3.5.4 Point Cloud Driven Kinematic Model

A generic model of a thigh and shank were created in MatLab as tapered cylinders (Figure 3.19). Anthropometric data of a subject was collected to generate a more accurate model. The circumference of a subject's ankle, knee, and upper thigh along with the length of thigh and shank were used to create an equation of the line for the taper of the cylinder. The cylinder consisted of circles a 0.5 mm increments from each other to create a point cloud of the thigh and shank. In the model the shank was rotated about the x-axis (axis of knee flexion/extension angle) to create a knee angle. Once the shank point cloud was rotated a single point cloud was created of the thigh and shank by merging them together. The centroid of the subject specific and model point cloud was determined in order to

translate the model point cloud to the location of the subject specific point cloud. This allowed for both point clouds to be in a similar location before ICP and rigid CPD was performed. An iterative process was used to change the angle of the model until it aligned with the target pose from the Kinect.

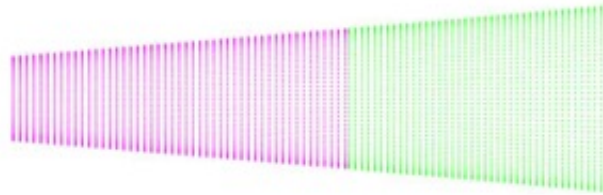


Figure 3.19: Initial alignment of the shank (magenta) and thigh (green)

Four Kinect cameras were used to create a 3D point cloud of the subject. A custom program was used to create an isolated point cloud of the subject from the Kinect depth data. From this subject specific point cloud, the right leg was segmented. Both the model point cloud and the subject specific point cloud were down sampled in a random fashion to have specified percentage of the original number of points. An ICP algorithm was used to register the model point cloud to the subject specific point cloud. The RMSE from the final iteration of ICP was used to change the knee angle to better align the model pose to the target pose. The method for changing the angle compared the previous RMSE to the current RMSE and if the error increased the last step would be subtracted from the angle and a step in the opposite direction would be taken but if the error was smaller the step would continue in the current direction. A rigid CPD algorithm was also used to register the model point cloud to the subject specific point cloud and a similar method for changing the knee angle based on the RMSE was used. A convergence tolerance was set to stop the iterative process once the RMS error

from the ICP or rigid CPD was smaller than the tolerance. Once the target pose was reached with rigid CPD a deformable CPD was run on the final model point cloud from the rigid CPD and the subject specific point cloud.

The percentage of down sampling applied to the two point clouds was varied from 20% to 60% by 10% increments when running rigid CPD. The impact on rigid CPD to the number of points in each cloud was determine by comparing the final RMSE and the number of iterations required to reach the desired tolerance.

Chapter 4

Results

The results presented here will answer the following study objectives:

- Initial validation of spatio-temporal and kinematic measures for new Kinect Azure camera
- Use of 3D surface mesh of a person to better understand body habitus

4.1 Initial System Validation

The results from the initial validation study with a single Kinect and the active marker-based gold-standard found the average difference in maximum knee angle between the two systems for all subjects to be 3.8° , with an experimental error of 6.1% between the Optotrak and Kinect systems. The minimum difference was 2.6° and the maximum difference was 6.0° . The knee angle overall during gait was similar in trend and magnitude between the two systems (Figure 4.1), with an average error of 5.1° . Initial stance phase of the gait cycle (0-20%) had the largest variation between the two systems for most subjects. The correlation coefficient of the maximum knee angle between the two system is $r=0.85$ with a $p<0.05$ (Figure 4.2). On average, when comparing clinical metrics for all subjects, the percent error for walking speed, stride length, and stance time

were 2.13%, 3.1%, and 5.1%, respectively. The correlation coefficient for walking speed and stride length were $r=0.986$ and $r=0.982$ respectively with $p<0.05$. Figure 4.3 is a representative trial of the differences in walking speed, stance time, and stride length between the two systems.

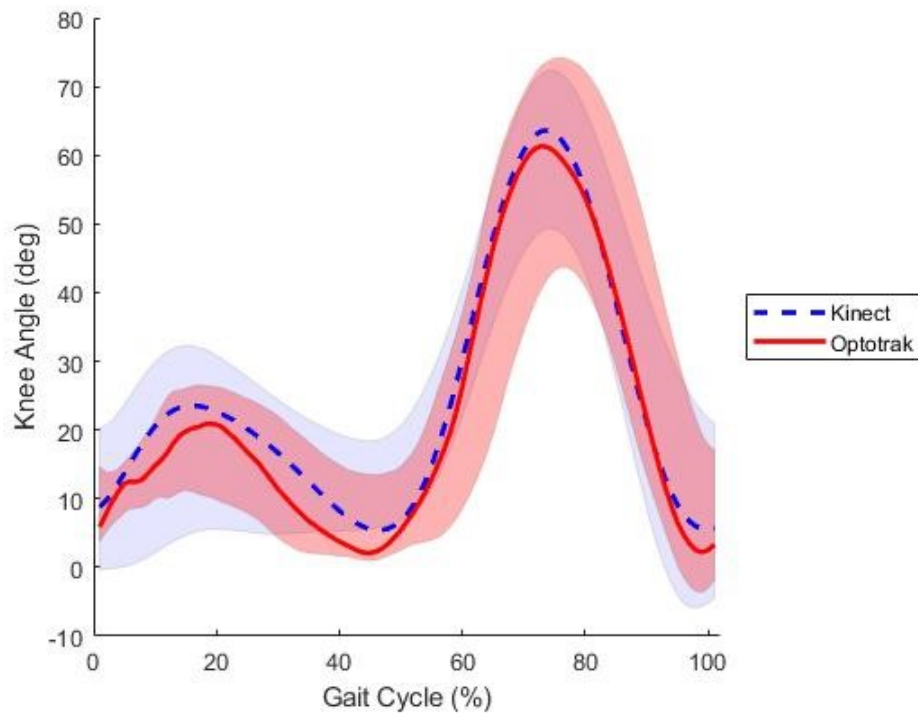


Figure 4.1: Representative knee flexion angle as a function of gait cycle measured using both systems showing ± 2 standard deviations around the mean at each point in the gait cycle for all subjects

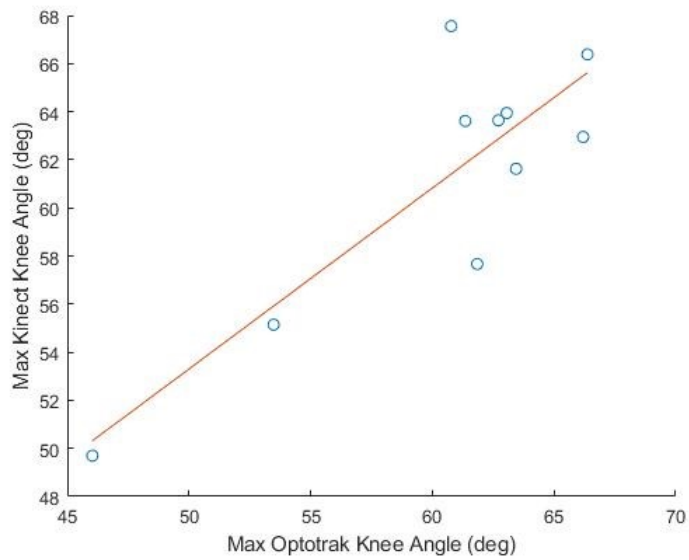


Figure 4.2: Correlation plot comparing Optotrak and Kinect max knee flexion angle for each subject

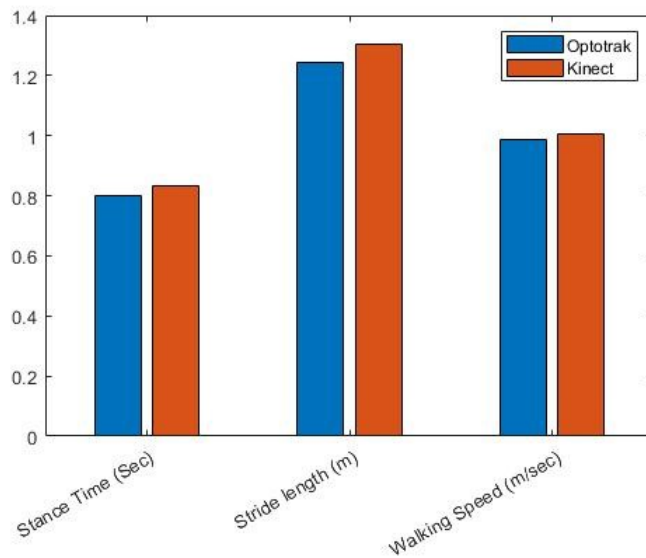


Figure 4.3: Representative trial comparing stance time, stride length, and walking speed between the two systems

4.2 Larger Validation Study

When comparing the left and right knee angle between the four Kinect Azure configuration and the Vicon the RMSE was $6.875^\circ \pm 2.8255^\circ$ and $6.0868^\circ \pm 2.8438^\circ$ respectfully. The average difference between the two systems for the maximum knee flexion angle was 2.84° with a maximum difference of 5.03° and a minimum difference of 0.245° . Figure 4.4 is a representative trial of knee angle during a single gait cycle for both the Kinect and the Vicon synced at heel strike. Stride length and waling speed had excellent correlation between the Vicon and the Kinect with r-values of 0.974 and 0.999 respectfully. Both stride length and walking speed had p-values <0.001 . Stance time had moderate correlation with a r-value of 0.591 and p-value of 0.122. A representative trial of stride length, stance time, and walking speed for the Vicon and Kinect can be seen in Figure 4.5.

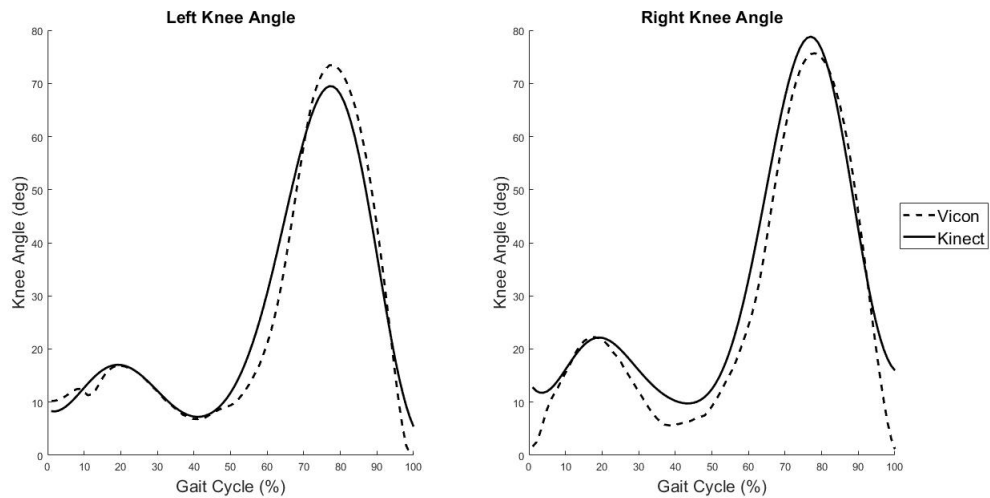


Figure 4.4: Representative subject knee angle for a gait cycle for Kinect and Vicon data

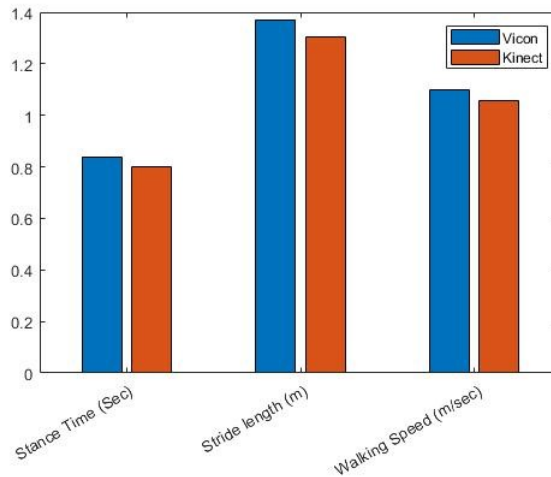


Figure 4.5: Representative subject stance time, stride length, and walking speed for Kinect and Vicon data

The Kinect Azure was also able to identify an asymmetric gait pattern.

Figure 4.6 shows an asymmetric gait pattern on the left and a symmetric gait pattern on the right.

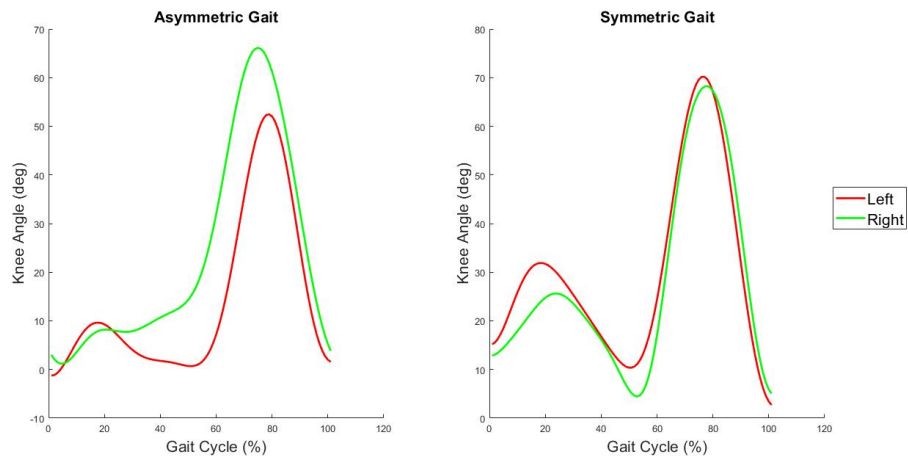


Figure 4.6: Asymmetric gait pattern (left) and symmetric gait pattern (right) as identified by Kinect Azure

4.2.1 Spatio-temporal Measures Compared to Prior Literature

The newer Kinect cameras yielded similar results to previous literature for determining spatio-temporal measures. There is excellent agreement between the Kinect and the gold-standard for walking speed and stride length, highlighted by the low p values ($p < 0.001$) and high correlation coefficients ($r > 0.97$). Clark et al. [38] found excellent agreement for walking speed and step length. The focused review of the validity of the Kinect by Springer et al. [25] found the studies to show good potential for using Kinect to determine spatio-temporal measures. The agreement between the Kinect and gold-standard can be seen in the bar plot where stance time, stride length, and walking speed are compared between the two systems (Figure 4.5).

4.2.2 Kinematic Measures Compared to Prior Literature

When determining kinematic measures with the newer Kinect camera, it had an average maximum knee flexion angle of 2.84° difference between the two systems with a correlation of $r=0.785$ compared to the gold-standard. The RMSE values for the knee flexion/extension angle were lower than that of Skals et al [5]. The Kinect Azure had an RMSE of $6.4809^\circ \pm 2.8346^\circ$ and Skals et al. found and knee angle RMSE of $11.08^\circ \pm 3.06^\circ$. The Kinect Azure both over-predicted and under-predicted the knee angle compared to the Vicon. This is in contrast to what Anderson et al. [31] found as the Kinect always predicted larger values than the gold-standard. However, the joint angle pattern between the two systems is similar to that found by Anderson et al. [31].

4.3 3D Point Cloud & Surface Mesh

The results from the custom program are seen in Figure 4.7 and Figure 4.8. A 3D point cloud of only the body is defined at each frame. An animation of all the point clouds or surface meshes for each frame allows for a visualization of the subject walking. Figure 4.8 presents three frames out of many that are used to create an animation.



Figure 4.7: Frontal (left), side (middle), and posterior (right) view of the result of the custom program yielding a 3D point cloud from four Kinect Azure cameras

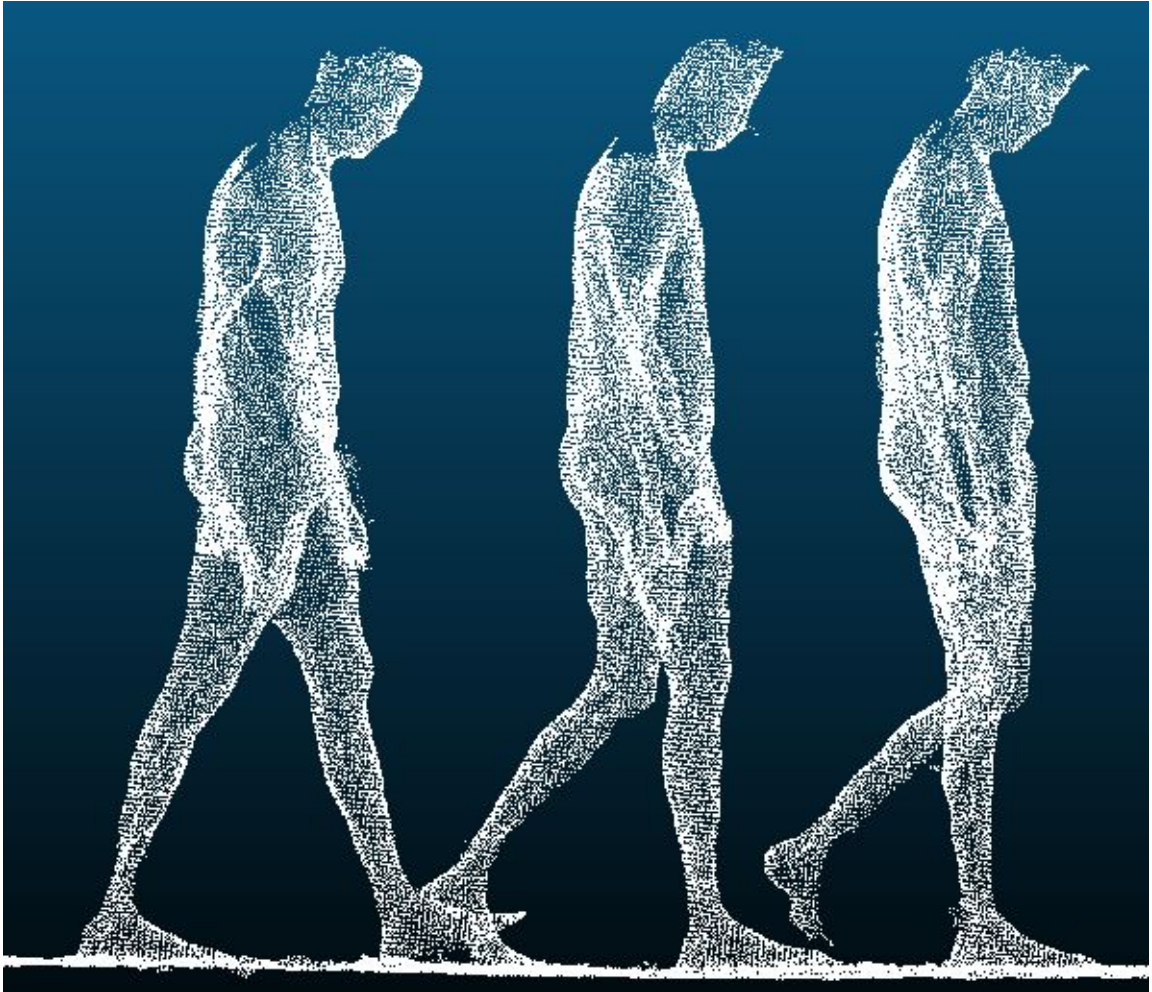


Figure 4.8: 3D point cloud subject walking at different time points

The results from the surface mesh generated by the custom program can be seen in Figure 4.9.

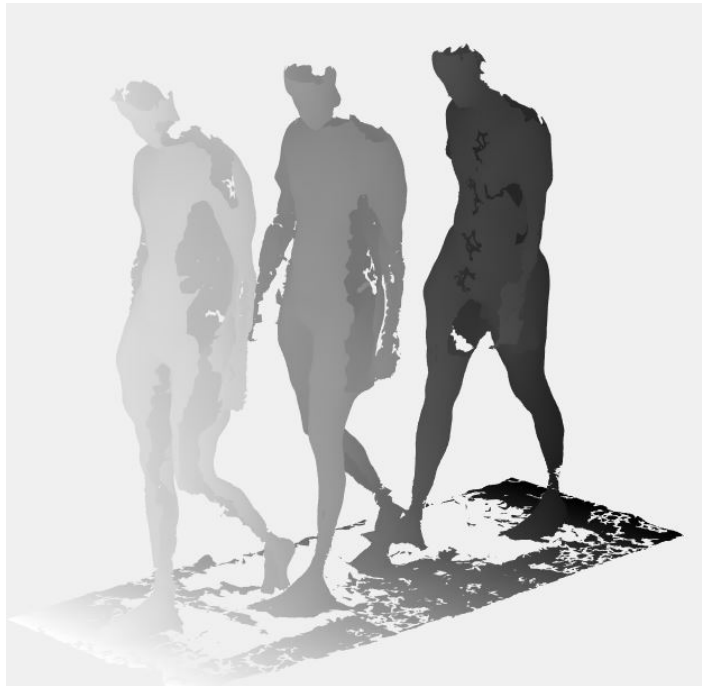


Figure 4.9: 3D surface mesh of subject walking at different time points

The results from determining the circumference from the 3D subject point cloud and measured circumference for the pelvis, thigh, and shank can be seen in Table 4.1. The section of where the circumference is measured for the pelvis, thigh and shank can be seen in Figure 4.10. The percent error in measuring the circumference of body segments is 1.25%, 2.2%, and 24% respectively for the pelvis, thigh, and shank. The point cloud estimation is greater than the measured circumference for both the pelvis and the thigh segments. The average error in the circumference estimation for the pelvis and thigh is 11.65mm. This error is not clinically significant and the low percentage errors for the pelvis and thigh segment lead to suggesting the circumference could be measured from the 3D point cloud of the subject.

Table 4.1: Difference between measured and estimated segment circumference for pelvis, thigh and shank

	C_{Pelvis} (mm)	C_{Thigh} (mm)	C_{Shank} (mm)
Measured	977.9	501.65	349.25
Point Cloud Estimation	999.0625	512.6619	265.2560

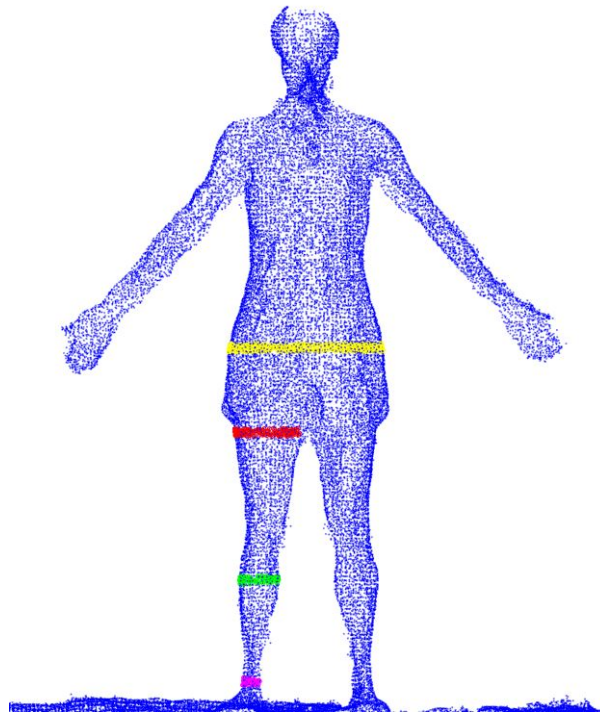


Figure 4.10: 3D surface mesh location of circumference measurement of pelvis (yellow), thigh (red), and shank (green)

4.3.1 Kinematic Model

Using ICP alone to align the two point clouds did not yield an accurate result, as seen in Figure 4.11. CPD was used with a rigid constraint on it in place of ICP and produced a reasonable RMS error of 17.6 mm and a final angle of 99.8° (Figure 4.12, right) for a point density of 40%. CPD, with the nonrigid constraint added, morphed the cylinders to fit the Kinect lower-extremity model better (Figure 4.12, left) and have a more anatomical appearance.

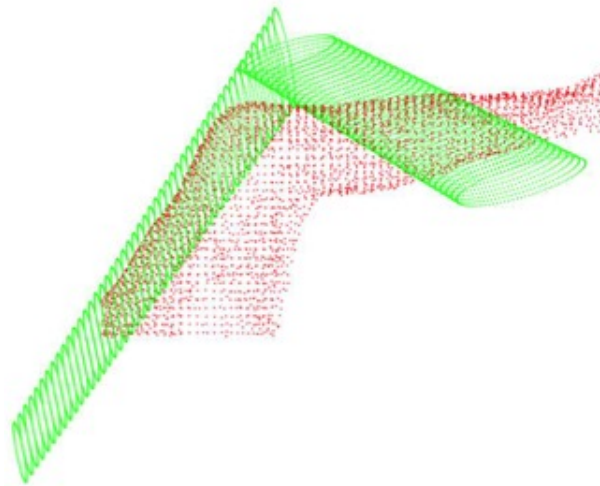


Figure 4.11: ICP's attempt at alignment of the two point clouds. Red represents the Kinect target model pose and green represents the cylindrical model pose.

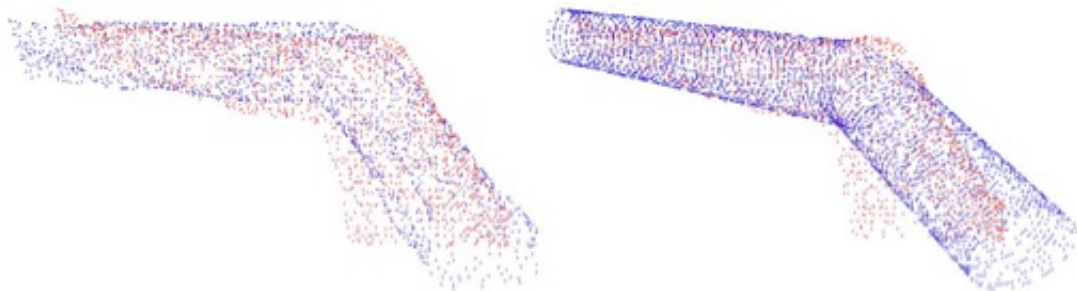


Figure 4.12: Final CPD result with both rigid and nonrigid CPD implemented (left). Final iteration of CPD rigid only (right). The red represents the Kinect target model pose and the blue represents the cylindrical model pose.

Reducing the point cloud density of both the model and subject specific meshes generally increase the error of rigid CPD and required more iterations to converge on the knee angle that best represented the subject specific point cloud. The results for five different point cloud density can be seen in Table 4.2.

Table 4.2: Effect of point cloud density on CPD results

Point Density	Knee Angle (deg)	RMSE (mm)	# Iterations
20%	111.25	17.3	30
30%	105.52	17.79	30
40%	99.79	17.6	30
50%	111.25	17.14	12
60%	116.98	17.12	11

To increase accuracy of the alignment, it was found that the final position of the cylinders was dependent upon the initial placement of the cylinders relative to the Kinect thigh and shank model.

Chapter 5

Discussion

The current gold-standard for collecting human movement has many limitations that reduce the number of people who can benefit from understanding how they are moving and identifying any deficits. There is a need for a marker-less motion capture system that addresses the current limitations and expanding the availability of a gait analysis. The new Kinect Azure from Microsoft can be used to create a marker-less motion capture system. The Kinect Azure is cost effective, requires no markers, and can be set up in any environment. Currently, there are no other systems using the newest version of the Kinect to capture human movement as designed in this study. There is little development which has occurred for any application with the Kinect Azure cameras.

The new Kinect Azure cameras were validated in two separate studies for spatio-temporal and kinematic measures. An initial study with 10 subjects was performed with a single Kinect Azure. The results showed excellent agreement for spatio-temporal measures between the two system as seen with the high correlation values and could be used in a clinical setting. The kinematic results showed promise but a larger study was needed to be conducted before it could be used in a clinical setting. Due to the promising results of a single Kinect

Azure camera a four Kinect Azure camera configuration was created. The results from the four Kinect Azure camera system also showed excellent agreement between the two systems for measuring spatio-temporal measures. The RMSE from the knee angle determined between the two system is lower than previous literature reports for the older versions of the Kinect. This suggests that the recent improvements made to the Kinect Azure depth camera can accurately estimate joint locations and spatio-temporal measures showing a promising alternative to laboratory-based motion capture for use in a clinical setting without markers or wearable sensors. Additionally, the Kinect configuration and software developed can determine clinically-relevant movement metrics, and use of this system in a clinical setting will aid in patient diagnostics, treatment, and monitoring of disease or rehabilitation progress.

The system developed used three Kinect Azure cameras to generate 3D subject data throughout time. A calibration process was designed to create a global coordinate system for all Kinect Azure cameras. Once the cameras were calibrated together, human movement could be collected and processed through a custom program to output transformed point clouds at each frame, surface meshes at each frame, and a file with joint locations using Microsoft's SDK for predicting body tracking. These files can then be post processed further to obtain relevant information about spatio-temporal measures, kinematic measures, body habitus information, or be animated through time to allow for dynamic modeling of the mesh. This is a simple device setup that will aid with the visualization of a subject's gait, while still being able to gain exact gait metrics. The 3D point cloud allows for calculations of body segments to help better inform a subject of their health than BMI. Calculating the circumference of segments like the waist, thigh,

and shank give more insight on health than BMI because extra mass in these areas would be better understood with a circumference calculation than with BMI. Additionally, the full 3D surface provides additional dimensionality that can be leverage to measure shape. Being able to extract this information from a static trial of a gait analysis will be beneficial for patient treatment. There are few systems available to achieve this end result of gait metrics, visualization of patient walking as well as health metrics around body habitus and shape modeling.

5.1 Limitations

There are several limitations for this study and the cameras used to collect human movement. This section will address the current limitations of the method of determining joint locations, lack of GRF data, using multiple TOF cameras together, and need for testing in a clinical setting.

The depth camera improvements made in the Kinect Azure camera improve the quality of the depth data generated with a capture however the body tracking still has difficulties correctly identifying the ankle and foot joint in certain situations. As mentioned before, the body tracking has difficulties with the ankle and foot joint locations when the foot is in contact with the floor. The poor tracking of the feet leads to errors in calculating kinematics at the ankle and knee along with errors in identify gait events. There is a need for a more accurate method for determining joint locations or joint angles from the depth data. A proposed method by Skals et al. [5] is to explore direct tracking of the depth data. Skals et al. [5] suggests direct tracking of the depth data by a musculoskeletal model but there could also be direct tracking using a target pose point cloud with an iterative method to align the two sets of point clouds together with the angle

between segments known in the target model as presented in this study. However, there is a more robust model needed in order to be able to track all lower body joint angles and not only the knee angle. In addition to the body tracking having troubles with the ankle and foot joint, there is still the problem of identifying joints when they are occluded by another part of the body. A method for determining joints where occlusions are not an issue is needed for a more robust monitoring system. Using the complete 3D surface data to determine joint angles could address the issue of occlusion for determining kinematic quantities but there would still need to be a better method for determining exact joint locations. As joint locations are important for determining spatio-temporal measures and detecting gait events.

Another limitation of this study is that it did not consider a method for predicting GRF data. GRF data is valuable for identifying discrepancies between limbs during walking and is needed to fully take the gait lab into the field and transition from a laboratory setting to a clinical setting. The current system performs the kinematic portion of a traditional gait analysis, adding the GRF prediction would complete the system. The discrepancies or deficits identified can be used for disease diagnosis and tracking. In order to have a complete system that can be used in a clinical setting, a method for determining GRF data without force plates is necessary. There are a few studies with methods already proposed for predicting GRF data without force plates, but none of them have been driven by the kinematic data from the Kinect Azure. There is a need for a study to explore the best method for determining GRF data when using Kinect

Azure to collect kinematics. However, the scope of a study to predict GRF from kinematic data is large and outside the objectives of this study due to the large amount of data needed.

Along with not considering a method for prediction of GRF data another limitation to the system is having multiple TOF cameras running simultaneously that can cause there to be interference in the depth data. The interference can lead to missing depth data or noisy depth data. There are methods to recover the lost depth data from having one of the TOF cameras interfering with the signal for another camera. This study does not account for this interference and therefore some of the quality of the depth data is reduced. In order to improve the accuracy of the depth data at every frame, a method for recovering the depth data when interference occurs is needed.

A final limitation of the study is the data collected was done so in a lab and not in a clinical setting. The four Kinect Azure camera system used in this study is also more than what most clinics would use. A two camera system would be better suited for a clinical setting. Therefore, it would be beneficial to setup a two camera system and determine the easiest method for calibrating the two cameras into a global reference frame. Additionally, to eliminate the need for trained personnel to run the system in a clinical setting the would need to be a user-friendly GUI to collect and process the data. This study created an initial system that would generate all the potential data required to better understand human movement but not all the features would be necessary in every clinical setting. Testing of the system in a clinic is needed to make sure the same results for human movement are found as in the lab.

5.2 Future Work

Future work would include further developing the program created in order to eliminate the need of multiple different programs, update the whole filling algorithm used to generate surface meshes, and simplify the calibration process.

The program created for this study to process the output files from the Kinect Azure could be improved to generate a more complete program. Currently there are three separate programs needed to collect and process the data. One program collects the data, another program processes the output from the Kinect Azure into a usable format, and the final program calculates spatio-temporal parameters, kinematic parameters, and information about body habitus. In the future a single program that completed all of the steps would be needed for a clinical setting. The current program would need to be updated to be able to collect the data and then process the data into a usable form and the calculations currently done in MatLab to generate information about gait metrics would need to be added to create a complete program. This would allow for a report to be generated by running a single program and not requiring the use of other programs like MatLab.

Additionally, to generate more realistic surface meshes, a better hole filling algorithm would be needed. Currently there are holes in the mesh which are not filled because finding an algorithm to fill only anatomically holes is challenging. It is difficult for an algorithm to know if the hole actually exists on the body or if it is a "fake" hole like between the side of the body and an arm.

Due to some holes still being present in the current version of the custom program, the body is not completely realistic. When the surface mesh is animated, the holes are visible.

Finally, the calibration process for all the Kinect cameras into a global space could be improved to reduce the amount of manual steps required to generate a transformation matrix for each of the subordinate cameras. It would be beneficial if an automated process was developed for calibrating the Kinect Azure cameras. This would increase the efficiency of the system and has the potential to allow for the Kinect Azure cameras to be calibrated in a clinic without the need for a trained personnel.

In conclusion, this study developed a marker-less motion capture system that could be used in a clinical setting. The Kinect Azure cameras are inexpensive, have a minimal collection time and allow for movement to be captured anywhere like in an exam room or hallway. The system developed uses four Kinect Azure cameras to create a 3D model of a subject walking or performing activities of daily living throughout time. A validation showed the Kinect Azure can accurately estimate gait metrics showing promise for a fixed-installation in a clinical setting without the need for markers or wearables. Additionally, the access to patient surface geometry will enable body habitus data to be included as a more comprehensive metric beyond the current use of BMI.

Bibliography

- [1] R. Baker, "The history of gait analysis before the advent of modern computers," *Gait & posture*, vol. 26, no. 3, pp. 331–342, 2007.
- [2] A. D. Kuo and J. M. Donelan, "Dynamic principles of gait and their clinical implications," *Physical therapy*, vol. 90, no. 2, pp. 157–174, 2010.
- [3] C. Frigo, M. Rabuffetti, D. Kerrigan, L. Deming, and A. Pedotti, "Functionally oriented and clinically feasible quantitative gait analysis method," *Medical and Biological Engineering and Computing*, vol. 36, no. 2, pp. 179–185, 1998.
- [4] B. Müller, W. Ilg, M. A. Giese, and N. Ludolph, "Validation of enhanced kinect sensor based motion capturing for gait assessment," *PloS one*, vol. 12, no. 4, p. e0175813, 2017.
- [5] S. Skals, K. P. Rasmussen, K. M. Bendtsen, J. Yang, and M. S. Andersen, "A musculoskeletal model driven by dual microsoft kinect sensor data," *Multibody System Dynamics*, vol. 41, no. 4, pp. 297–316, 2017.
- [6] S. Corazza, L. Muendermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. P. Andriacchi, "A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach," *Annals of biomedical engineering*, vol. 34, no. 6, pp. 1019–1029, 2006.
- [7] M. Eltoukhy, C. Kuenze, M. S. Andersen, J. Oh, and J. Signorile, "Prediction of ground reaction forces for parkinson's disease patients using a kinect-driven musculoskeletal gait analysis model," *Medical engineering & physics*, vol. 50, pp. 75–82, 2017.

- [8] Y. Hirano, D. Kushida, and H. Matsumoto, "Contactless motion analysis system using a kinect and musculoskeletal model," in *2017 IEEE Life Sciences Conference (LSC)*, pp. 308–311, IEEE, 2017.
- [9] J. Fuller, L.-J. Liu, M. Murphy, and R. Mann, "A comparison of lower-extremity skeletal kinematics measured using skin-and pin-mounted markers," *Human movement science*, vol. 16, no. 2-3, pp. 219–242, 1997.
- [10] C. Buckley, L. Alcock, R. McArdle, R. Z. U. Rehman, S. Del Din, C. Mazzà, A. J. Yarnall, and L. Rochester, "The role of movement analysis in diagnosing and monitoring neurodegenerative conditions: Insights from gait and postural control," *Brain sciences*, vol. 9, no. 2, p. 34, 2019.
- [11] K. Aminian, C. Trevisan, B. Najafi, H. Dejnabadi, C. Frigo, E. Pavan, A. Telonio, F. Cerati, E. Marinoni, P. Robert, *et al.*, "Evaluation of an ambulatory system for gait analysis in hip osteoarthritis and after total hip replacement," *Gait & posture*, vol. 20, no. 1, pp. 102–107, 2004.
- [12] M. Alnowami, A. Khan, A. H. Morfeq, N. Alothmany, and E. A. Hafez, "Feasibility study of markerless gait tracking using kinect," *Life Science Journal*, vol. 11, no. 7, pp. 514–523, 2014.
- [13] T. Seel, J. Raisch, and T. Schauer, "Imu-based joint angle measurement for gait analysis," *Sensors*, vol. 14, no. 4, pp. 6891–6909, 2014.
- [14] A. Leardini, G. Lullini, S. Giannini, L. Berti, M. Ortolani, and P. Caravaggi, "Validation of the angular measurements of a new inertial-measurement-unit based rehabilitation system: comparison with state-of-the-art gait analysis," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 136, 2014.

- [15] T. Watanabe and H. Saito, "Tests of wireless wearable sensor system in joint angle measurement of lower limbs," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5469–5472, IEEE, 2011.
- [16] E. P. Washabaugh, T. Kalyanaraman, P. G. Adamczyk, E. S. Claflin, and C. Krishnan, "Validity and repeatability of inertial measurement units for measuring gait parameters," *Gait & posture*, vol. 55, pp. 87–93, 2017.
- [17] S. S. Yeo and G. Y. Park, "Accuracy verification of spatio-temporal and kinematic parameters for gait using inertial measurement unit system," *Sensors*, vol. 20, no. 5, p. 1343, 2020.
- [18] C. Yang, U. Ugbolue, B. Carse, V. Stankovic, L. Stankovic, and P. Rowe, "Multiple marker tracking in a single-camera system for gait analysis," in *2013 IEEE International Conference on Image Processing*, pp. 3128–3131, IEEE, 2013.
- [19] T. T. Verlekar, L. D. Soares, and P. L. Correia, "Automatic classification of gait impairments using a markerless 2d video-based system," *Sensors*, vol. 18, no. 9, p. 2743, 2018.
- [20] C. Krishnan, E. P. Washabaugh, and Y. Seetharaman, "A low cost real-time motion tracking approach using webcam technology," *Journal of biomechanics*, vol. 48, no. 3, pp. 544–548, 2015.
- [21] *Lokomat*, (accessed July 28, 2020).
- [22] S. Totilo, *Microsoft: Project Natal Can Support Multiple Players, See Fingers'*, 2009 (accessed June 16, 2020).

- [23] S. Totilo, *Why Xbox 360 New Controller Is Called 'Natal'*, 2009 (accessed June 16, 2020).
- [24] T. Warren, *A closer look at Microsoft's new Kinect sensor*, 2019 (accessed June 16, 2020).
- [25] S. Springer and G. Yogev Seligmann, "Validity of the kinect for gait assessment: A focused review," *Sensors*, vol. 16, no. 2, p. 194, 2016.
- [26] R. A. Clark, K. J. Bower, B. F. Mentiplay, K. Paterson, and Y.-H. Pua, "Concurrent validity of the microsoft kinect for assessment of spatiotemporal gait variables," *Journal of biomechanics*, vol. 46, no. 15, pp. 2722–2725, 2013.
- [27] J. Behrens, C. Pfüller, S. Mansow-Model, K. Otte, F. Paul, and A. U. Brandt, "Using perceptive computing in multiple sclerosis-the short maximum speed walk test," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 89, 2014.
- [28] S. Vernon, K. Paterson, K. Bower, J. McGinley, K. Miller, Y.-H. Pua, and R. A. Clark, "Quantifying individual components of the timed up and go using the kinect in people living with stroke," *Neurorehabilitation and neural repair*, vol. 29, no. 1, pp. 48–53, 2015.
- [29] D. J. Geerse, B. H. Coolen, and M. Roerdink, "Kinematic validation of a multi-kinect v2 instrumented 10-meter walkway for quantitative gait assessments," *PloS one*, vol. 10, no. 10, p. e0139913, 2015.
- [30] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis," *Journal of medical engineering & technology*, vol. 38, no. 5, pp. 274–280, 2014.

- [31] M. S. Andersen, J. Yang, M. de Zee, L. Zhou, S. Bai, and J. Rasmussen, "Full-body musculoskeletal modeling using dual microsoft kinect sensors and the anybody modeling system," in *14th International Symposium on Computer Simulation in Biomechanics*, pp. 23–24, 2013.
- [32] L. Ren, R. K. Jones, and D. Howard, "Whole body inverse dynamics over a complete gait cycle based only on measured kinematics," *Journal of biomechanics*, vol. 41, no. 12, pp. 2750–2759, 2008.
- [33] S. E. Oh, A. Choi, and J. H. Mun, "Prediction of ground reaction forces during gait based on kinematics and a neural network model," *Journal of biomechanics*, vol. 46, no. 14, pp. 2372–2380, 2013.
- [34] M. Damsgaard, J. Rasmussen, S. T. Christensen, E. Surma, and M. De Zee, "Analysis of musculoskeletal systems in the anybody modeling system," *Simulation Modelling Practice and Theory*, vol. 14, no. 8, pp. 1100–1111, 2006.
- [35] Y. He, B. Liang, Y. Zou, J. He, and J. Yang, "Depth errors analysis and correction for time-of-flight (tof) cameras," *Sensors*, vol. 17, no. 1, p. 92, 2017.
- [36] A. Eustace, B. Mason, C. Myers, and P. Rullkoetter, "Validation of a new biplane depth-camera-based automated gait assessment for a clinical setting," in *ORS 2020 Annual Meeting*, 2020.
- [37] J. Zeni Jr, J. Richards, and J. Higginson, "Two simple methods for determining gait events during treadmill and overground walking using kinematic data," *Gait & posture*, vol. 27, no. 4, pp. 710–714, 2008.

- [38] R. A. Clark, Y.-H. Pua, A. L. Bryant, and M. A. Hunt, "Validity of the microsoft kinect for providing lateral trunk lean feedback during gait retraining," *Gait & posture*, vol. 38, no. 4, pp. 1064–1066, 2013.

Appendix A

Kinect Users Guide

The following is a user guide to assist in the setup and data collection of the Kinect Azure camera(s)

A.1 Hardware requirements

- Kinect Azure Camera(s)
- Power cable(s)
- USB-C data cable(s)
- 3.5mm audio cable(s) (if syncing multiple devices)
- Tripod(s)
- Laptop(s)

A.2 Setting Up Multiple Kinects

1. Connect the Kinect cameras to the tripods and set the height of the tripod to BLANK
2. Plug the power cable into the Kinect
3. Plug the USB-C data cable into the Kinect
4. Set the tripods with Kinects in the desired spot in the collection space

5. Decide which camera is going to be the master camera. See Figure A.1 for how the cameras were setup for this study. Plug one of the audio cables into the sync out port of the master Kinect. Plug the other end of the audio cable into the sync in port of the subordinate 1 Kinect.

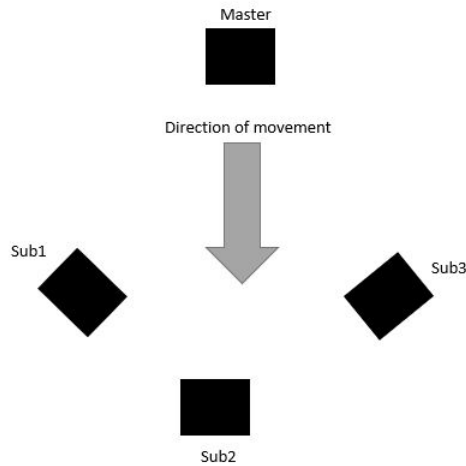


Figure A.1: Bird-eye view of the Kinect Setup for the validation study.

6. Plug another audio cable into the sync out port of subordinate 1 Kinect and plug the other end of the audio cable into the sync in port of subordinate 2 Kinect.
7. Plug another audio cable into the sync out port of subordinate 2 Kinect and plug the other end of the audio cable into the sync in port of subordinate 3 Kinect.
8. Plug the USB-C cable from the master Kinect and subordinate 1 Kinect into one laptop
9. Plug the USB-C cable from subordinate 2 Kinect and subordinate 3 Kinect into the other laptop

A.3 Calibrating Kinects

1. Place the 3 calibration object throughout the capture volume making sure they are at different height and spread through the entire volume for best calibration results as in Figure A.2



Figure A.2: Bird-eye view of the Kinect Setup for the validation study.

2. Open the `k4aviewer.exe` for all Kinect cameras and select `WFOV_UNBINNED` depth and start the camera.
3. Ensure all corners of the calibration objects are visible in the depth view (Figure). Those are the 12 points which will be used to calibrate all subordinate cameras to the master or global reference frame. If all corners are not visible adjust the angle of the calibration object until they are.
4. Close all the `k4a` viewers (If a viewer is still open and the Kinect is running

on it then you wont be able to start it from the command window for data collection)

5. Open the remote desktop to be able to control both laptops from one laptop. Any program can be used this study used "Remote Desktop Connection" and the following steps layout the process for using "Remote Desktop Connection"

(a) For remote desktop you will need the IP address of the other laptop the following steps describe how to obtain the IP address

(b) Open a command window and type ipconfig then enter

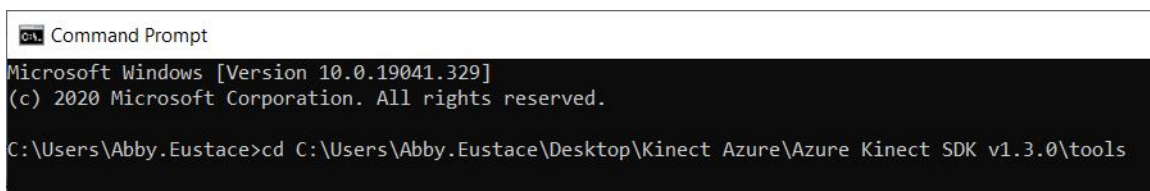
(c) The IPV4 address is the one you will need

(d) Type the IP address into the computer section of the remote desktop connection window and type PAULRX15\Abby into the username

(e) Then a window will pop up asking for the password of the computer you are trying to remote into

6. Open a command windows on each laptop

7. Use the command cd PATH to navigate to the folder on the laptop with the k4arecorder.exe (i.e. cd C:\Users\Abby.Eustace\Desktop\Kinect Azure\Azure Kinect SDK v1.3.0\tools) Figure A.3



```
Command Prompt
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\Abby.Eustace>cd C:\Users\Abby.Eustace\Desktop\Kinect Azure\Azure Kinect SDK v1.3.0\tools
```

Figure A.3: Example of the command to change the working directory of the command window

8. Now the working directory for the command window is the desired folder

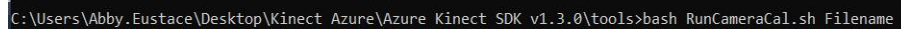
9. The k4arecorder executable will be used to collect data from the Kinect cameras. The following is a list of options for the k4arecorder.exe

- -h, -help Prints list of commands
- -list List the currently connected K4A devices
- -device Specify the device index to use (default: 0)
- -l Limit the recording to N seconds (default: infinite)
- -c Set the color sensor mode (default: 1080p), available options: 3072p, 2160p, 1536p, 1440p, 1080p, 720p, 720p_NV12, 720p_YUY2, OFF
- -d Set the depth sensor mode (default: NFOV_UNBINNED), available options: NFOV_2X2BINNED, NFOV_UNBINNED, WFOV_2X2BINNED, WFOV_UNBINNED, PASSIVE_IR, OFF
- -depth-delay Set the time off-set between color and depth frames in microseconds (default: 0) A negative value means depth frames will arrive before color frames. The delay must be less than 1 frame period
- -r Set the camera frame in Frames per Second. Default is the maximum rate supported by the camera mode. Available options: 30, 15, 5
- -imu Set the IMU recording mode (ON, OFF, default: ON)
- -external-sync Set the external sync mode (Master, Subordinate, Standalone default: Standalone)

- `-sync-delay` Set the external sync delay off the master camera in microseconds (default: 0) This setting is only valid if the camera is in Subordinate mode
- `-e` Set the manual exposure value (-11 to 1) for the RGB camera (default: 0)

10. The following set of commands are used to trigger two Kinect from a single command window using bash scripting for calibration

(a) `bash RunCameraCal.sh Filename`



```
C:\Users\Abby.Eustace\Desktop\Kinect Azure\Azure Kinect SDK v1.3.0\tools>bash RunCameraCal.sh Filename
```

Figure A.4: Example of the command line argument to run 2 Kinects with bash scripting for calibration

(b) The command arguments are the same for both laptops.

(c) Make sure to fire the bash command on the laptop with sub2 and sub3 before firing the bash command on the master Kinect. The master Kinect needs to be fired last to ensure a sync signal is sent to the subordinate Kinects

If you want to run the Kinects for individual command windows follow the following steps

11. Open a command window per Kinect plugged into a laptop. (i.e. if 2 Kinects plugged into a single laptop open 2 command windows)
12. The following set of commands were used to trigger of the devices from the command window

- (a) Master:k4arecorder.exe –device 0 –external-sync Master
-d WFOV_UNBINNED -l 5 Filename.mkv

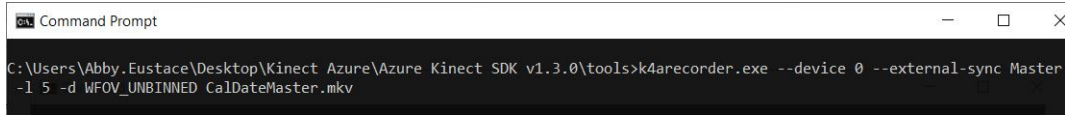


Figure A.5: Example of the command used to trigger a master Kinect camera

- (b) Subordinate:k4arecorder.exe –device 0 –external-sync Subordinate
-d WFOV_UNBINNED -l 5 Filename.mkv

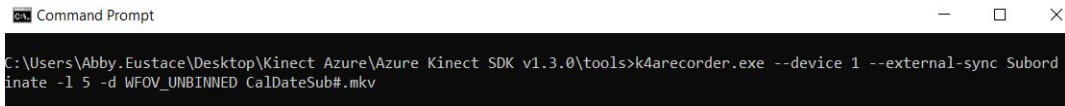


Figure A.6: Example of the command used to trigger a subordinate Kinect camera

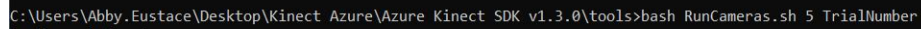
- (c) The device number for each Kinect will depend on which USB port the device is plugged into. Trigger each Kinect through the command window to ensure the correct device is running.
- (d) Trigger the subordinate cameras in the command window and the master camera last
- (e) If a length of capture is not set in the command window arguments then use Ctrl-C to end each capture
- (f) All the .mkv files will be saved to the working directory folder of the command window unless a specific folder path is defined

A.4 Running the Kinects for Collection of Human Movement

1. Once the subject is ready for data to be collected start with collecting a static Tpose.

2. The bash scripting command for collecting subject trials is the following

(a) bash RunCameraCal.sh Filename



```
C:\Users\Abby.Eustace\Desktop\Kinect Azure\Azure Kinect SDK v1.3.0\tools>bash RunCameras.sh 5 TrialNumber
```

Figure A.7: Example of the command line argument to run 2 Kinects with bash scripting

(b) The command arguments are the same for both laptops.

(c) Make sure to fire the bash command on the laptop with sub2 and sub3 before firing the bash command on the master Kinect. The master Kinect needs to be fired last to ensure a sync signal is sent to the subordinate Kinects

If you are running the collection in separate command windows per Kinect use the following

3. Command window arguments are the same for as they were for calibration unless a different depth mode is desired then update the depth mode argument. Make sure to also change the filename or the calibration file will be over-written. (i.e. T01_Sub#.mkv or T01_Master.mkv)
4. Depending on the activity the length of the collection time may also need to be changed. 5 seconds is usually long enough for walking
5. After each trial make sure to update the filename

Appendix B

Kinect Processing Guide

B.1 Introduction

The Kinect Azure Processor is a software designed to process depth data from multiple Kinect Azures and output joint data, point clouds, and meshes. It takes in a set of Kinect captures with depth data, along with a configuration file for each camera setup. This guide will not cover how to record captures. Instructions for recording captures with the Kinect Azure can be found here:

<https://docs.microsoft.com/en-us/azure/Kinect-dk/record-external-synchronized-units>

B.2 Basics

The source code can be found at
<https://github.com/jeremykeller001/Kinect-Azure-Processing>

This repository may be marked as private and inaccessible to normal users. For access issues, please contact Jeremy at jeremykeller001@gmail.com The code must be built in Visual Studios (2019 version recommended).

System requirements:

- Nvidia CUDA Toolkit 10.2
- Point Cloud Library 1.10.1 (with 3rd party libraries)

- Azure Kinect Sensor SDK 1.3.0
- Azure Kinect Body Tracking SDK 1.0.1
- CMake 3.15

The Kinect Azure Processor program is run through calling the executable via the command line. It takes in a series of arguments to determine its behavior. The full feature set and usage guide will be covered at the end. A usage guide can also be printed to the console by specifying the `-h` argument. Once built, it can be started in the command line from:

```
“code directory64-Debug.exe”
```

Output `.mkv` files for each set of captures must be placed within a folder designated for that specific group. This program will automatically process all the `mkv` files in the folder path it is given, so do not place multiple sets of captures within the same folder for processing.

Theoretically, any number of cameras are supported, but this program has only been tested with up to a 4-camera setup. Captures may be in different depth modes, but must be run at the same frames per second. There is a configuration file read in that can be configured to align captures using different resolutions and fields of view. Calibration should be performed once per camera alignment. If any camera is bumped or moved, the captures should be recalibrated.

B.3 Calibration

For multiple Kinect captures to be processed, they must be calibrated to the same coordinate system. This is done by applying a 4x4 transformation matrix

to each capture. The exact method of determining the transformation matrices is up to the user, but in this guide we will cover a manual point pair alignment based calibration method.

The software program, CloudCompare, will be used to aid in this calibration process.

Note: This guide will be performed using an example 4 camera setup. The naming convention for the captures is Master, Sub1, Sub2, and Sub3. All captures will be aligned to the master coordinate system.

B.3.1 Capture

Prep the calibration object by placing the calibration object at random spots throughout the capture space, such that all cameras will have a clear view of all of them. The goal of this is to generate key points with x, y, and z coordinate variance. It is recommended to keep calibration objects at least 6" away from each other so there is a distinct corners when viewing the depth data.

(Sample calibration layout with a single box with markers on it)

Assign a number to each corner. This number will remain the same, irrespective of different camera perspectives. It is recommended to draw this out on a piece of paper for reference later.

(Sample marker number assignment based on the calibration image above)

Obtain a 5 second capture at the desired field of view from each camera,

running in synchronous mode. It is recommended to use the highest resolution available so that the markers are captured with as much detail as possible. The commands for collecting the calibration data can be found in the Kinect User Guide Appendix A.

Process the captures by running the Kinect Azure Processor in initial calibration mode. The initial calibration behavior can be triggered by running the code in calibration mode without a transform file. This can be specified by using the `-c` argument.

Usage example:

```
./KinectAzureProcessor.exe "\\path\\to\\mkv\\capture\\folder" -c
```

This will output the 15th group frame processed by default. The first 30 sets of frames from each capture will be used to aid the internal group calibration process, so they will be skipped over automatically. This corresponds to about 1.5 seconds in for a 30fps capture, or 3 seconds in for a 15fps capture.

If the 15th frame is a bad capture, or maybe not the specific spot in the capture you want, you may manually select a frame by adding the option to the command line using the `-f` argument followed by a frame number. Example to request frame 20 instead of 15:

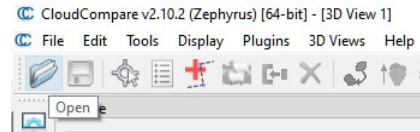
```
-f 20
```

The output files for this calibration process will be generated in the same folder as the original `.mkv` files. They will be raw point cloud dumps from each individual `.mkv` file from a single group frame. The file naming convention is:

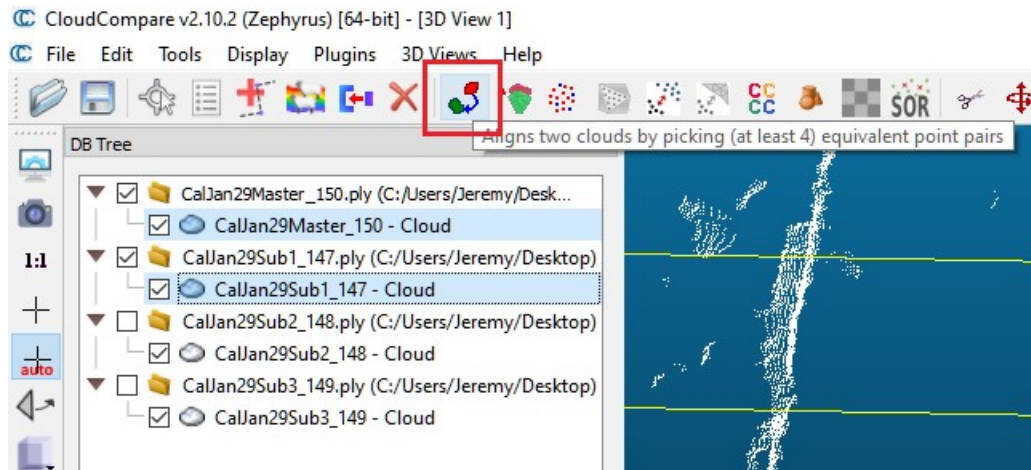
```
Unfiltered_IndividualPc_GroupNumber_OriginalFileName.pcd
```

B.3.2 Alignment Using CloudCompare

Open up CloudCompare and load all of the generated point clouds from the captures. There should be one per Kinect camera.

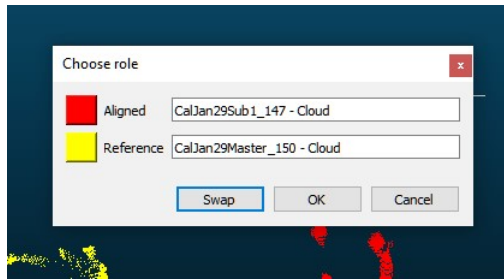


Once all the files are loaded, uncheck all boxes that do not correspond to the Master and Sub1 clouds. Select both the Master and Sub1 files (the cloud files, not the enclosing folders). Holding Control + Left Mouse Click allows you to select multiple files. Once the files are selected, click on the Point Pair Alignment function in the upper toolbar.



A popup will appear, asking you to designate an alignment cloud. Master will be the reference, and Sub1 will be the to-align cloud. If only an Align option is displayed, designate this as the Sub1 capture.

Begin the Point Pair Alignment process by showing the master cloud. Refer to the drawing you made earlier, with the numbers for each point at

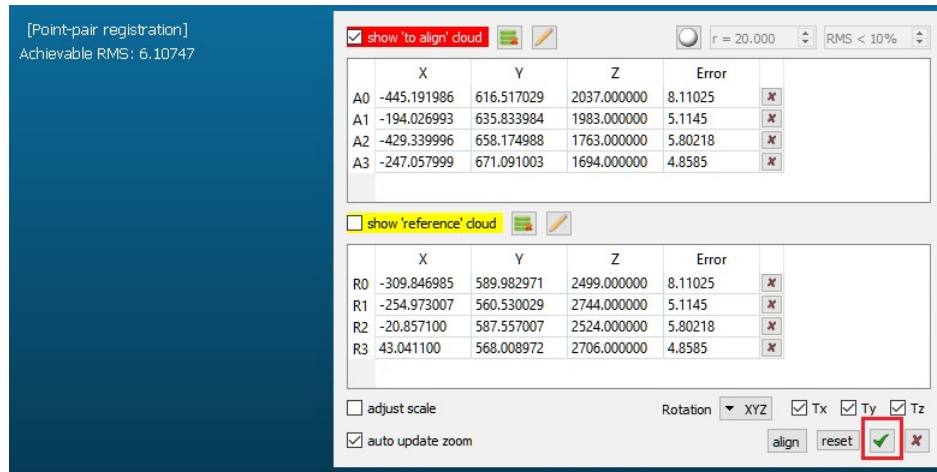


different perspectives. In order of point number, estimate the top of the marker by selecting it with a mouse click. If a prompt is displayed about estimation, hit yes. Keep in mind the distortion of the reflective marker. The top of the marker should be above and behind this reflective distortion. See imagery below for sample estimations.

Continue this process for all master points. Copy them to a text file, as you will need to refer to them later.

Now onto the to-align cloud. Hide the 'reference' cloud and show the 'to-align' cloud. Estimate the tops of the markers for this capture now. Refer to the drawing you made earlier, with the numbers of each point at different perspectives. If you make a mistake, you may delete the point, but ensure you do not delete the corresponding alignment point as well, or you will have to reselect points from the 'reference' cloud.

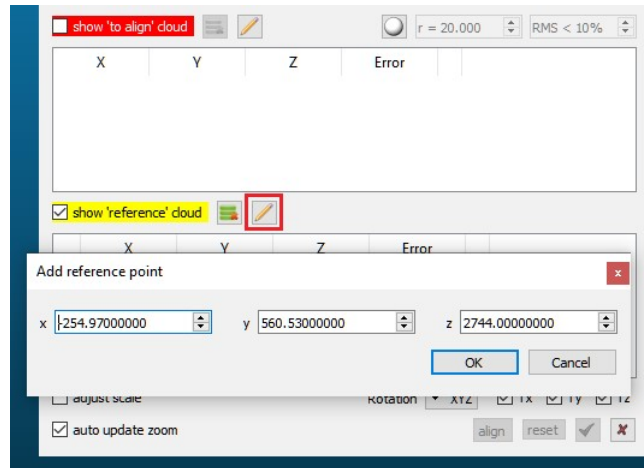
Once all points from the two captures are selected, note the 'Achievable RMS' number at the top of the screen. This is the estimated accuracy of the calibration between the two point clouds. It should be as low as possible, with an ideal value of < 25.0 . You may need to re-attempt this process if this number is too high.



Align the points by clicking the checkmark at the bottom. This will produce a 4x4 transformation matrix in the bottom console, which you can copy to your clipboard. Keep note of this matrix. This will be the transformation matrix for converting Sub1 points to the master coordinate system.

Now that you have the 4x4 transformation matrix, select the Sub1 cloud from the sidebar again. Navigate to [Edit (At the very top of the window bar) → Apply Transformation], then input the 4x4 matrix that was just calculated and click 'OK'. Manually verify the alignment. If it looks off, reload the Sub1 file and reattempt the point pair alignment process.

This process will need to be completed for all other, non-master captures, (Sub2 → Master, Sub3 → Master, etc.) When performing future point pair alignments, do not reselect the Master coordinates. Instead, manually enter them using the 'Pencil' option.



This concludes the initial calibration process using CloudCompare. Ensure you capture the 4x4 transformations once they are calculated, as they will be used to create a configuration file, covered in the next section. This calibration process must be performed for every camera configuration. If a camera is moved, or even slightly bumped, this process must be performed again. The accuracy of the calibration will be determined by the point selection precision. It is recommended to use the RMS calculation displayed for each alignment as a reference metric for improving your calibration technique.

B.4 Construct a Configuration File

Once the initial calibration is performed, and 4x4 transformation matrices are obtained for each Kinect capture, a configuration file will need to be created to relay this information to the program. This is done by creating a new text file and specifying the transformations to be applied to each capture, as denoted by its file suffix.

The format is as follows:

File Suffix.mkv

X X X X

X X X X

X X X X

X X X X

The triple pound/hashtag lets the program know that a transformation matrix follows. The file suffix with a .mkv extension is used to identify files from multiple different captures. This allows for captures to receive different file name identifiers, as long as they contain a common suffix for each calibration setup.

There are a few other options which can be specified in the configuration file.

Body tracking can be specified in the configuration file. This will enable captures to be isolated in space by using the Kinect body tracking joint locations to filter all points out except for the capture subject's body. A body tracking file suffix can be specified with a triple forward slash /// followed by a new line with the file suffix with the .mkv extension. This must match the identifier for the 4x4 transformation if it exists. Example:

///

Body Tracking File Suffix.mkv

Capture space bounds can be specified as well. This will determine what parts of the capture space are deemed to be valid outputs. If a body tracking file is specified (see above paragraphs for details), any frame with joint locations

outside of the defined capture space will not be processed nor output. If body tracking is not specified, only points within this capture space will be output for each frame. The capture space bounds can be configured with a triple dash – – – followed by a new line for each coordinate axis which will be bound, along with the minimum and maximum allowed values for that axis, space separated.

Example:

```
– – –  
X -9999 9999  
Y -9999 9999  
Z -9999 9999
```

Example Transformation File

Master.mkv

```
1.000000000000 0.000000000000 0.000000000000 0.000000000000  
0.000000000000 1.000000000000 0.000000000000 0.000000000000  
0.000000000000 0.000000000000 1.000000000000 0.000000000000  
0.000000000000 0.000000000000 0.000000000000 1.000000000000
```

Sub1.mkv

```
0.024707291275 0.124999701977 -0.991849064827 1669.198242187500  
-0.091880299151 0.988236606121 0.122255660594 -82.158508300781  
0.995463490486 0.088110782206 0.035901658237 1609.352416992188  
0.000000000000 0.000000000000 0.000000000000 1.000000000000
```

Sub2.mkv

```
-0.999605357647 -0.027915893123 0.003138161264 520.377258300781
```

-0.026642732322 0.977525353432 0.209127455950 -257.568237304688
-0.008905611932 0.208961308002 -0.977883338928 3434.870361328125
0.000000000000 0.000000000000 0.000000000000 1.000000000000

Sub3.mkv

-0.023229138926 -0.125729322433 0.991792619228 -1513.514526367188
0.065307550132 0.989750444889 0.127000033855 -151.239349365234
-0.997594773769 0.067721642554 -0.014779977500 1821.580322265625
0.000000000000 0.000000000000 0.000000000000 1.000000000000

///

Sub2.mkv

X -9999 9999

Y -9999 9999

Z -9999 9999

B.5 Secondary Calibration (Optional)

A secondary calibration process using an iterative closest point (ICP) algorithm can be performed to smooth out user error from the initial calibration process and ensure a more accurately aligned model. If your initial calibration is very accurate, this procedure may be skipped.

This procedure will be performed using CloudCompare.

Note: This method has only been tested using a 4 camera setup with 90 degree camera offsets. For other camera setups that do not have a front and back

capture, it may have to re-adapted, or may not function correctly at all.

Start by obtaining a capture with a human subject in the center of the capture space in a static pose, arms slightly out to the side.

The secondary calibration pipeline can be triggering by running the Kinect Azure Processor in calibration mode with a configuration file specified. This can be done using the `-c` (calibration) and `-t` (configuration file specification) arguments. Example:

```
./KinectAzureProcessor.exe "\\path\to\mkv\capture\folder"  
-c-t\path\to\config\file.txt
```

The output files for this will be created in the same folder as the original .mkv files. They will be filtered and transformed point clouds from the .mkv files for a single group frame (frame 15 if the `-f` argument is not specified). The file naming convention will be:

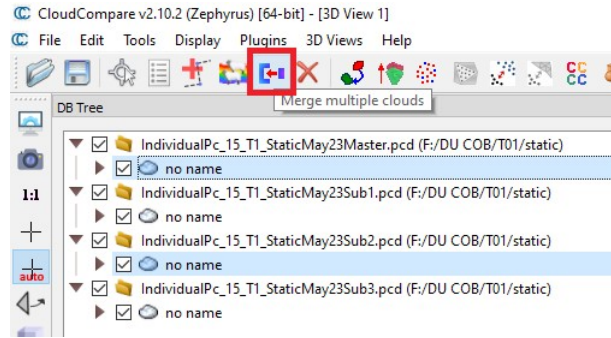
IndividualPc_GroupNumber_OriginalFileName.pcd

CloudCompare will be used for the rest of the calibration process.

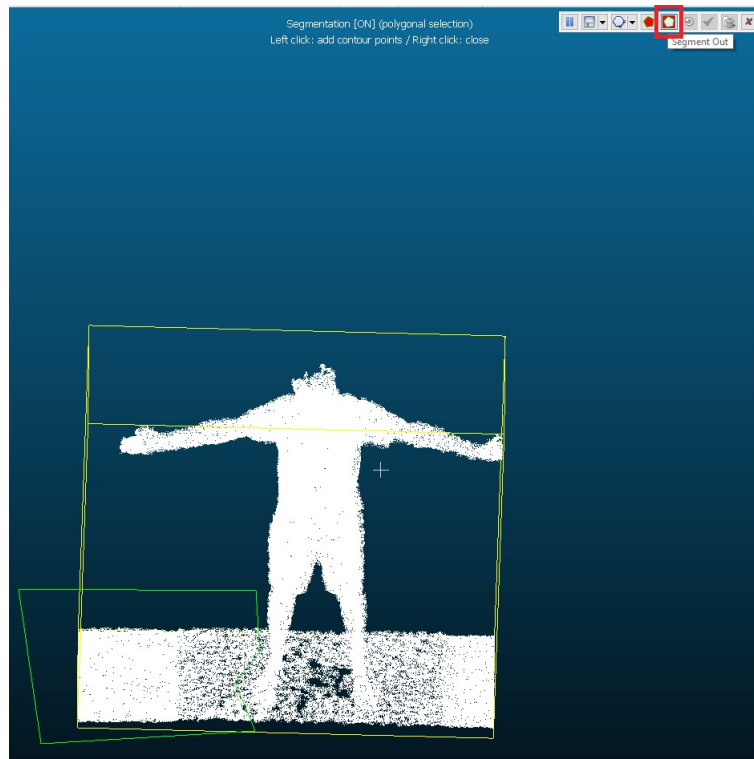
Start up CloudCompare and open up the newly output file (IndividualPc_15_xxx.pcd). Merge the front and back of the subject (Master and Sub2 in this example).

A window may popup asking about preserving the original cloud index. Whichever option you select will not matter, as we will not be preserving this merged cloud.

If the captures contain a layer of points on the floor, it would be best to



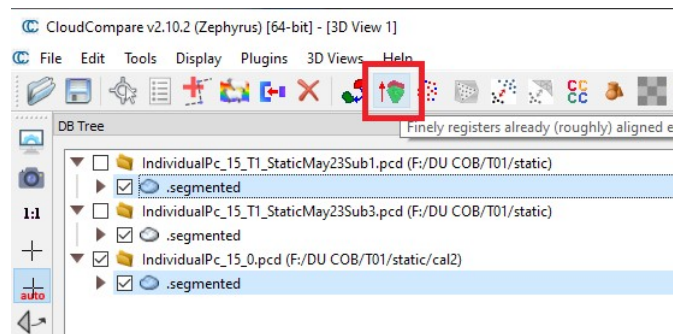
filter these out. Uncheck the list of captures on the left so only a single is showing at a time. To filter out the floor, select and display a single capture, then use the Edit → Segment option. Select the surrounding floor areas and segment them out. Hit the green checkbox in the top right once you are finished.



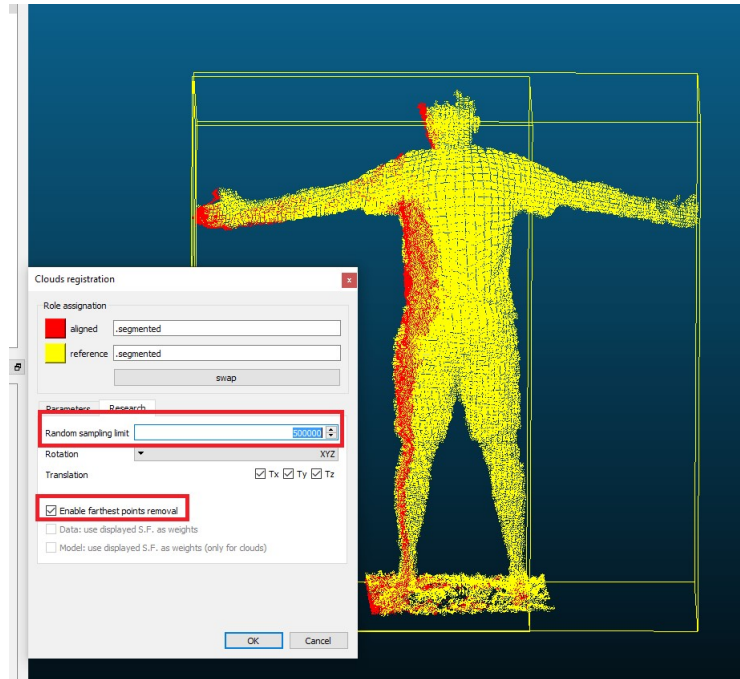
(Removing the floor from the capture)

Once the point cloud is segmented, it will show two inner options in the file picker tree, .remaining and .segmented. Click on the .remaining cloud and delete it, since it contains the floor points we just removed. Continue this process for each capture.

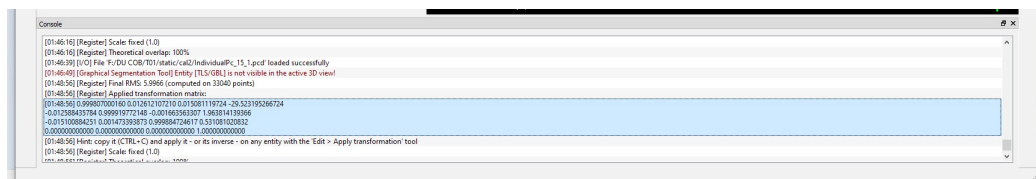
Next up is applying an ICP algorithm to the sides of the captures so that they more accurately align with the front and back of the body. Select the merged front/back capture as well as one of the sides, then hit the ICP logo in the top bar.



Ensure the reference cloud is the merged front/back capture, then go into the Research tab and increase the random sampling limit to 500,000 and Enable farthest point removal.



This will then generate a secondary transform with a calculated RMS. The RMS during the ICP should be relatively low (<10). A popup of the transform will appear. You can disregard and close this. A copy of the transformation will appear in the console which can be copied and pasted.



Continue this process with the other side capture, performing the ICP to merge it with the front/back capture then copying its transform output.

If extra accuracy is desired, an additional ICP may be applied to the front/

back capture that is not the master capture. This can be done by reloading both the front and back captures (and segmenting their floors again if they are

present), merging all but the non-master front/back capture, then applying an ICP algorithm with the same parameters as before (extra point sampling, enable farthest points removal) to align it to the merged capture.

Once this secondary calibration process is finished, you must update the transformation file so that it takes into account this secondary transform. To combine this updated transformation with the original, multiply the two matrices in this order: Updated * Original.

Once this is complete, the calibration process is complete and real capture processing can be started.

B.6 Step 4: Depth Data Processing

Supported Features:

The full functionality of the Kinect Azure Processor will transform, merge, filter, and mesh the depth data of the captures, as well as track and output joint positions if a body tracking file is specified. Output files will be placed in the same folder as the original mkv files. The output types are .pcd for point cloud outputs and .obj for mesh outputs.

For the full processing pipeline, the only arguments required are the folder of the .mkv files and the configuration file. Example:

```
./KinectAzureProcessor.exe "\\path\to\mkv\capture\folder"  
-t\path\to\config\file.txt"
```

Usage Guide (Command Line Arguments):

For latest usage guide, run

```
./KinectAzureProcessor.exe
```

or

```
./KinectAzureProcessor.exe -help
```