School of Computer Sciences

CAT405 Intelligent Computing Major Project

Final Report

Freemocap++: An AI-Based Multi-camera Markerless Motion Capture System with Joints Angle & REBA Score Calculation

ARYA DIVA PRADHANA

154439

Supervisor: Asc. Prof. Dr. Ahmad Sufril Azlan Mohamed

Examiner: Dr. Mohd Halim Mohd Noor

Academic Session

2023/2024

DECLARATION

“I declare that the following is my own work and does not contain any ***unacknowledged*** work from any other sources. This report was undertaken to fulfill the requirements of the Undergraduate Major Project for the Bachelor of Science in Computer Science (Honors) program at Universiti Sains Malaysia”.

Signature : arya ………………………………

Name : Arya Diva Pradhana

Date : 25 May 2024

ABSTRAK

Teknologi Motion Capture berdiri sebagai inovasi terobosan dalam grafik komputer, menggunakan aplikasi meluas merentasi pelbagai bidang seperti animasi komputer dan kejuruteraan. Walaupun teknologi tangkapan gerakan kontemporari sudah berleluasa, mereka kebanyakannya bergantung pada penanda pasif yang dilekatkan pada pelbagai bahagian badan subjek untuk memastikan gerakan yang lebih lancar dan konsisten dengan penggunaan *"marker"*. Walau bagaimanapun, kaedah ini memberikan cabaran yang wujud dan terbukti menyusahkan, mencetuskan keinginan yang bersungguh-sungguh untuk inovasi. Dalam projek ini, tumpuan beralih ke arah pelaksanaan pendekatan dipacu penglihatan komputer, berusaha untuk memperkenalkan tangkapan gerakan tanpa penanda dalam ruang tiga dimensi dengan penggunaan berbilang kamera. Inti dari inisiatif ini melibatkan memanfaatkan keupayaan tiga kamera yang diposisikan secara strategik, didorong oleh model *"Artifical Intelligence"* yang canggih. Matlamat utama adalah untuk menyediakan fungsi yang boleh mengira sudut sendi khusus subjek yang diperlukan untuk mengira skor postur REBA keseluruhan. Fasa seterusnya memerlukan pemprosesan data yang dikumpul dengan teliti, memudahkan algoritma dalam misinya untuk menentukan dengan tepat dan meramalkan setiap pergerakan bernuansa subjek dan mengira sudut dengan formula yang tepat. Melalui usaha ini, projek ini berhasrat bukan sahaja untuk menyediakan fungsi tambahan kepada sistem *Motion Capture* tetapi juga untuk mentakrifkan semula intipati bagaimana kami melihat maklumat tambahan yang diunjurkan dalam gerakan tiga dimensi dalam persekitaran yang dijana komputer.

ABSTRACT

Motion Capture technology stands as a groundbreaking innovation within computer graphics, wielding widespread applications across diverse fields such as computer animation and engineering. While contemporary motion capture technologies are already prevalent, they predominantly rely on passive markers affixed to various body parts of the subject to ensure a smoother and more consistent motion with the use of *"markers"*. However, this method presents inherent challenges and proves to be cumbersome, triggering a fervent desire for innovation. In this project, the focus shifts toward the implementation of a computer-vision-driven approach, striving to introduce markerless motion capture in a three-dimensional space with the use of multiple cameras. The crux of the initiative involves leveraging the capabilities of three strategically positioned cameras, driven by a sophisticated "*Artificial Intelligence*" model. The primary goal is to provide a functionality that can calculate the subjects' specific joints angle that is necessary to calculate the overall REBA posture score. The subsequent phase entails the meticulous processing of the gathered data, facilitating the algorithm in its mission to accurately pinpoint and predict the subject's every nuanced movement and calculate the angles with a precise formula. Through this endeavor, the project aspires not only to provide additional functionality to the motion capture system but also to redefine the very essence of how we perceived additional information projected in the three-dimensional motion in computer-generated environments.

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to all those who contributed to the successful completion of this report. First and foremost, I would like to express my sincere appreciation to Dr. Sufril as my FYP supervisor, whose guidance and unwavering support proved invaluable throughout the entirety of this project and for his expertise and insightful feedback that significantly enriched the content of this report. Furthermore, I extend my appreciation to Universiti Sains Malaysia for providing the necessary resources and facilitating an environment conducive to my research and development for my FYP. Finally, my sincere thanks to my colleagues and friends who provided encouragement and assistance during this journey. This collective effort has undoubtedly contributed to the fruition of this report, and I am grateful for the collaborative spirit that has shaped its outcome.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

CNNs = Convolutional Neural Networks

REBA = Rapid Entire Body Assessment

AI = Artificial Intelligence

OOP = Object Oriented Programming

Freemocap = Free Motion Capture

3D = 3 Dimension

1 INTRODUCTION

1.1. Background

Motion capture technology has assumed growing significance across various fields, including computer animation, engineering, and sports, owing to its capability to track and predict human movement. The conventional motion capture method typically involves the attachment of markers to specific body parts, with cameras capturing real-time movement data. In recent years, however, markerless motion capture has emerged as a popular alternative, utilizing cameras to record the subject's pose from specific angles.

Markerless motion capture offers a set of advantages and disadvantages. On the positive side, it eliminates the requirement for costly equipment such as markers, thus presenting a cost-effective solution. However, it is important to note that the accuracy of markerless motion capture generally falls short of traditional methods. This discrepancy is largely due to the nature of the data collected via cameras, which is in the form of images, making it somewhat challenging to extract and analyze pertinent information related to the subject's movement.

Notwithstanding its limitations, markerless motion capture has found numerous applications, particularly in the domains of sports and healthcare. For instance, researchers have leveraged markerless motion capture to dissect the movement patterns of athletes engaged in various sports activities, furnishing valuable insights into their performance, and contributing to injury prevention. Furthermore, within the realm of physical therapy, markerless motion capture has been instrumental in assessing patients' movements and formulating customized rehabilitation programs. Another promising application resides in the realm of virtual reality, where markerless motion capture technology holds the potential to facilitate realistic avatar movements, thereby enhancing the immersion and quality of gaming experiences.

One example of such software that uses multicamera detection is iPi soft. iPi Soft, a motion capture and animation software company, primarily utilizes markerless motion capture technology in its products. Markerless motion capture relies on computer vision algorithms and multiple cameras to track and analyze the motion of a subject without the need for physical markers or sensors. iPi Soft's software typically requires multiple cameras to capture a subject from different angles, and it employs advanced algorithms to process the data and reconstruct the subject's movements in a digital format.

iPi Soft's technology is designed to work with webcams or digital cameras, making it accessible for a wide range of users. The software is used in various industries, including animation, video game development, film production, sports analysis, and healthcare, where accurate motion capture is essential for creating realistic animations, evaluating athlete performance, or assisting in physical therapy.

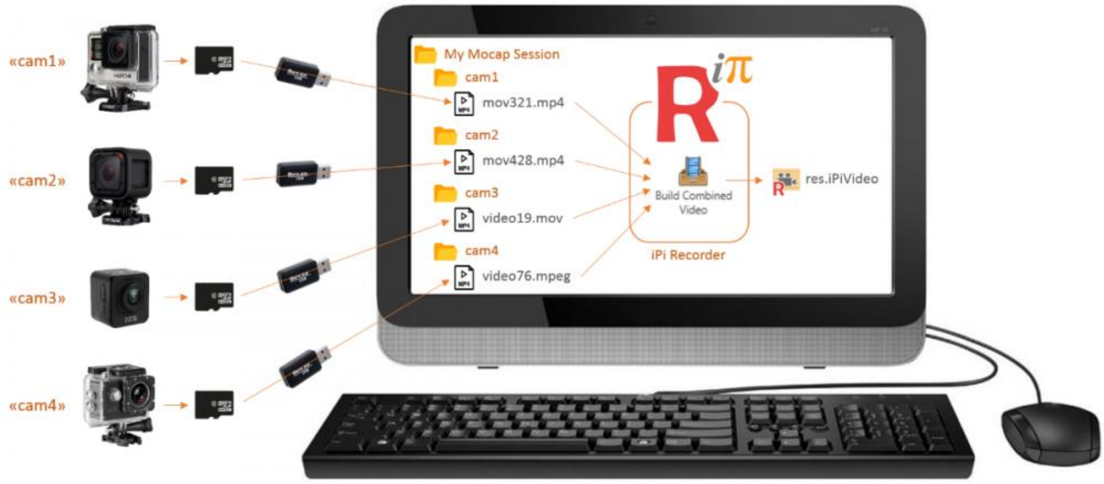


Figure 1; iPiSoft General Flow

src: Markerless Motion Capture. iPi Soft. (n.d.). (https://ipisoft.com/)

iPi Soft primarily uses computer vision and machine learning techniques in its motion capture technology. While iPi Soft's technology is not specifically focused on AI in the traditional sense, it employs advanced computer vision algorithms and machine learning methods to track and analyze human motion. iPi Soft's software uses computer vision techniques to process the image data captured by multiple cameras. Computer vision algorithms identify and track key points or features on the subject's body to reconstruct their movements in a 3D digital space. These algorithms are responsible for identifying and correlating the subject's body parts across camera views and for accurately capturing the motion, and machine learning methods may also be used to improve the accuracy and robustness of motion tracking. This can involve training algorithms to recognize and track body movements and poses in different scenarios and lighting conditions. Machine learning models can adapt and learn from the data to enhance the precision of motion capture.

Several Artificial Intelligence models such as CNNs and RNNs, leverage the power of neural networks to analyse, predict, and generate human motion, making them valuable tools in markerless motion capture for applications such as animation, sports analysis, and behavioural tracking. The choice of model often depends on the specific requirements and nature of the motion capture task.

1.2. Problem Statements

The existing motion capture model's limitation of relying on just two to three cameras for capturing subject motion raises concerns about its accuracy and tracking alignment. This limitation is exacerbated when compared to traditional marker-based methods. To address this issue and enhance the usability of motion capture data, an investigation into displaying specific joint angles from the existing data using an existing open-source motion capture system is imperative.

The markerless motion capture system dependency on a limited number of cameras for motion capture leads to potential inaccuracies and tracking misalignments, especially when contrasted with marker-based techniques. Instead of increasing the number of cameras, extracting and displaying joint angles from the existing data can provide valuable insights and improve the interpretation of motion capture results. This approach involves refining the data processing algorithms to accurately calculate and present joint angles, thereby enhancing the overall utility of the motion capture system.

By focusing on the detailed analysis of joint angles from the existing three-camera setup, we can use this functionality to help analyze the overall REBA posture score. This method leverages the existing infrastructure while aiming to deliver accurate joint angle measurements, contributing to improved motion capture accuracy and reliability in comparison to traditional marker-based methods.

1.3. Motivation

The primary motivation behind my choice of this project originates from a deep-seated desire to enhance and innovate within the field of motion capture technology. Specifically, I aim to augment an existing motion capture system by introducing a new functionality that focuses on the accurate display of specific joint angles. This enhancement is not merely an incremental improvement but a significant addition that can provide substantial benefits in various practical scenarios. By undertaking this project, I aspire to achieve my personal objective of pushing the boundaries of current motion capture capabilities. Furthermore, I am committed to contributing valuable advancements to the communities and industries that rely on this technology. This project represents a meaningful opportunity to address specific needs, deliver practical solutions, and ultimately support the advancement of motion capture systems for diverse applications.

1.4. System Objective

1. **Implement Data Processing Capabilities**
   - Develop a new algorithm to accurately calculate joint angles and REBA posture score from existing data.
2. **Frame by Frame Data Visualization**
   - Integrate a post processed frame-by-frame visualization module to display data such as the joint angles & REBA score during motion capture sessions.
3. **Integration with Existing Systems**
   - Ensure seamless integration of the new functionality with the current motion capture system without compromising existing features.

1.5. Proposed Solutions

To achieve the outlined system objectives for enhancing and extending the functionalities of the existing open-source multi-camera markerless motion capture system *‘’Freemocap”* by displaying specific set of joint angles and its REBA posture score, the following solutions are proposed:

To accurately calculate joint angles from the existing multi-camera data, I propose the development and implementation of data processing algorithms. These algorithms will leverage the use of precise mathematical formulas to ensure precise joint identification, angle calculation, and REBA posture score by using the data produced from the existing system.

To enable post processed frame-by-frame visualization of the newly processed data during motion capture sessions, I propose integrating a dedicated visualization module/method into the existing system. This module will feature a user-friendly interface designed for intuitive interaction, allowing users to view and monitor specific joints dynamically. The post processed frame-by-frame aspect will be facilitated by applying the mathematical formulas for each frame and display it frame-by-frame in the same module from an external .csv file, ensuring that the data is displayed promptly as the necessary data is acquired. This functionality will significantly enhance the usability of the system, providing immediate feedback and enabling a quick adjustment during motion capture sessions.

To ensure seamless integration of the new joint angle display functionality and the REBA posture score with the current motion capture system, I will maintain compatibility with existing infrastructure and adopt a modular software design. This functionality will be developed as an independent module that interacts with the main system through well-defined classes and methods, enabling smooth data exchange. Robust module & data synchronization methods will ensure smooth synchronization among data input & output.

1.6. Benefits and Uniqueness of the Proposed Solutions

By developing and implementing algorithms for accurate joint angle and REBA posture score calculation, I significantly improve the precision of motion capture data. This addition not only boosts the accuracy of motion capture but also extends the system's applicability to more precise and demanding use cases.

The proposed solution is an innovation of an existing project with the same goal as a previous model. This project is based on a Three-camera motion capture assisted with an improved Artificial Intelligence model with the CNN architecture as its base deep learning model. The latest model version of this project uses Two-cameras and an outdated algorithm to compute the subject pose and motion, but now by using Four cameras and an additional improvement to the whole algorithm complete with more reliable data, we can expect an increase in its accuracy and performance.

Integrating a post processed frame-by-frame visualization module transforms the user experience by providing immediate information after the motion capture sessions. This feature allows users to dynamically frame-by-frame view specific joints angle data and the subject's REBA posture score, facilitating more effective and efficient motion analysis. The simple interface ensures that this functionality is easily accessible to users with varying levels of technical expertise.

The modular design of the new functionality ensures seamless integration with the existing motion capture system. Maintaining compatibility with current infrastructure and data formats minimizes disruption and leverages existing investments, allowing users to quickly adapt to and benefit from the enhanced system capabilities.

These proposed solutions collectively provide a robust, precise, and user-friendly enhancement to the existing motion capture system, setting it apart from traditional methods and offering additional data and versatility.

1.7. Organization of the Report

This report is structured into six primary sections, which are: Introduction, Background & Related Works, System Requirements & Design, System Design & Implementation, System Testing & Evaluation and Conclusion & Future Works.

The introductory segment offers a project overview that includes background details, problem statements, objectives, benefits, and a concise introduction to the proposed solution. It functions as a guide, aiding readers in grasping the contextual framework of the project.

In the second chapter, the background and related works of the project were elaborated, providing an in-depth examination of pertinent projects. The chapter delves into existing models and related deep learning architecture, discusses their strengths and weaknesses, and introduces the proposed work as a segue into the subsequent section.

The third chapter provides an analysis of the overall multi-camera markerless motion-capture system, beginning with an elucidation of the project scope. This chapter covers system capabilities, limitations, project management, development methodology, system analysis, and the deployed technology of the system. Moreover, it delineates the system design, encompassing a system architecture diagram that illustrates hardware communication and a comprehensive flow chart detailing the proposed solution.

The fourth chapter features a more detailed system design explanation and each module component elaboration in detail. Additionally, it will also describe the method/approach used to develop the system in depth so that readers can gain a more comprehensive understanding of the improved system.

The fifth chapter will explain the testing strategy used to test the system, test cases and scenarios, summary of test results, evidence that the system meets the requirements and objectives, and evaluation of the works. In essence, this chapter will show the output of various scenarios in different test cases.

The concluding chapter serves as the end and concludes the analysis report, complete with references and an appendix section following at the end to provide more information. In addition, it will also suggest further work and potential for the project.

2 BACKGROUND & RELATED WORK

2.1. Status of The Project

The current project is based on an existing open source markerless motion capture system called ‘*Freemocap’*. It is built in 2021 and is still in development by a group of developers based on the United States as of now.

The current project is currently in its finalization phase and has met most of the expected outcome by extending the functionality of the existing system and is expected to be finished in a few weeks.

2.2. Background

Traditional motion-capture techniques often rely on the use of *markers*. These markers often come as pairs with a black suit and several small balls/cues attached to the suit key point of the body thus introducing complexities in terms of preparation, calibration, and constraints on natural movement. In response to these challenges, the markerless four-camera motion-capture system emerges as a promising avenue for enhanced accuracy and freedom of movement. There are currently no projects or systems that are similar to the one that I am developing, since it is a unique module addition to the existing open-source motion capture system ‘*Freemocap’*.

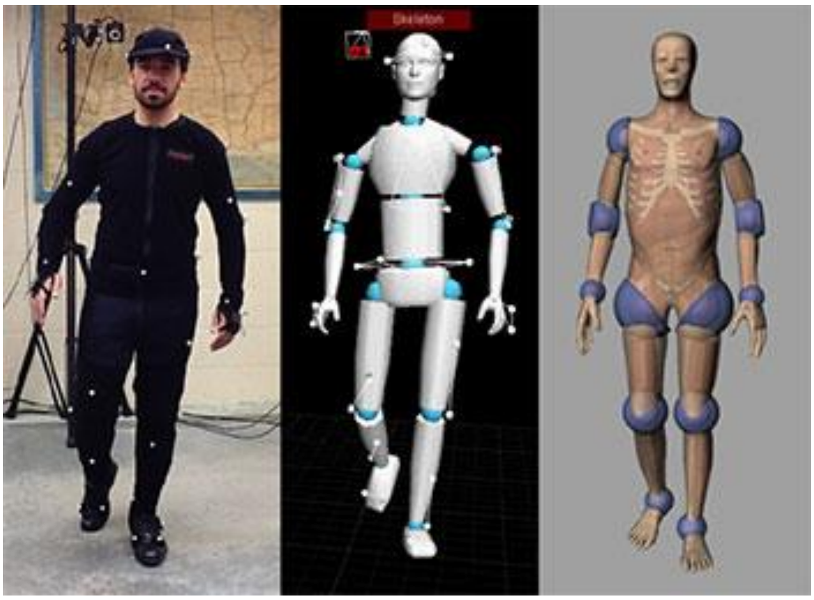


Figure 2; Image of motion capture suit with markers.

Deep learning models have been increasingly used in markerless motion capture to accurately track and predict human motion. Here are several existing CNN-based deep learning architectures used in motion capture & pose estimation based on many sources:

1. **Region-Based Convolutional Neural Network (R-CNN):**
   R-CNN is a type of CNNs model. The architecture uses the selective search algorithm that generates approximately 2000 region proposals. These 2000 region proposals are then provided to CNN architecture that computes CNN features. These features are then passed in an SVM model to classify the object present in the region proposal.
2. **Convolutional Neural Networks (CNNs):**
   CNNs is a type of deep learning model particularly well-suited for processing grid-like data, such as images. It consists of multiple layers, including convolutional layers that apply filters to input data to detect features, pooling layers that reduce dimensionality, and fully connected layers that perform classification. CNNs are highly effective for tasks like image recognition, object detection, and pattern recognition due to their ability to automatically learn hierarchical feature representations.
3. **Generative Adversarial Networks (GANs):**
   GANs can be used for generating realistic human motion sequences. These models can learn and generate motion patterns, making them valuable for creating lifelike animations and for data augmentation in motion capture.

While the algorithm employed by numerous studies exhibits adequacy for the development of a markerless motion capture system, insufficient data poses a considerable impediment, thereby compromising the efficacy, precision, and fluidity of the tracking process in a sub-optimal way. This underscores the rationale behind augmenting the system with additional cameras to enhance both accuracy and smoothness in motion tracking. Concomitantly, the introduction of a novel algorithm is proposed to synchronize data emanating from disparate cameras, each strategically positioned at distinct angles encompassing the subject. However, a noteworthy challenge is discerned in the synchronization process, as each frame necessitates capturing identical motions across all cameras to render the data usable. This intricacy introduces multifaceted challenges to the synchronization endeavour.

This project will introduce how multiple cameras affected the accuracy and smoothness of a subject motion without the use of any markers with the assistance of an Artificial Intelligence model and show additional data to help visualize new information alongside the motion capture results. This program will process and synchronize all the footage from multiple cameras and use that footage as data in a form of pre-recorded videos and will then be processed by the AI model to track the subject movement frame by frame in a 3D space as a connected landmarks/stickman figure. Additionally, my module will process the 3D coordinates of the captured joints for each frame and display the joints angle and REBA posture score for each frame by calculating it through mathematical formulas.

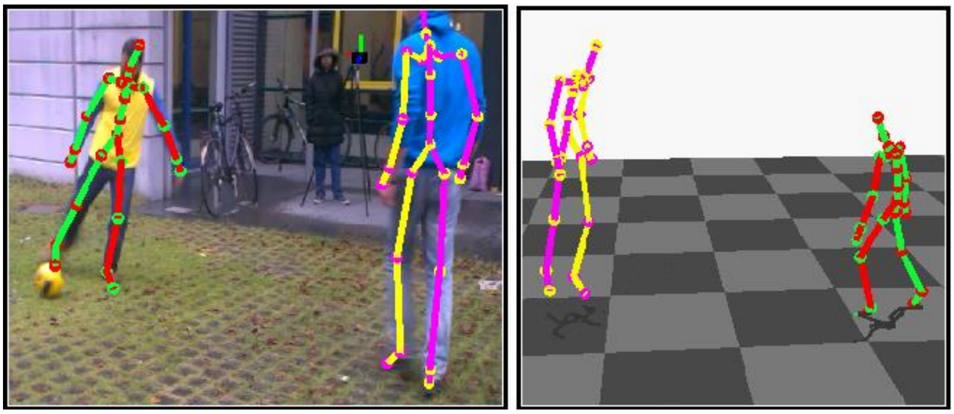


Figure 3; General Pose Estimation Preview in a 3D Space.

2.3. Related Work

While several existing markerless motion-capture have already existed, most of them only use a single camera as its only source of data, and there are only several few that use multiple cameras. Of all of these existing systems, motion capture software like IpiSoft and FreeMoCap have successfully utilized and improved the markerless motion-capture system using their own method and algorithm. These projects will serve as a valuable reference in developing my own system.

2.3.1. iPiSoft

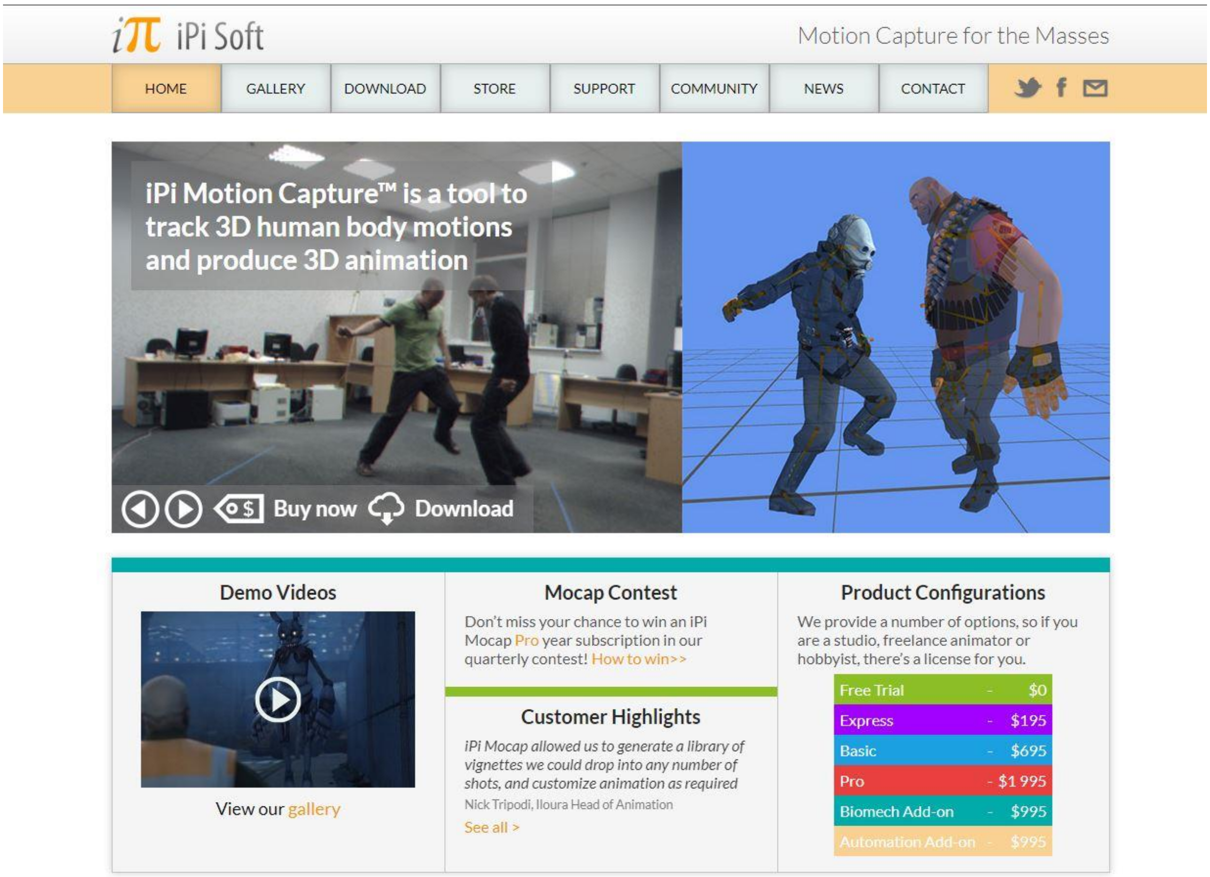


Figure 4; IpiSoft Homepage

"iPi Motion Capture™ is a scalable markerless motion capture software tool that supports 1 to 4 depth sensors or 3 to 16 RGB cameras (Sony PS3 Eye / Logitech C922 / action cameras) to track 3D human body motions, transfer motion to 3D character and produce 3D animation."

(src: https://ipisoft.com/pr/iPi_Motion_Capture_Brochure.pdf)

2.3.2. FreeMoCap Project

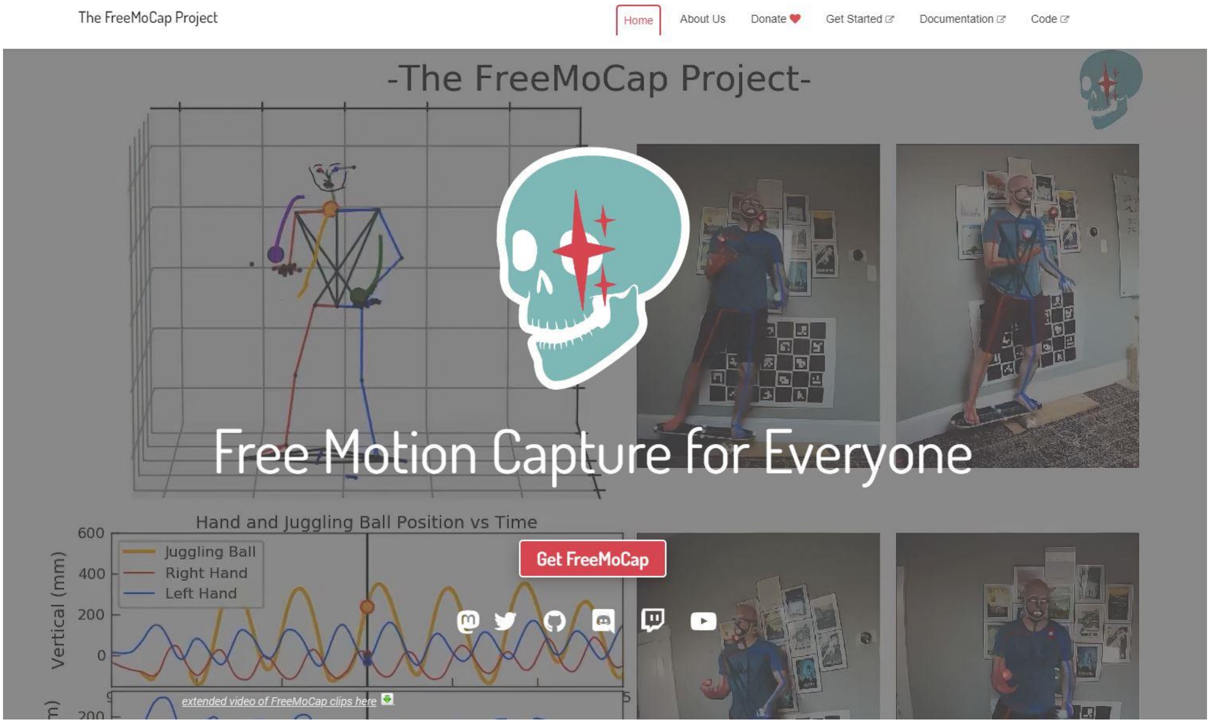


Figure 5; FreeMoCap Project Homepage

“FreeMoCap (Free Motion Capture) is a free open-source markerless motion capture system designed to provide research-quality motion capture data using free software and generic, minimal-cost webcams. The data it provides can be useful for any project that would benefit from high quality 3D measurements of human movement, including scientific research, 3D animation, sports biomechanics, and more.”

This software is also the one I aim to extend its functionality as the main objective for my project since it is an open-source project available in its public github repository.

(src: https://freemocap.github.io/documentation/about_us/)

2.4. Strength & Weakness of Existing System

Table 1; Strength & Weakness Table of Existing System

Existing System	Strengths	Weaknesses
iPiSoft	• The software is designed to be user-friendly. • iPiSoft supports multiple cameras, allowing users to set up a multi-camera system for more accurate and detailed motion capture. • It provides options for exporting motion capture data in standard formats (e.g., FBX), making it compatible with various animation and 3D modeling software.	• Achieving optimal results may depend on having a sufficiently powerful computer and a setup with multiple cameras, which could be a limitation for users with less powerful hardware. • Sensitive to lighting conditions. Changes in lighting may affect the accuracy of motion capture results. • The price may seem too expensive for some people.
FreeMoCap Project	• It is an Open-Source Project, thus makes it accessible to a wide range of users. • FreeMoCap is designed to work with low-cost hardware, making it more accessible for users who do not have access to high-end kit.	• FreeMoCap's performance might be limited by the capabilities of the hardware it supports. • The project's development and updates depend on community engagement, thus may become outdated if there’s a lack of engagement.

3 SYSTEM REQUIREMENTS / ANALYSIS

3.1. Project Scope

- **Markerless Motion Capture**

  The primary goal is to develop a motion capture system that tracks human movement without the need for physical markers. The system should utilize computer vision algorithms to analyze video input from multiple cameras.

- **Multi-Camera Setup (3 cameras)**

  The system will involve the integration and synchronization of data from three cameras placed at different angles. This allows for a comprehensive capture of the subject's movement from various perspectives.

- **Post-processing Motion Capture process**

  The system should be able to perform post-processed since the algorithm will need the recorded footage as data to be able to track the subject’s pose & motion.

- **Algorithm Enhancement**

  Develop and refine algorithms to accurately calculate both the joint angles and the overall REBA posture score from the existing multi-camera motion capture data.

- **Integration with Existing Systems**

  Ensure the new functionality and modules integrate seamlessly with the existing motion capture system without requiring extensive modifications.

3.2. System Capabilities

- **Multi-Camera Pose Estimation**

  The system can simultaneously capture and estimate the pose of a subject from a single camera viewpoint and use the data from the other cameras to maximize the accuracy and precision of the motion tracking.

- **Post-processing Feedback**

  Provide post-processed feedback on the subject's movement, allowing for analysis and adjustments after processing the data.

- **Calculate Additional Data with Accuracy**
  Provide an accurate calculation of the subject's specific joint angles with a mathematical formula and estimate the subject's overall REBA posture.
- **User-Friendly Interface**
  Implement a user-friendly interface to ease accessibility to various users regardless of skill level. Additionally, it will also show additional data with minimal user interaction.

3.3. System Limitations

- **Computational Requirements**
  The system may have high computational demands, especially when processing data from multiple cameras in real-time. This could limit deployment of low-spec hardware and may affect the speed of the whole process.
- **Space and Setup Constraints**
  The accuracy of the motion capture may be affected by the available space and the specific setup of the cameras. Limited space might restrict the range of movements that can be captured effectively, and the number of cameras may also hinder the process.
- **Subject Identification Challenges**
  The system may face challenges in accurately identifying and tracking the subject within the field of view, particularly if they overlap or if there are rapid movements. So green screens or unicolor clothes may ease the identification process.
- **Limited Field of View**
  Depending on the camera placement, there may be blind spots or limitations in the field of view, impacting the system's ability to capture certain movements or poses.
- **Cost and Accessibility**
  The cost of the hardware, cameras, and computing resources might limit the accessibility of the system to certain users or applications.

3.4. Project Management

3.4.1. Work Breakdown Structure Diagram

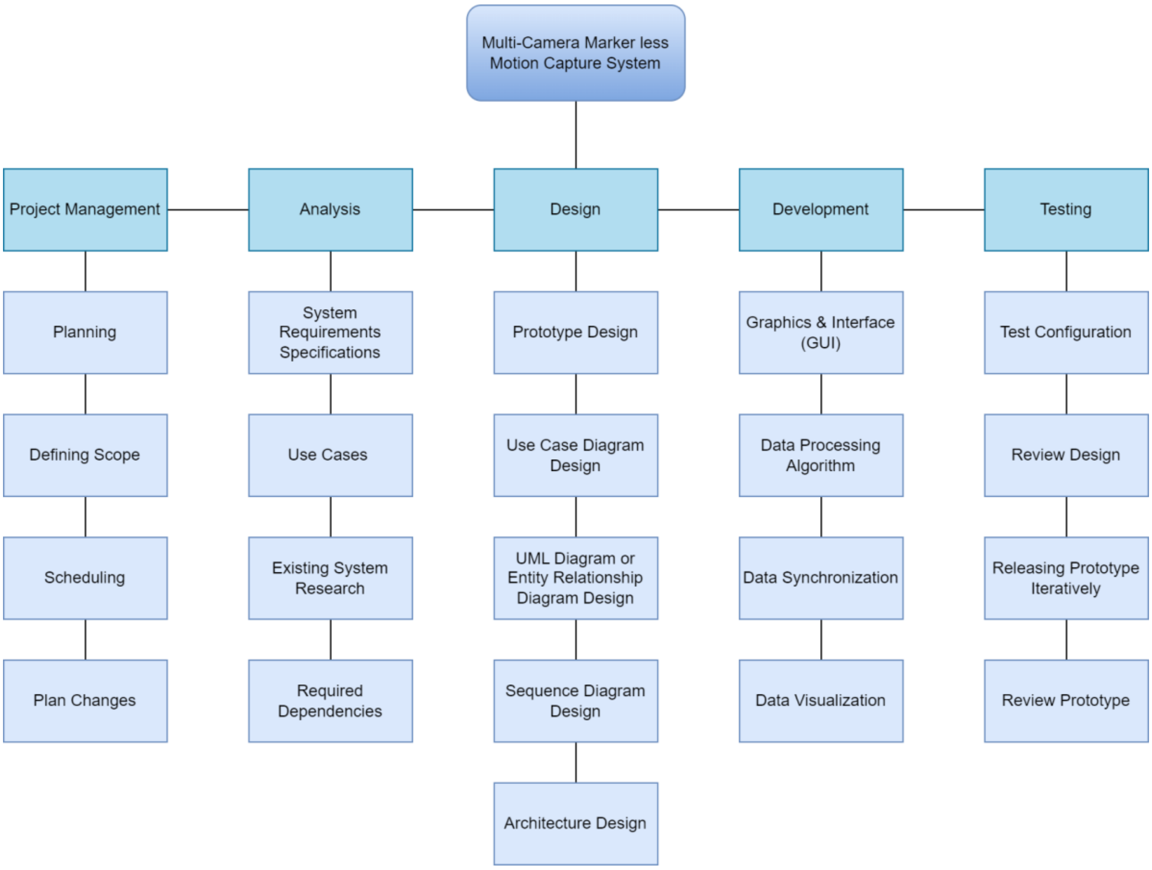


Figure 6; Work Breakdown Structure

3.4.2. Split Gantt Chart

	Task Name	Duration	Start	ETA
1	Project Management	8 Week	1 November	30 November
2	Planning	5 Week	Week 1	Week 2
3	Defining Scope	2 Week	Week 3	Week 3
4	Scheduling	2 Week	Week 3	Week 3
5	Analysis	4 Week	1 December	31 December
6	System Requirements Specifications	2 Week	Week 1	Week 2
7	Use Cases	2 Week	Week 3	Week 4
8	Existing System Research	4 Week	Week 1	Week 4
9	Required Dependencies	2 Week	Week 4	Week 4
10	Design	6 Week	1 December	15 January
11	Prototype Design	2 Week	Week 1	Week 2
12	Use Case Diagram Design	2 Week	Week 2	Week 3
13	UML Diagram or Entity Relationship Diagram Design	4 Week	Week 2	Week 4
14	Sequence Diagram Design	2 Week	Week 5	Week 6
15	Architecture Design	1 Week	Week 6	Week 6
16	Development	20 Week	15 December	1 May
17	Graphics & Interface (GUI)	7 Week	Week 1	Week 7
18	Cameras Synchronization	6 Week	Week 7	Week 13
19	Subject Recognition & Pose Estimation	3 Week	Week 9	Week 11
20	Data Processing	3 Week	Week 9	Week 11
21	Tracking Algorithm	12 Week	Week 5	Week 16
22	Render Tracking and Display Output	2 Week	Week 16	Week 17
23	Testing	4 Week	1 May	25 May
24	Test Configuration	2 Week	Week 1	Week 2
25	Review Design	1 Week	Week 2	Week 2
26	Review Prototype	1 Week	Week 3	Week 3

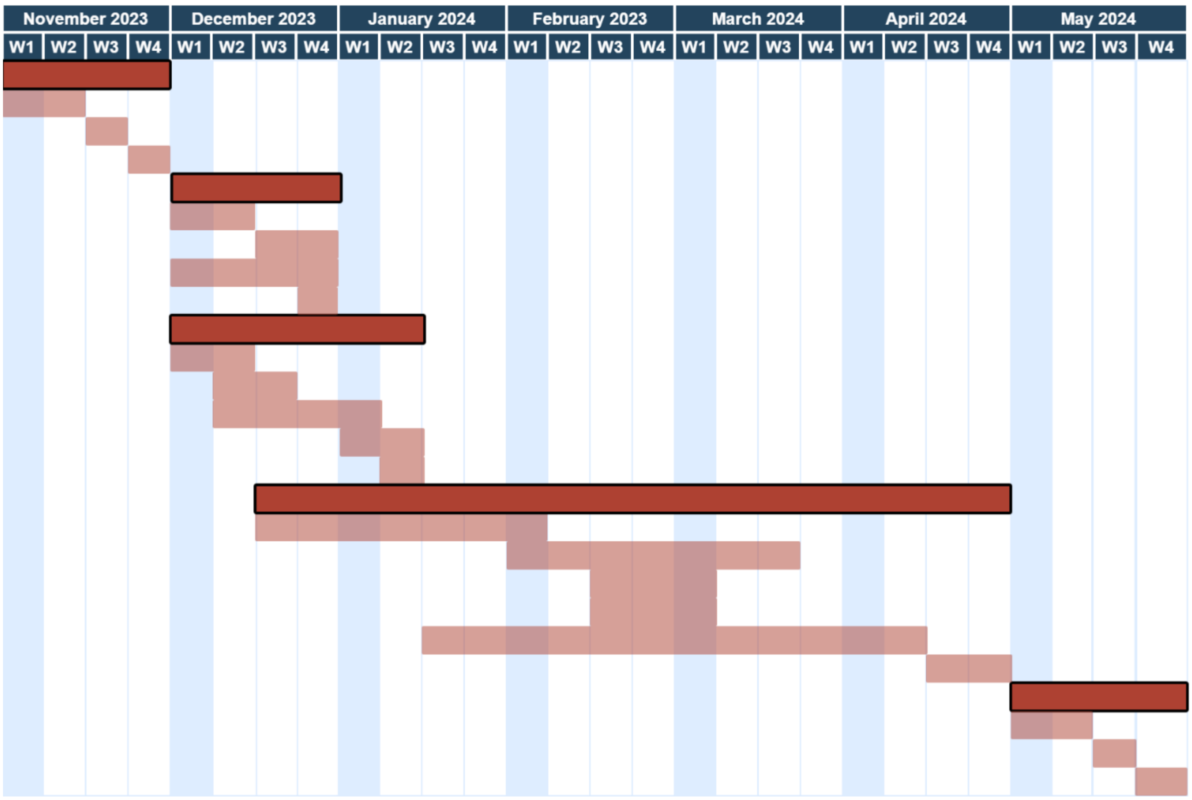


Figure 7; Split Gantt Chart

3.4.3. Swot Analysis

Strengths	Weaknesses
- Relatively High Precision Pose Estimation. - Markerless Motion Capture Method/Non-Invasive. - Reduced Setup Time. - Quick Joint Angles & REBA Calculation.	- Complex Calibration. - Computational Intensity. - Limited Tracking Range. - Sensitivity to Lighting Condition.
Opportunities	**Threats**
- AI Integration & Innovation. - Various Applications. - Simulations & Analysis. - Other Technologies Integration.	- Competition from Marker-based System. - Developers & Community Dependent. - Technological Obsolescence.

Figure 8; SWOT Table

3.5. Development Methodology

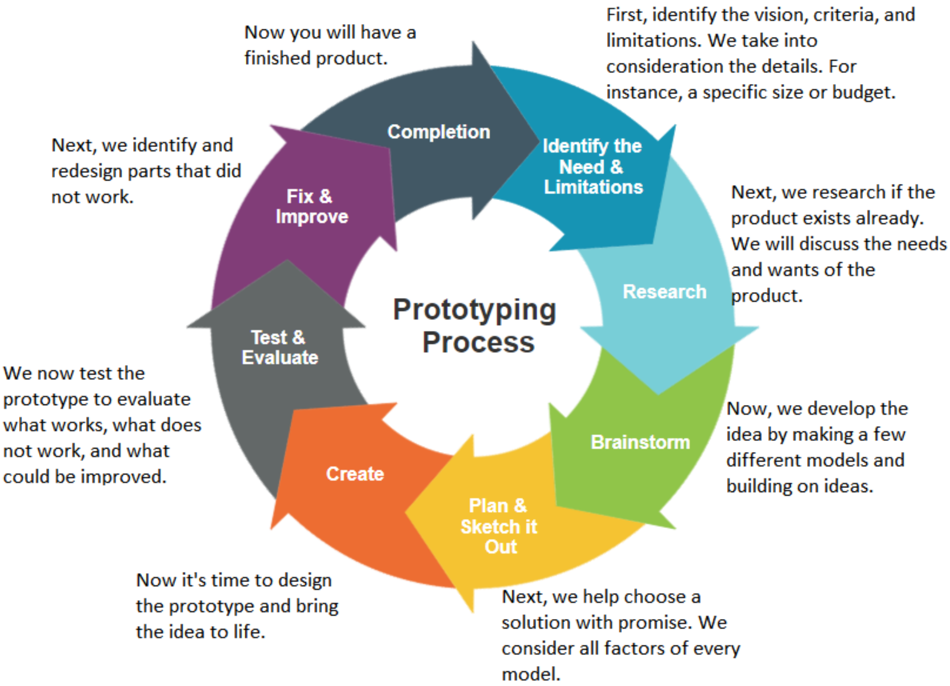


Figure 9; Prototyping Process

Src: (https://www.teraengineering.com/prototyping-process/)

The proposed system is an implementation of Computer Vision, choosing the **Prototyping Development Methodology** seems suitable due to the absence of developer-client relationships. In addition, this project will not require frequent feedback and validations from other parties thus minimizing the overall time to develop, since the training & test data can be reused in the form of recorded footage and thus this will allow for a faster progress in development.

The iterative nature of prototyping allows for faster development cycles, with short feedback loops enabling swift adjustments and reducing development time. This approach fosters design validation, and accommodates changes efficiently and iteratively, ultimately leading to a more adaptable and cost-effective development process. In summary, prototyping stands as a suitable methodology for this project, aligning well with the dynamic nature of this system development.

3.6. Detailed Requirements

3.6.1. Functional Requirements

- **Calibration**
  The system should support an accurate calibration process for each camera for each session to ensure precise spatial mapping.
- **Post-Processing Motion Tracking**
  The system should be capable of post-processed tracking of the subject's overall movement based on pre-recorded videos.
- **Subject Recognition & Calibration**
  The system should be able to successfully identify and differentiate each subject pose in every frame within the capture area.
- **Multi-Camera Synchronization**
  The system should ensure synchronization among multiple cameras to capture seamless motion across different viewpoints.
- **Additional Data Processing**
  The system should be able to correctly calculate the additional data (Joint angles & REBA posture score) from the existing data and display it with visuals.
- **File & Data Synchronization**
  External files and data that are generated and used by the system should be synchronized for each session and ensure data availability.

3.6.2. Non-Functional Requirements

- **Performance**
  The overall process should be performed within a reasonable time and not exceeding hours. Additionally, the additional data processing/calculation should be performed instantaneously right after the motion capture process.
- **Scalability**
  The system scalability should extend to the size of the capture area, allowing the system to handle larger spaces without significant degradation in performance.

- **Reliability**

  Robust error recovery mechanisms should be in place to handle and recover from unexpected errors or disruptions during the motion capture process. Such as error message or error logs.

- **Usability**

  The user interface should be intuitive, user-friendly, and require minimal training for operators to configure and operate the system effectively.

- **Additional Data Availability**

  The system should provide a separate additional external file located in each session folder. The external file will contain the numbers of joint angles data and REBA posture score for users to have the option to analyze it directly.

3.7. System Analysis

a) Module Diagram

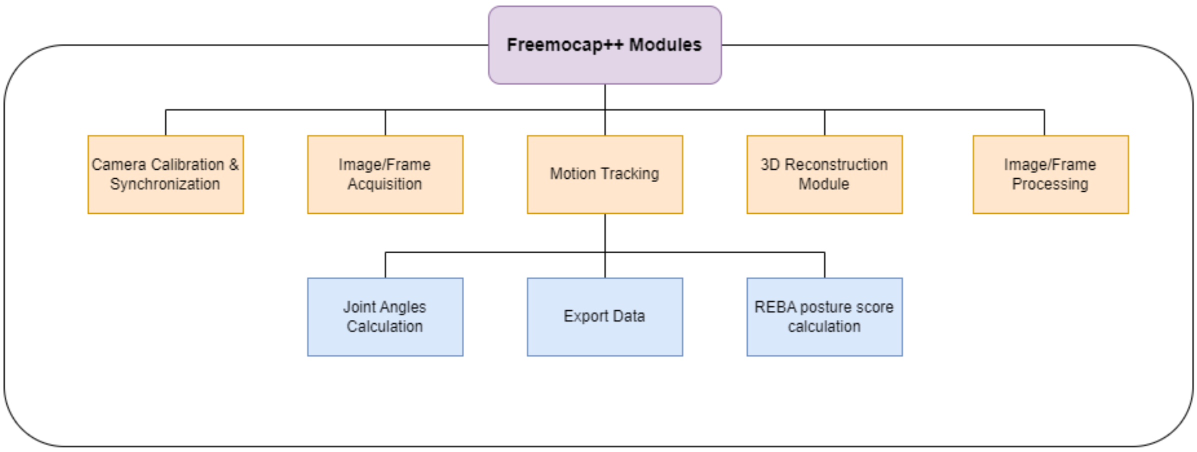


Figure 10; Updated System Module Diagram

❖ Existing Modules

- **Camera Calibration & Synchronization**

  Responsible for calibrating multiple cameras to ensure accurate spatial relationships between them using an external .toml file.

- **Image/Frame Acquisition**

  Functions by extracting each frame of the video/footage as images to be processed for later use.

- **Motion Tracking**

Integrates the pose data from each camera frame by frame, to track the motion of subjects in a coordinated and synchronized manner as well as estimate the pose of the subjects using Blazepose/CNNs.

- **3D Pose Estimation**

  To create and visualize the subject skeleton in a 3D space by using the Anipose library based on the motion capture process data.

- **Image/Frame Processing**

  Works alongside the Motion Tracking module by detecting and classifying specific features/body part for each picture/frame in the video.

❖ Additional Modules

- **Joint Angles Calculation**

  Responsible for calculating the joint angles of specific joints using a mathematical formula from the motion capture data.

- **Export Data**

  To export the result of the calculations to an external .csv file for each session.

- **REBA Posture Calculation**

  Responsible for calculating the overall REBA posture score of the subjects by making use of the joint angles results.

b) Use Case Diagram

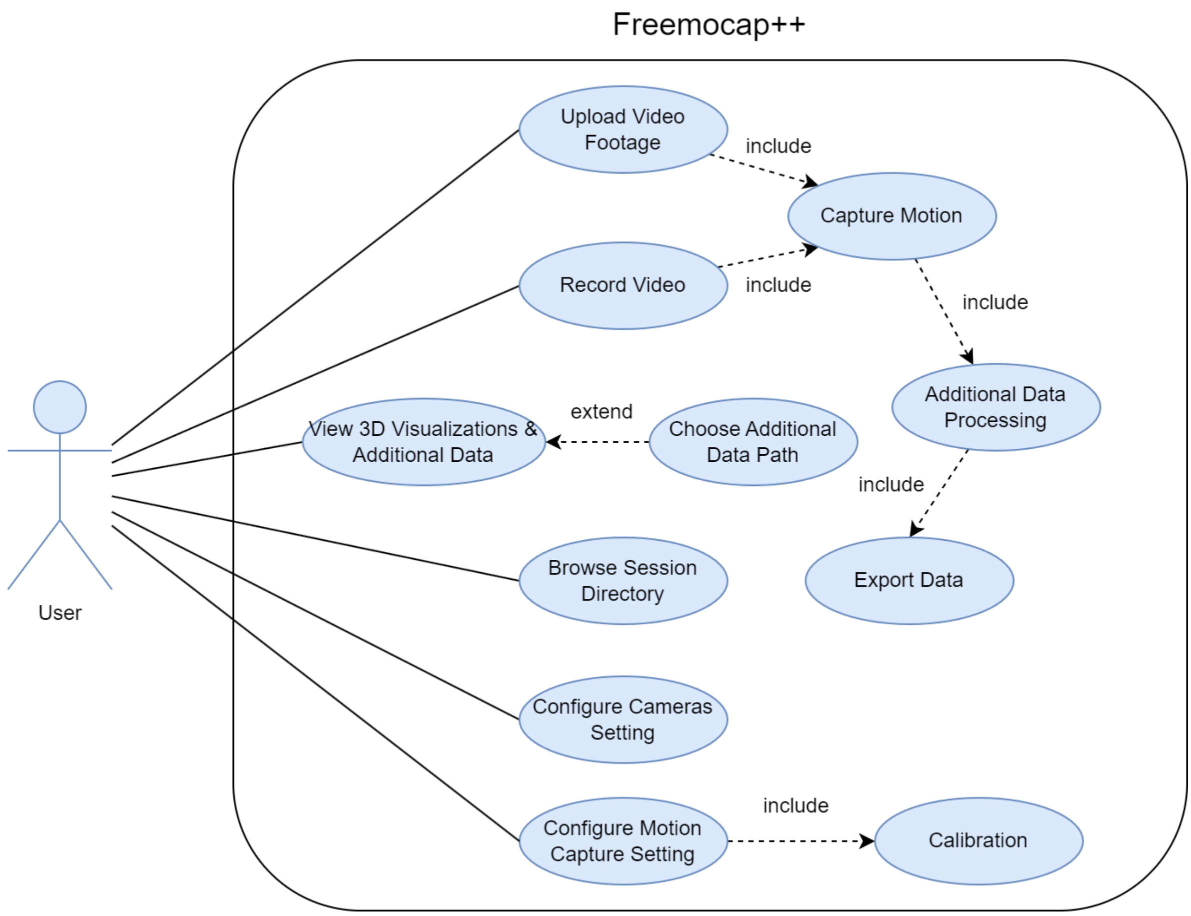


Figure 11; Updated Use Case Diagram

c) Use Case Description (Updated)

Table 2; Use Case 'Upload Video Recordings'

<table>
<tr><th>Use Case Name</th><th colspan="2">Upload Video Recordings</th></tr>
<tr><td>Scenario:</td><td colspan="2">A new user wants to upload their separate footage into the program.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">User clicks on the ‘load recordings’ or ‘import videos’ button on the home screen.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">User can choose their separate recording via the use of these buttons by opening file explorer to locate your recordings.</td></tr>
<tr><td>Actors:</td><td colspan="2">User</td></tr>
<tr><td>Related Use Cases:</td><td colspan="2">Capture Motion</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">User</td></tr>
<tr><td>Preconditions:</td><td colspan="2">User has initiated the program.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">User successfully uploaded their footage onto the program without errors.</td></tr>
<tr><td rowspan="2">Flow of Activities</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Click on either the ‘load recordings’ or ‘import videos’ button.
2. Choose the location of the recordings in file explorer.</td><td>1. Verify uploaded footage.
2. Checks for the correct format.
3. Check whether it is synchronized.
4. Prepare the calibration file.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">The recordings are un-synchronized if it is with multiple cameras and the calibration file is not the correct one.</td></tr>
</table>

Table 3; Use Case 'Record Video'

<table>
<tr><th>Use Case Name</th><th colspan="2">Record Video</th></tr>
<tr><td>Scenario:</td><td colspan="2">A new user wants to record the videos using the system itself.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">User clicks on the ‘new recordings’ button on the home screen.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">User can then record themselves using the system with either a single or multiple cameras.</td></tr>
<tr><td>Actors:</td><td colspan="2">User</td></tr>
<tr><td>Related Use Cases:</td><td colspan="2">Capture Motion</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">User</td></tr>
<tr><td>Preconditions:</td><td colspan="2">User has initiated the program.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">User successfully recorded their video onto the program without errors.</td></tr>
<tr><td rowspan="2">Flow of Activities</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Click on the ‘new recordings’ button.
2. Click ‘detect available cameras’.</td><td>1. Will record the synchronized video if using multiple cameras.
2. Create a new session</td></tr>
</table>

	3. Start recording. 4. Proceed to the capture motion process	containing the video. 3. Prepare the video be processed at the next step.
Exception conditions:	There are problems with the cameras, or the videos are unsynchronized resulting camera error and system failure.	

Table 4; Use Case 'Capture Motion'

Use Case Name	**Capture Motion**	
Scenario:	A new user wants to track their motion using the recordings.	
Triggering Event:	User clicks the 'start motion capture recording' or 'process motion capture videos'.	
Brief Description:	Both buttons will start the motion capturing process using a calibration file if using multiple cameras.	
Actors:	User	
Related Use Cases:	Additional Data Processing.	
Stakeholders:	User	
Preconditions:	User has passed the previous step without any errors and using the correct calibration file.	
Postconditions:	The program successfully analyzed and tracked all the pose & motion within the videos with minimum error margin.	
Flow of Activities	Actor	System
	1. Click on either the 'start motion capture recording' or 'process motion capture videos'. 2. Wait for the process to finish.	1. Apply calibration setting. 2. Track subject's joints and key points. 3. Estimate pose. 4. Process it frame by frame.
Exception conditions:	The program got terminated early and unexpected software/hardware internal failure.	

Table 5; Use Case 'Additional Data Processing'

Use Case Name	**Additional Data Processing**
Scenario:	A new user wants to process and calculate the additional data such as the joint angles & REBA score.
Triggering Event:	Is automatically calculated at the end of the capture motion process.
Brief Description:	There is no button, instead it is a built-in function that is integrated to the main modules that will be executed alongside the main modules of the system.
Actors:	User
Related Use Cases:	Export Data.
Stakeholders:	User

<table>
<tr><td>Preconditions:</td><td colspan="2">User has successfully finished/almost finished the motion capture process.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Will finish the calculation process and produce the additional data.</td></tr>
<tr><td rowspan="2">Flow of Activities</td><td>Actor</td><td>System</td></tr>
<tr><td>-</td><td>1. The additional module will automatically run alongside the main module.
2. Calculate the additional data using the existing data.
3. Produce a new data.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">Can’t read/process the existing data, code failure, or algorithm mismatch.</td></tr>
</table>

Table 6; Use Case 'Export data'

<table>
<tr><th>Use Case Name</th><th colspan="2">Export Data</th></tr>
<tr><td>Scenario:</td><td colspan="2">A new user wants to view the external additional data file in file explorer.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">Is automatically processed at the end of the additional data processing.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">There is no button, instead it is a built-in function that is integrated to the main modules that will be executed alongside the main modules of the system.</td></tr>
<tr><td>Actors:</td><td colspan="2">User</td></tr>
<tr><td>Related Use Cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">User</td></tr>
<tr><td>Preconditions:</td><td colspan="2">User has successfully finished/almost finished the motion capture process.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Will finish the initial process and automatically export the data in the form of a .csv file.</td></tr>
<tr><td rowspan="2">Flow of Activities</td><td>Actor</td><td>System</td></tr>
<tr><td>-</td><td>1. Will automatically export the data processed from the initial step to the current recordings folder.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">Can’t find the specified path, code failure, or algorithm mismatch.</td></tr>
</table>

Table 7; Use Case 'View 3D Visualizations & Additional Data'

Use Case Name	**View 3D Visualizations & Additional Data**	
Scenario:	A new user wants to see the result of the motion capture and its newly processed data.	
Triggering Event:	User clicks on the data viewer after the motion capture process is completed.	
Brief Description:	The button will simply move to another tab and show the results.	
Actors:	User	
Related Use Cases:	-	
Stakeholders:	User	
Preconditions:	User has finished the motion capture process.	
Postconditions:	The program will move to another tab and show the appropriate results based on the results obtained through the motion capture process.	
Flow of Activities	Actor	System
	1. Click on the 'data viewer' tab. 2. User will be provided the option to see the 3d visualizations, recordings, and the newly processed data frame by frame.	1. The program will simply use the data obtained through the motion capture process and display them on this window.
Exception conditions:	Data synchronization failure or unfinished process during initial step.	

Table 8; Use Case 'Choose Additional Data Path'

Use Case Name	**Choose Additional Data Path**	
Scenario:	A new user wants to see the result of the newly processed additional data (Joint angles & REBA posture score) within the data viewer tab.	
Triggering Event:	User is on the data viewer tab and fill the correct path in the given textbox to locate the external file location and click submit.	
Brief Description:	A section that the user can fill in the textbox with the correct path and click submit.	
Actors:	User	
Related Use Cases:	View 3D Visualizations & Additional Data	
Stakeholders:	User	
Preconditions:	User has finished the motion capture process.	
Postconditions:	The program will then display the data for viewers to see.	
Flow of Activities	Actor	System
	1. Click on the 'data viewer' tab. 2. Fill the textbox with	1. The program will simply extract the data and display it frame by frame that matches

	the correct newly additional data file .csv path. 3. Reload GUI	the video frame.
Exception conditions:	Wrong path or internal failure.	

Table 9; Use Case 'Configure Motion Capture Settings'

Use Case Name	**Configure Motion Capture Settings**	
Scenario:	A new user wants to do motion capture but needs to adjust the settings and create a calibration file from active recording.	
Triggering Event:	User will need to adjust the appropriate motion capture setting and generate a calibration (.toml) file.	
Brief Description:	A feature that can output varying results depending on the motion capture settings.	
Actors:	User	
Related Use Cases:	Calibration	
Stakeholders:	User	
Preconditions:	User has already a recording ready to process.	
Postconditions:	The program will process the recording using the personalized settings.	
Flow of Activities	Actor	System
	1. Go to the control panel window and choose the process data tab. 2. Adjust the motion capture setting however you like (advanced)	1. System will use the personalized setting to help process the motion capture process.
Exception conditions:	Internal failure or setting is unsuitable/wrong setting.	

Table 10; Use Case 'Calibration'

Use Case Name	**Calibration**
Scenario:	A new user wants to do motion capture but needs to create a calibration file from active recording.
Triggering Event:	User will need to generate a suitable calibration (.toml) file from the active recording.
Brief Description:	A feature that can generate the calibration file that is essential to start the motion capture process.
Actors:	User
Related Use Cases:	-
Stakeholders:	User

Preconditions:	User has already a recording ready to process.	
Postconditions:	The program will successfully capture the motion of the subject with minimal error margin using the suitable calibration file.	
Flow of Activities	Actor	System
	1. Click either the 'record calibration' or 'calibrate from active recording'. 2. Make sure the subject is holding a charuco board that is visible in at least 2 cameras.	1. System will generate the appropriate calibration file from that recording. 2. Prepare the calibration file for the next motion capture process.
Exception conditions:	Internal failure, setting is unsuitable/wrong setting, or charuco board is not detected.	

Table 11; Use Case 'Browse Session Directory'

Use Case Name	**Browse Session Directory**	
Scenario:	A new user wants to browse the current active session directory.	
Triggering Event:	User will just need to click the directory view tab.	
Brief Description:	After clicking the tab, it will switch view and display the current active directory.	
Actors:	User	
Related Use Cases:	-	
Stakeholders:	User	
Preconditions:	User has started the program.	
Postconditions:	The program will successfully show the current active recording directory similar to that of a file explorer.	
Flow of Activities	Actor	System
	1. Click the Directory view tab. 2. User can view the directory that has folders and files.	1. System will generate the path of the current directory. 2. Will show the tree of the directory in the tab.
Exception conditions:	Internal failure.	

Table 12; Use Case 'Configure Motion Capture Settings'

Use Case Name	**Configure Motion Capture Settings**	
Scenario:	A new user wants to do motion capture but needs to adjust the camera settings.	
Triggering Event:	User will need to click ‘detect available cameras’ in the cameras tab.	
Brief Description:	A feature that can show the available cameras and adjust its settings such as the resolution and framerate.	
Actors:	User	
Related Use Cases:	-	
Stakeholders:	User	
Preconditions:	User has started the program.	
Postconditions:	The program will show the current available cameras and the settings applied in real time.	
Flow of Activities	Actor	System
	1. Go to the cameras tab. 2. Click the ‘detect available cameras’ button. 3. See the current active cameras. 4. Adjust the camera settings however the user liked.	1. System will connect to each of the cameras. 2. Display the real time footage. 3. Synchronized the settings applied to the cameras in real time.
Exception conditions:	Internal failure or setting is unsuitable/wrong setting.	

d) System/Process Flowchart

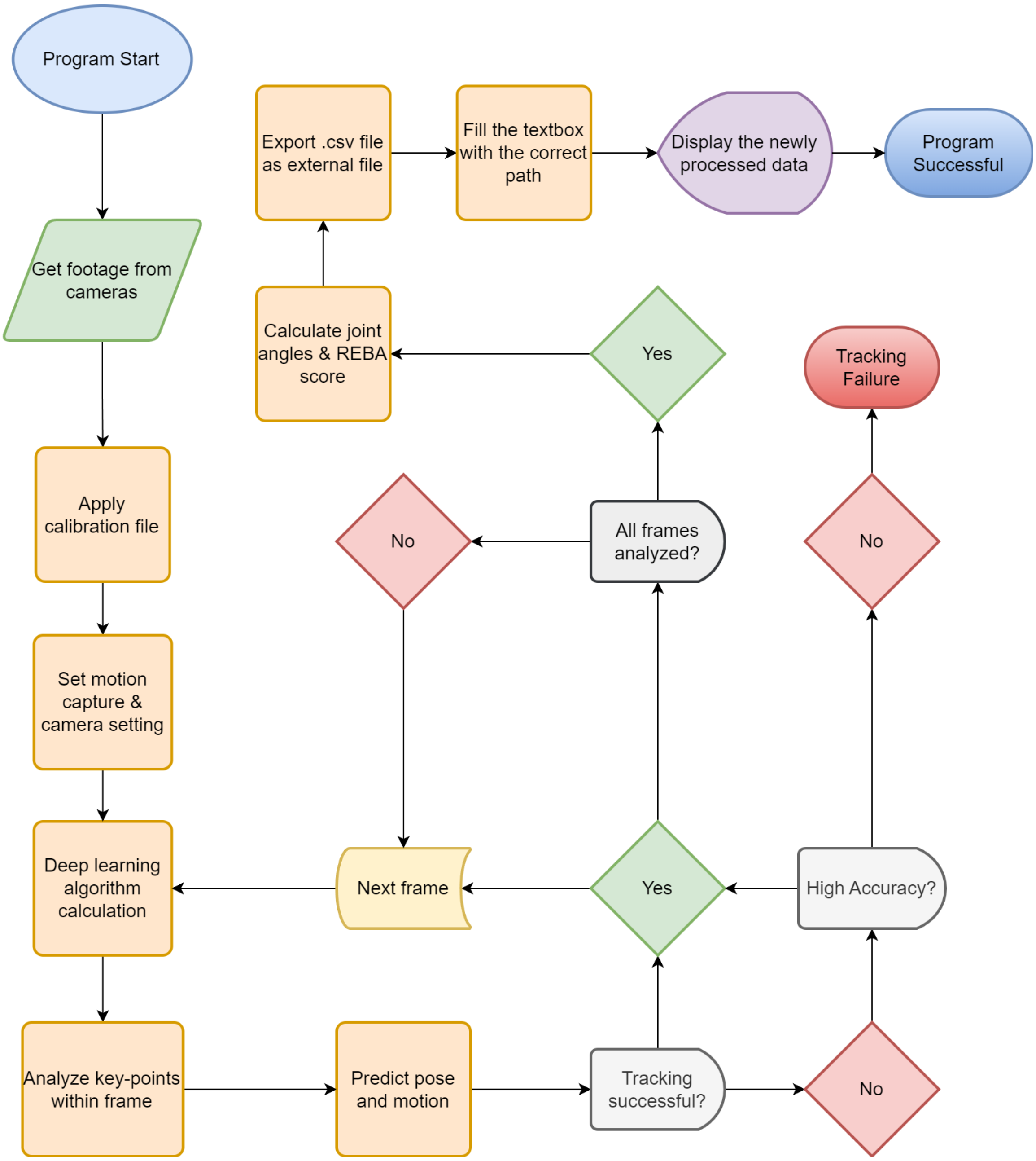


Figure 12; Updated System Flowchart

e) Overall Class Diagram

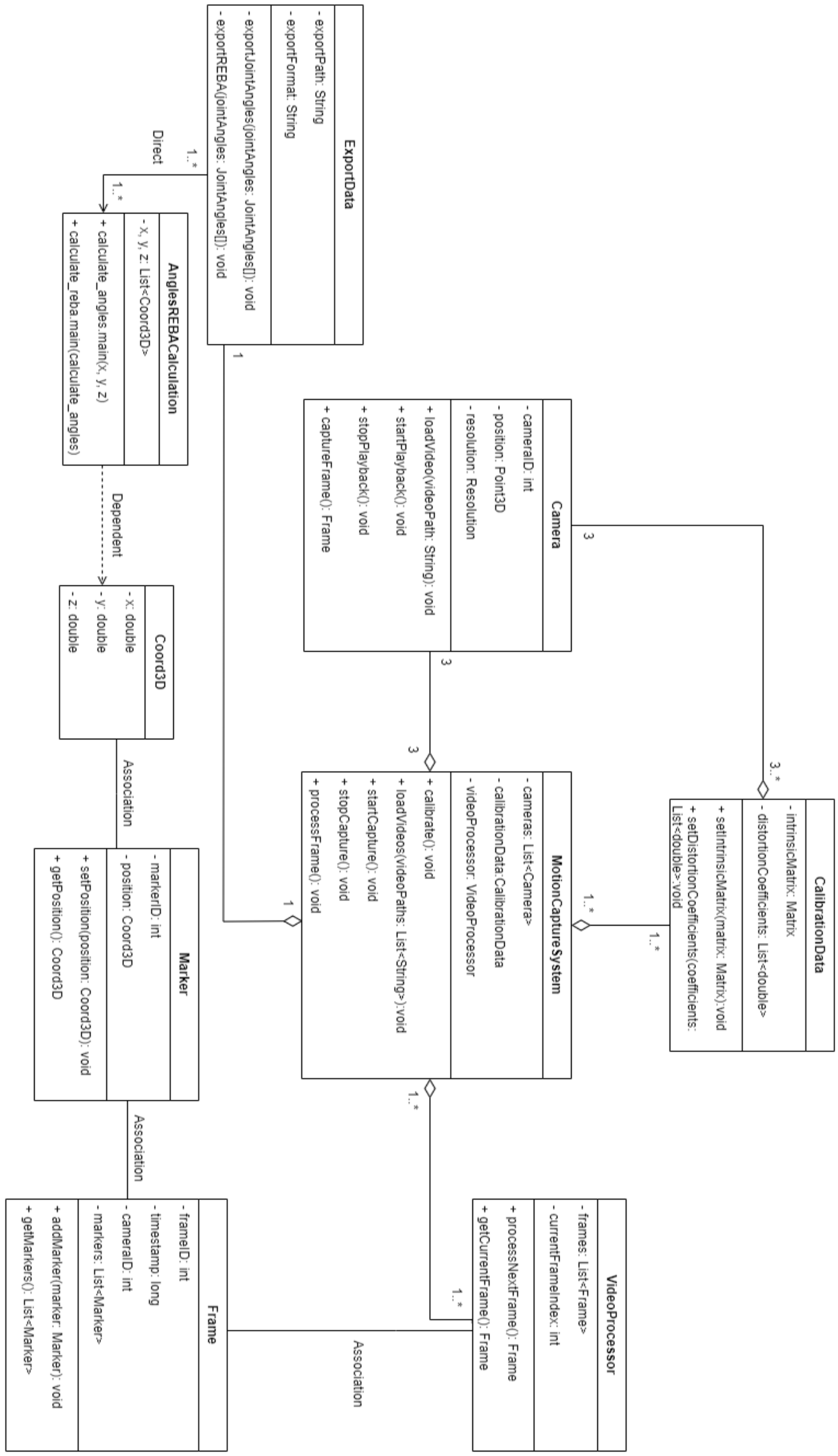


Figure 13; Updated Overall Class Diagram

f) Full System Architecture Diagram

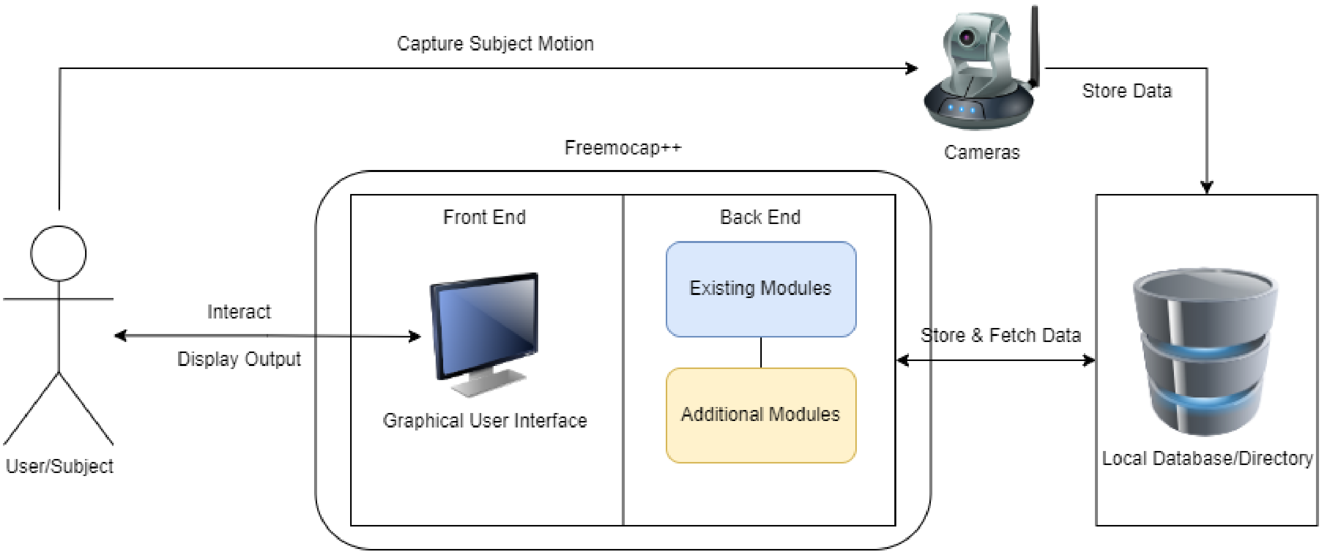


Figure 14; Updated System Architecture Diagram

g) Intelligent Method/Algorithm Used

A. Existing System:

- Mediapipe Blazepose CNNs (Convolutional Neural Network)
- Anipose Camera Calibration & 3D Reconstruction

B. Extended Functionalities / Added Modules:

- Dynamic Programming Algorithm
- Object Oriented Programming (OOP)

h) Technology Deployed

The software and hardware that was planned during the future development process of this program are as follows.

Table 13; Software & Hardware Used

<table>
<tr>
<td>

<u>Software</u>

Visual Studio Code (as the IDE)
Python 3.11 (as the main programming language)
Anaconda (as the language platform interpreter)
Relevant Libraries (OpenCV, Numpy, Mediapipe, Pandas, Matplotlib, etc)
*Other libraries may be added during development
Google Chrome (as web browser and source of information/data)
Windows 10 (as the Operating System (OS))
Draw.io web application (as the main designing tool)
Freemocap (as the base software that I aim to extend its functionalities)

</td>
<td>

<u>Hardware</u>

3pcs Eken 4K H9R Action Camera
3pcs 3m Micro-USB Cable
1pcs USB Port Hub
1pcs Custom Made Personal Computer

CPU: Intel Core i7-10700K @ 3.80GHz
RAM: 32GB DDR4
GPU: Nvidia Geforce RTX 3060Ti

1pcs HP Spectre x360 Convertible-Laptop

CPU: Intel i7-8565U @ 1.8GHz
RAM: 16GB
GPU: Intel UHD Graphics

</td>
</tr>
</table>

4 SYSTEM DESIGN & IMPLEMENTATION

4.1. Comprehensive Elucidation of System Architecture

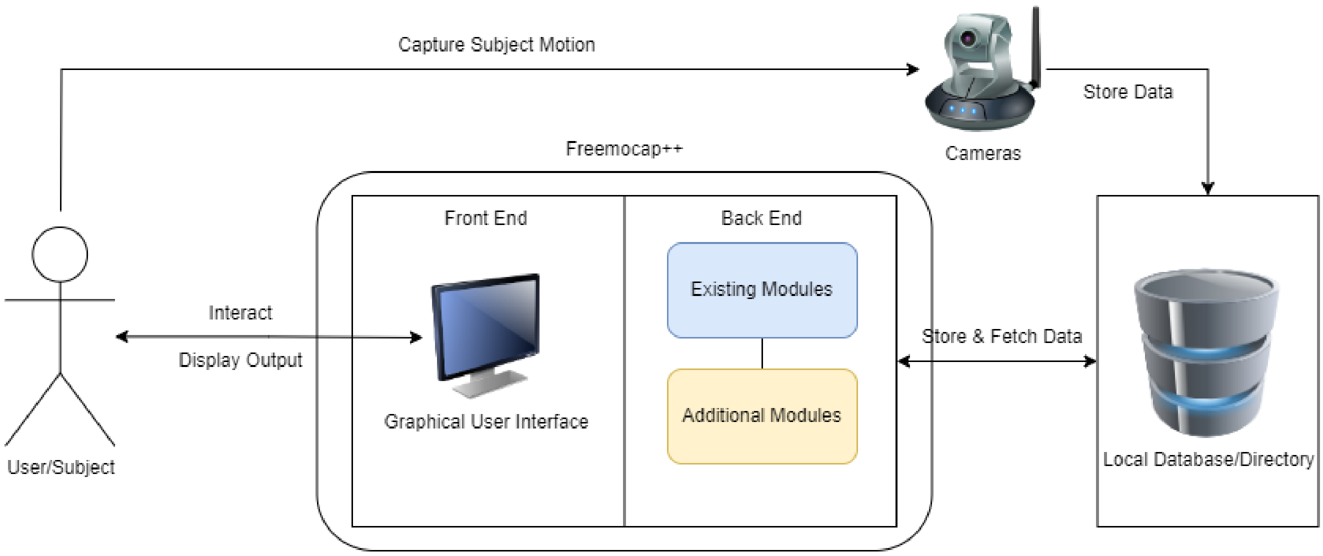


Figure 15; Updated System Architecture Diagram

a) **User/Subject**

The user/subject will act as the main component of the whole architecture as the main source of data for capturing its motion & tracking its pose in the form of video recordings via the use of cameras.

b) **Cameras**

This component functions as the tool to capture the raw data of the subject movement in the form of video recordings from multiple angles using multiple cameras.

c) ***Freemocap++***

This component acts as the main software that consists of the front end/GUI and the back end/System. Additionally, it is also a hub that's connected to all the components.

User/subject can interact with the software through the GUI and make use of all its functionality through it, including the cameras preview and local database/directory.

d) Local Database/Directory

Acts as the database component of the architecture to store and fetch data from the local directory of each of the sessions. The software can read and access the necessary data to do motion capture from this directory and store the process data in the same directory.

4.2. Detailed Elucidation of Intelligent Method/Algorithm Implementation Used in The System

4.2.1. Implementation of Intelligent Computing Methods in the System

A. Existing System (Freemocap)

- **Mediapipe Blazepose CNNs**

  MediaPipe BlazePose is an advanced open-source pose estimation model developed by Google, designed to track human body landmarks in real-time. The model uses *Convolutional Neural Networks* (CNNs) to detect and track human poses accurately and efficiently.

- BlazePose is divided into two main components:
  - **BlazePose Detector**: Detects the presence and rough location of a person in the image or ROI (Region of Interest) that contains the person using lightweight CNN, such as MobileNetV2 to extract features from input images.
  - **BlazePose Landmarks**: Refines the detected region and accurately predicts the positions of specific body landmarks as 33 key body landmarks using the same feature extraction method as the Blazepose Detector.

- Pipeline workflow of Blazepose:
  1. **Frame Acquisition**: Capture an image or video frame from the camera.
  2. **Person Detection**: The BlazePose detector identifies the presence and location of a person.
  3. **ROI Extraction**: Crop the image to the region around the detected person.

4. **Landmark Prediction**: The BlazePose landmarks model predicts the precise coordinates of the body landmarks within the ROI.
5. **Post-Processing**: Refine the landmark positions and possibly apply smoothing to reduce jitter in real-time applications.

- Training process of Blazepose:
  - **Data**: Model is trained on a large dataset of annotated human poses. The dataset includes diverse poses, body types, and occlusions to ensure robustness by Google.
  - **Loss Function**: The training process involves a multi-task loss function such as ***Bounding Box Loss, Heatmap Loss, Offset/Regression Loss***.
  - **Optimization**: Standard optimization techniques like stochastic gradient descent (SGD) are used to minimize the loss function during training.

Although the model is optimized for real-time applications to provide instantaneous feedback, Freemocap employs a post-processing method for multi-camera setups to enhance the overall accuracy and precision of the landmarks detection.

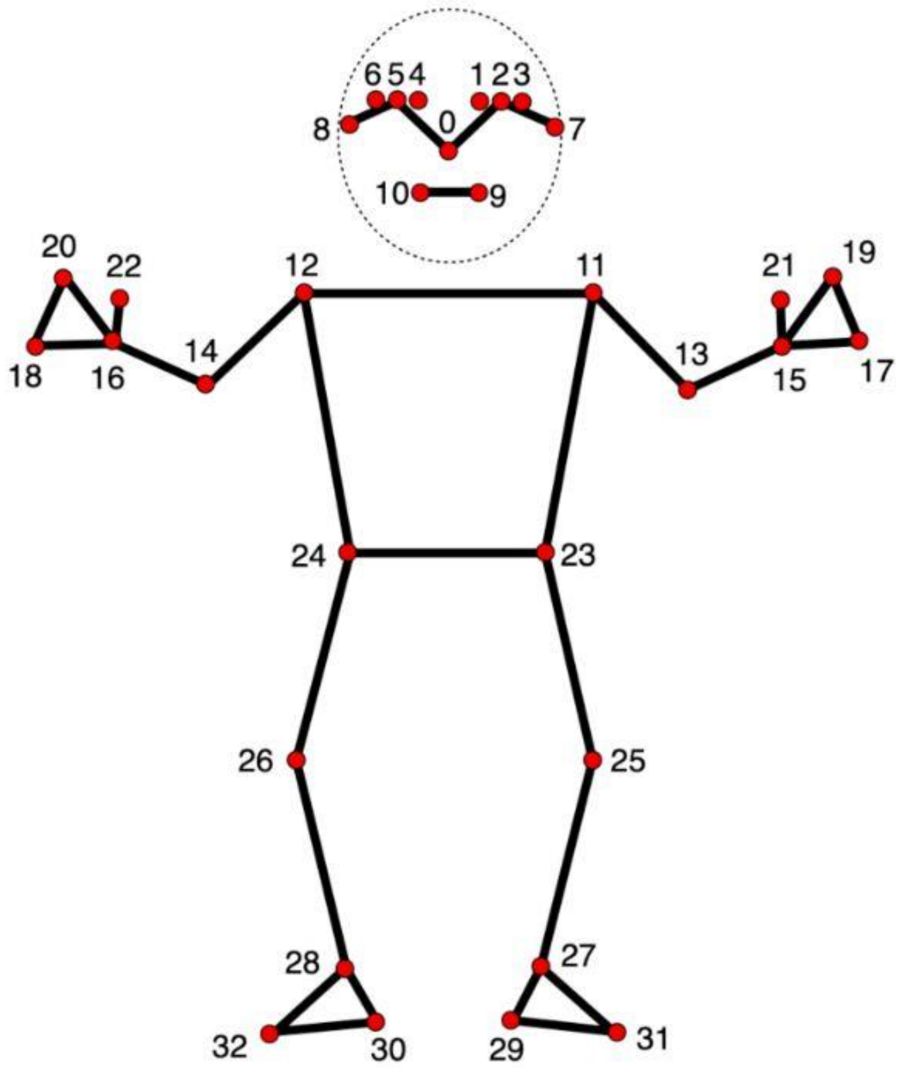


Figure 16; Mediapipe Pose Landmarker Model (33 Landmark Locations)

Src: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

- **Anipose Camera Calibration & 3D Reconstruction**

  Anipose is an open-source tool designed for markerless 3D pose estimation, particularly used in research settings for animal behavior analysis and biomechanics. It uses deep learning techniques to track the 3D positions of key points on animals from multiple camera views without the need for physical markers.

  Anipose key features are as follows:

  - **Markerless Pose Estimation**: Tracks 3D positions of key points on animals without using physical markers.

- **Multi-Camera Setup**: Utilizes synchronized video footage from multiple angles for accurate 3D reconstruction.
- **Deep Learning Models**: Employs models like DeepLabCut for 2D keypoint detection from each camera view.
- **3D Reconstruction**: Uses triangulation to compute 3D coordinates from 2D detections.
- **Camera Calibration**: Includes tools for calibrating camera setups to ensure accurate spatial measurements.

▪ Pipeline workflow of Anipose:

- **Setup & Calibration**: Camera setup should capture the subject from various angles using multiple cameras. Calibration is needed to determine the intrinsic parameters and extrinsic parameters (like position and orientation relative to each other) to ensure an accurate 3D reconstruction from 2D views.
- **Synchronized Recording**: Video recording must be synchronized from all cameras simultaneously to ensure frames correspond to the same point.
- **2D Keypoint Detection**: (Using mediapipe Blazepose CNN).
- **3D Reconstruction**: Use the 2D keypoints and calibration data to perform **triangulation** that generates the 3D keypoints and reconstruct it in a 3D space.

▪ Training process of Anipose:

- **Data Annotation**: Collect and annotate images with keypoints of interest.
- **Model Training**: Train a deep learning model (e.g., DeepLabCut) on the annotated images to recognize keypoints or use a pre-trained model such as mediapipe blazepose.
- **Model Evaluation**: Assess model performance using metrics like *accuracy* and *mean squared error* (MSE).
- **Fine-Tuning**: Adjust hyperparameters or refine annotations if necessary to improve model performance.

- **Inference**: Apply the trained model to new video frames to detect 2D keypoints.

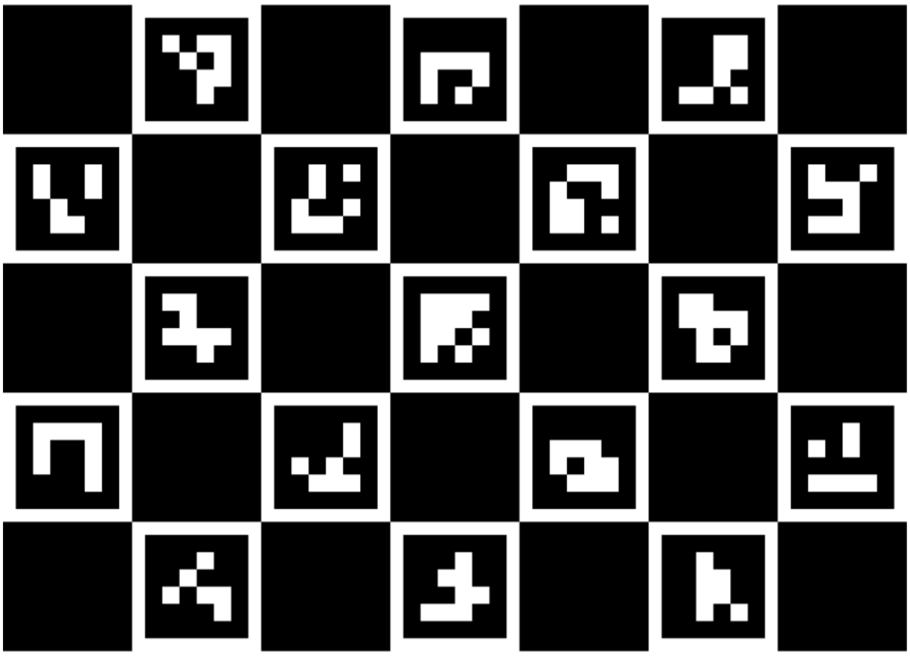


Figure 17; Image of the Charuco Board used for the Calibration Process

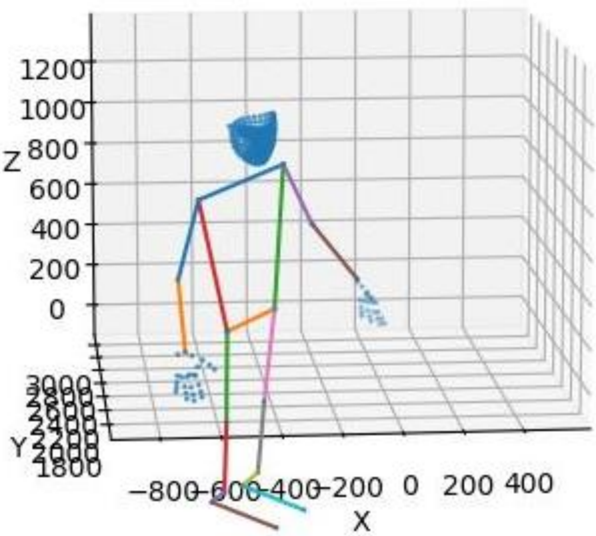


Figure 18; Anipose 3D Reconstruction

B. Extended Functionality / Added Module

- **Dynamic Programming Algorithm & OOP**

  Dynamic programming is a powerful algorithmic technique used to solve problems by breaking them down into simpler subproblems and storing their solutions to avoid redundant computations. I plan to implement DP algorithms for data processing tasks, especially when dealing with classes and functions distributed across multiple file sources, requires careful design, understanding of file relationship & dependency, and the use of OOP and their concepts.

  The enhanced functionality/module I've developed displays precise joint angles for specific limbs that are required for another use, computed through a defined set of classes and methods. This involved extracting 3D coordinates from each frame of the motion capture process, applying a mathematical formula to calculate the angles, and exporting the resulting data as a separate .csv file.

  After obtaining the joint angle set, the data is leveraged for frame-by-frame computation of the REBA posture score method, minimizing user interaction. While the angles are automatically processed in the REBA calculation, certain inputs such as the neck or trunk condition (twisted or side bending) require manual user input.

  (*as of this point the REBA posture score calculation module isn’t yet completed but will be finished before the final presentation)

  To ensure the algorithm functions accurately, the user must also provide another crucial input: the file location of the separate .csv file. However, due to time constraints and my limited understanding of the main software *‘Freemocap’*, automatic retrieval of the file path is not feasible yet.

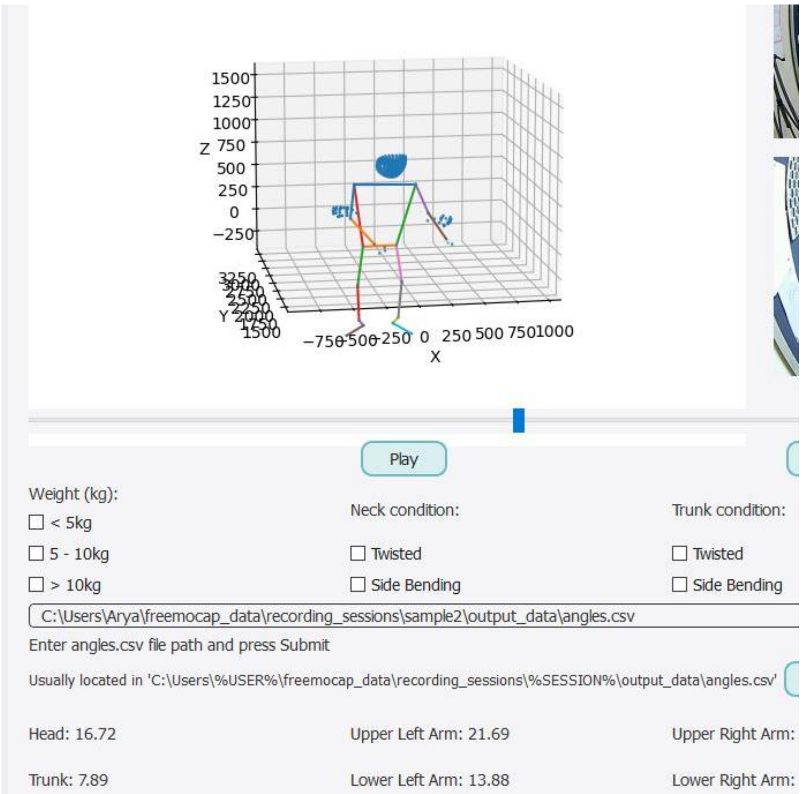


Figure 19; Detailed Data Viewer Tab

Once all calculations are completed successfully, the results are presented in a designated tab within the main software interface. Here, users can examine the joint angles and REBA posture score frame-by-frame and adjust the condition and file path as needed.

4.2.2. Updated System Module Diagram Elucidation

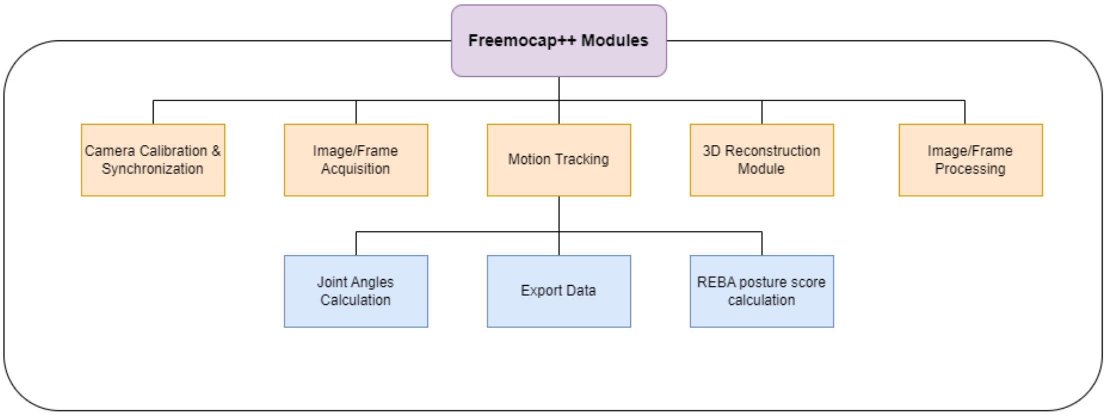


Figure 20; Updated System Module Diagram

**although the module diagram that's presented doesn't fully represent all of the existing system modules, due to lack of information & documentation, it would still provide as a rough outline of the overall modules involved in the existing motion tracking system (Freemocap).*

Existing Modules Elaboration:

- **Camera Calibration & Synchronization**
  This module calibrates the cameras to accurately determine the floor's positions and orientations and the subject's position relative to these points, ensuring precise 3D reconstructions from 2D views. It uses sophisticated algorithms and a Charuco Board to help align the camera perspectives into a cohesive 3D framework. Additionally, the module verifies the synchronization of recordings from multiple cameras, checking for temporal alignment to ensure all camera feeds capture frames simultaneously.
- **Image/Frame Acquisition**
  This module is for acquiring each frame of the video as an image that's ready to be processed by another module, specifically the motion tracking module.
- **Motion Tracking Module**
  This module's main function is to track the movement of the subject and to determine the subject's important key points, landmarks, and features to be ready to get processed by another module, specifically the image/frame processing module.
- **Image/Frame Processing Module**

This is the most important module since this module will process all the data that is sent from multiple modules to generate an accurate estimation of the subject's pose and its 2D coordinates for all the frames that are in the video recordings.

- **3D Reconstruction Module**
  This module is designed to generate an accurate 3D representation from the 2D coordinates obtained by the initial module. It achieves this through the use of triangulation and other advanced algorithms that process and integrate multiple 2D perspectives.

Extended Modules Elaboration:

- **Joint Angles Calculation Module**
  This module serves as the main module that calculates the set of various 3D coordinates obtained through the full motion capture process to a more refined data, which is the set of joint angles for the subject's neck, trunk, upper & lower arms, upper & lower legs for all the frames within the video recordings.
- **REBA Posture Score Calculation Module**
  This module is another module that needs the joint angles data and the condition of specific body parts to be able to calculate its posture score through the REBA method. Once all conditions are fulfilled, the module will be able to produce the subject REBA posture score frame by frame.
- **Export Data**
  Once all the calculations are finished, this module will then export all the processed data (joint angles & REBA posture score) to an external .csv file. This is to provide the user access to the raw data of the joint angles and REBA posture score frame by frame.

4.2.3. System/Process Flowchart Elucidation

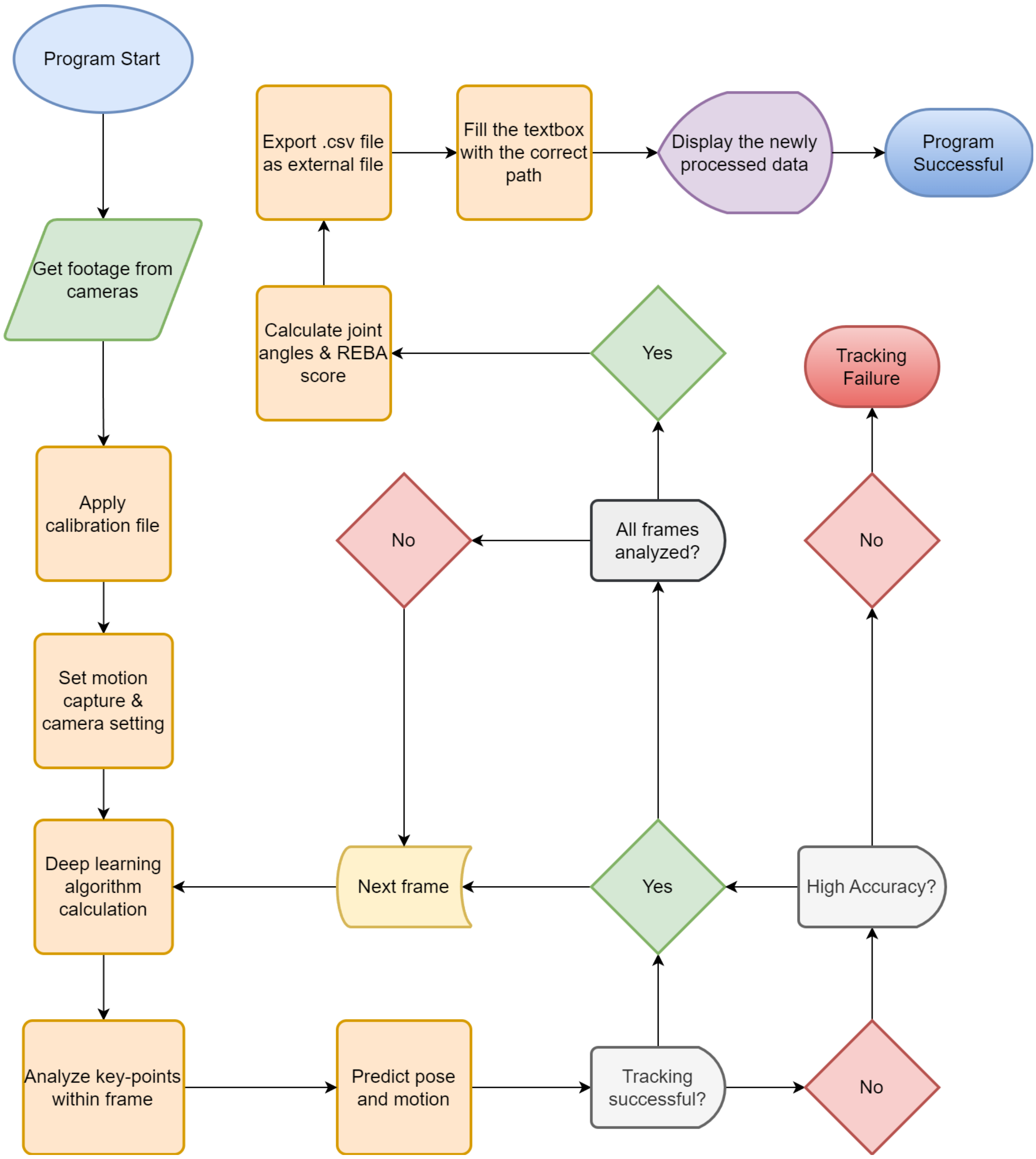


Figure 21; Updated System Flowchart

Based on the flowchart diagram, the overall system flow should be as follows.

1. Start the program.
2. Get synchronized video recording (Manual or record using the system).
3. Get calibration file from video recording using a Charuco board.
4. Set suitable motion capture settings.
5. Start motion capture process.

6. Analyze key points & landmarks within the frame.
7. Predict probable poses using the key points & landmarks.
8. Proceed to the next frame if failed and still proceed if it’s not accurate.
9. Repeat step number 5 if tracking successful or has high accuracy.
10. Check if all frames have been processed, if not continue analyzing next frame.
11. If it has too low accuracy, the tracking will continue but should be considered a failure.
12. After checking if all frames have been processed and analyzed, proceed to calculate the set of joint angles & REBA posture score using a custom mathematical formula/algorithm.
13. Export the newly processed data as a .csv file in the current active directory.
14. Fill the textbox with the correct path to the mentioned .csv file.
15. Click submit, reload GUI/restart system.
16. It should display the new data and can be manipulated by the user via checkboxes & slider to see the video frame-by-frame.
17. Program successful.

4.3. User Interface Design/System Screenshot

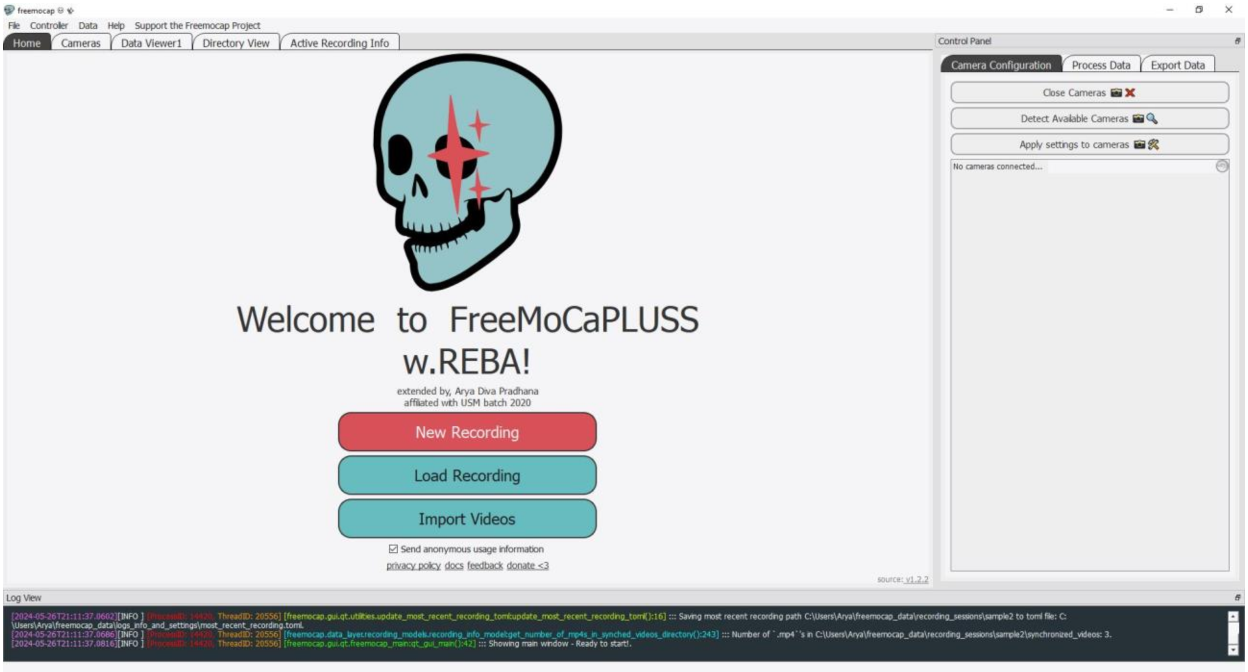


Figure 22; Fremocap++ Home Tab

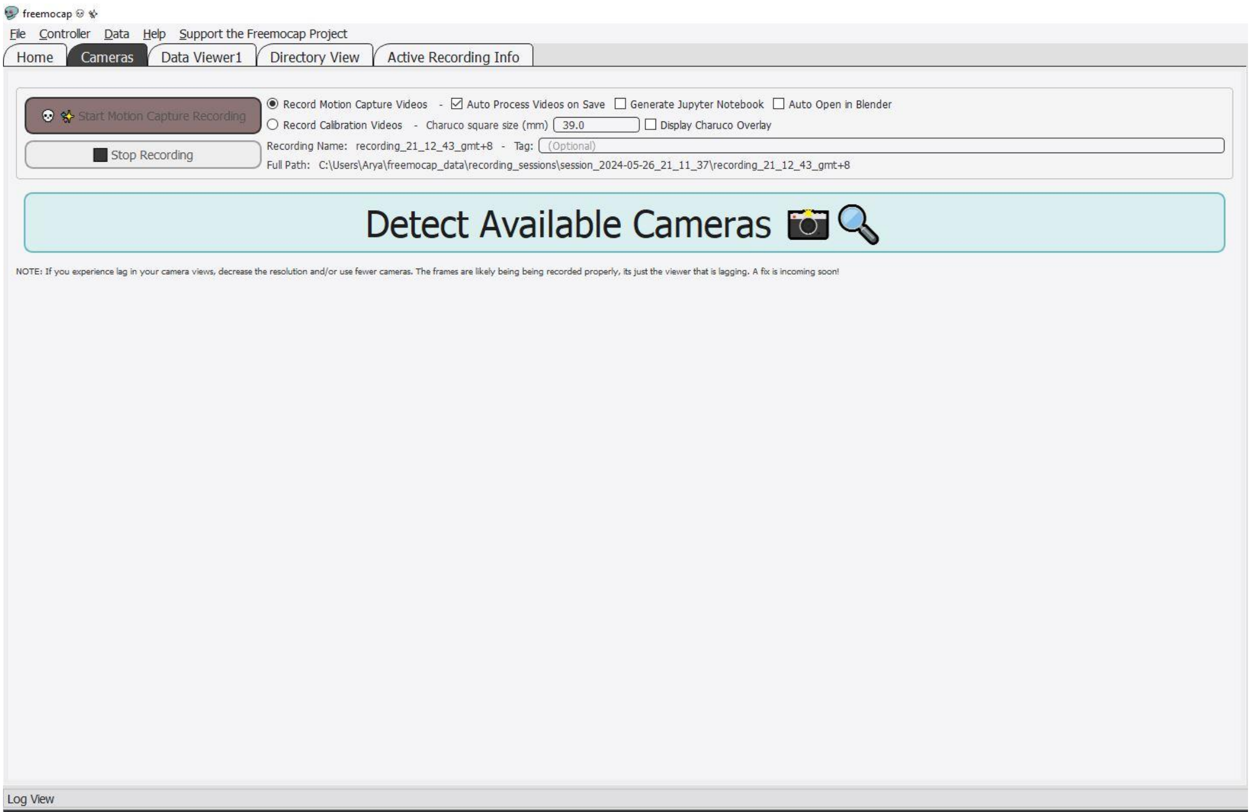


Figure 23; *Freemocap*++ Cameras Tab

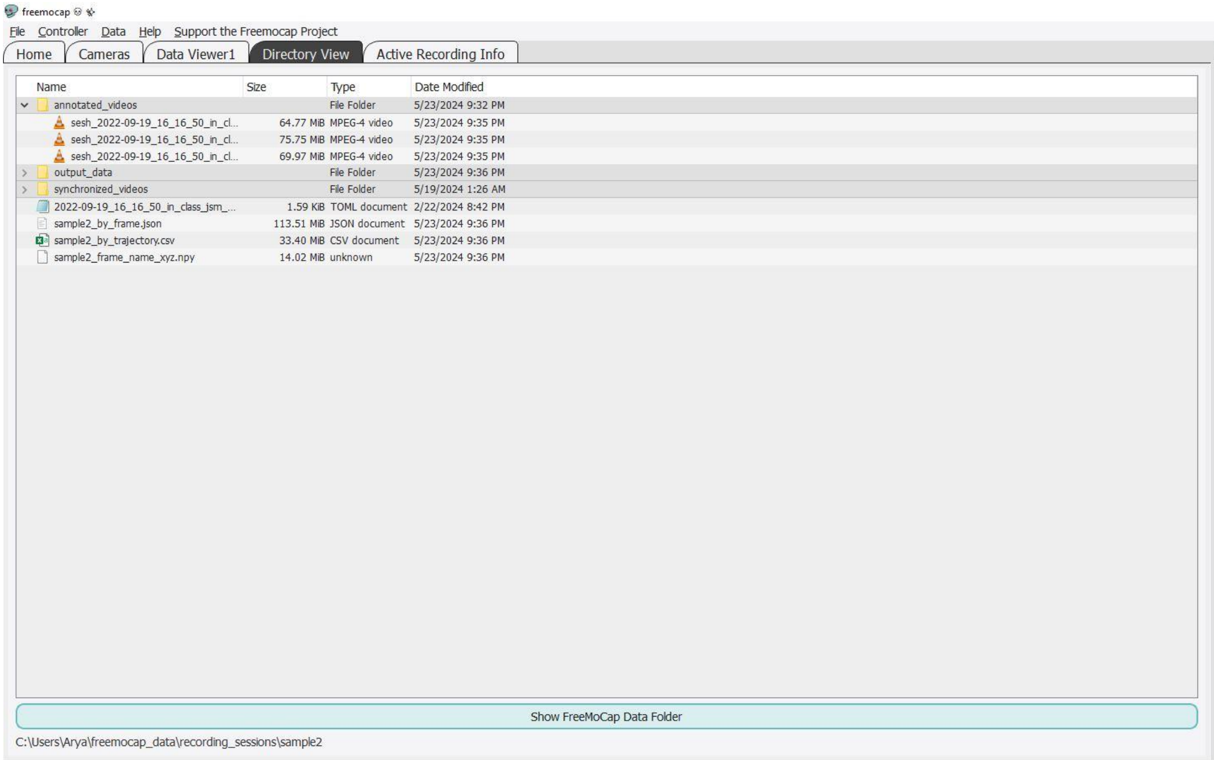


Figure 24; *Freemocap*++ Directory View Tab

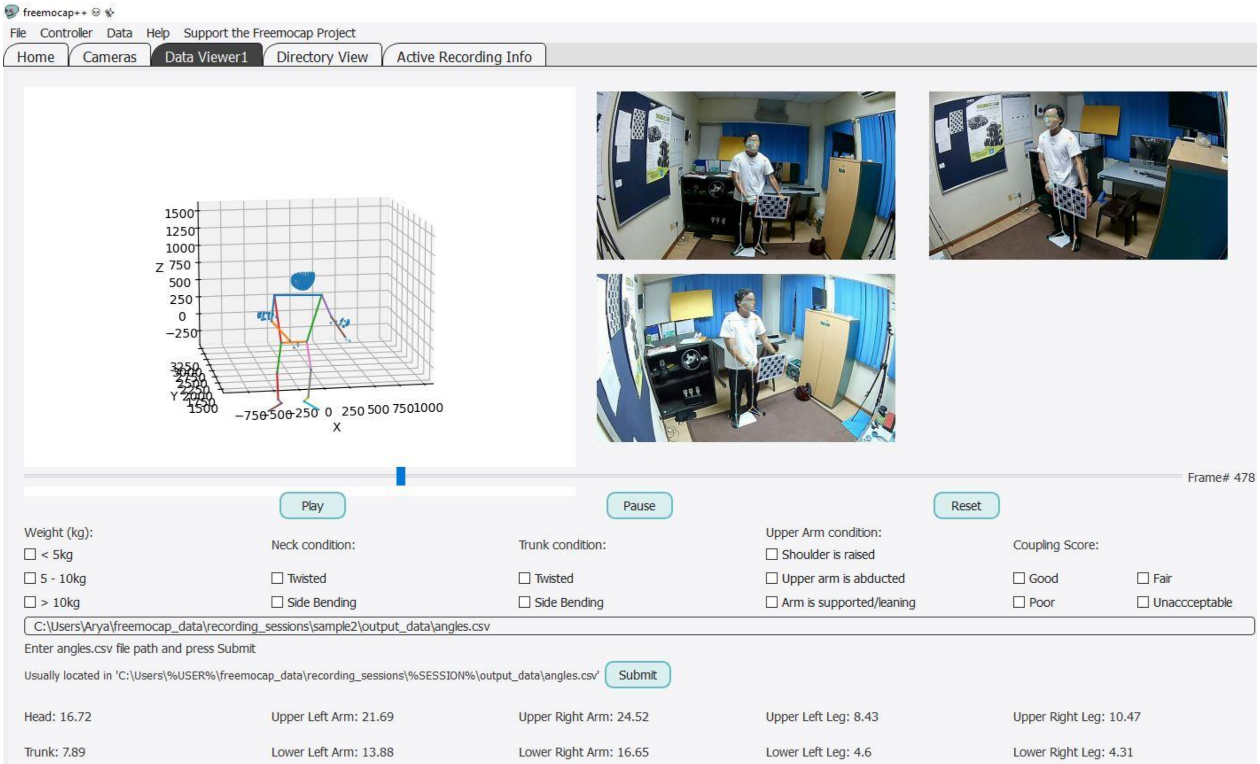


Figure 25; *Freemocap*++ Data Viewer Tab

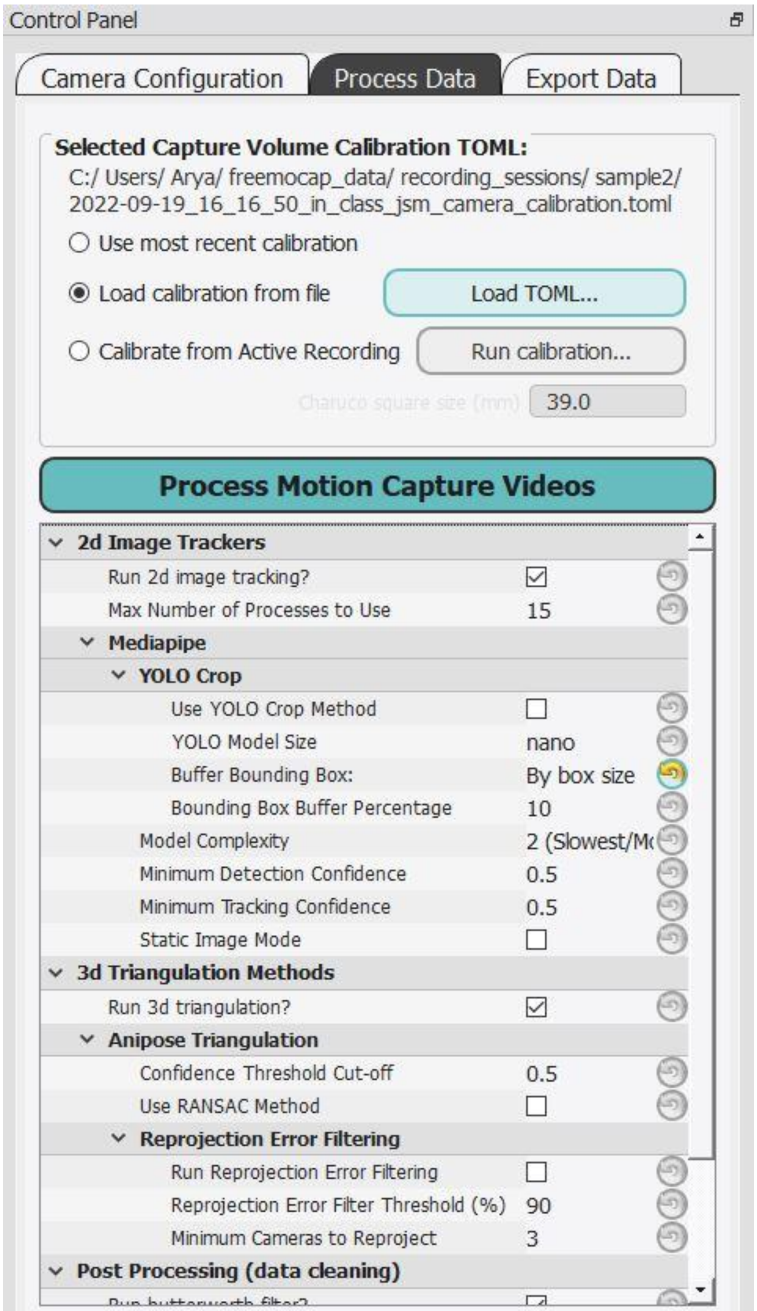


Figure 26; Motion Capture Settings

4.4. Implementation Strategy

Due to the nature of my project, it is essential to initially develop my modules from scratch using a dummy dataset to verify the functionality of my modules. The process begins at the most fundamental level, gradually building upwards to create a cohesive and cooperative module that seamlessly integrates with the main modules of the existing system modules. Therefore, I have chosen the **Bottom-Up approach** for my project. To successfully integrate my modules with the existing ones, I need to thoroughly understand the functions, classes, files, and variable relationships. This ensures that my modules can be integrated into the main system without causing errors or internal failures that might lead to unexpected system crashes.

Additionally, this approach is ideal because the modules I plan to add will extend the overall system's functionality by processing data, exporting the processed data, and displaying the results through the GUI of the existing system. These operations require input from other modules to function correctly. Thus, the best way to implement my modules without risking system failures is to initially use temporary dummy data. This allows me to develop and test the new modules in isolation, ensuring their correctness and stability. Once the modules are functioning as expected, I can adjust them to align with the formats and protocols used by the other existing modules, facilitating smooth integration and interoperability within the overall system.

4.5. System Design & Implementation Overview

In general, my project mainly focuses on extending the functionalities of an existing open source multi camera markerless motion capture system ***'**Freemocap'*, as well as studying and analyzing it by understanding the mechanisms & process behind it.

4.5.1. Overall System Structure

The system architecture comprises of 4 main components:

- **User/Subject:** as the main subject for capturing the subject's movement/pose data.
- **Cameras:** as the main tool to help capture the video recordings of the subject's movement.
- **Local database/local directory:** act as the local database to store and fetch the necessary data/recordings in the user local directory.
- ***Freemocap++*:** as the main software to process the overall motion capture process and act as the main system/platform in which I aim to extend its functionality by adding several additional modules.

4.5.2. Overall Intelligent Method/Algorithm Used

The intelligent method/algorithm used in development for the system are divided into 2 categories: Methods used in the **existing system** and methods used in the **extended functionality**.

i. Existing system intelligent methods/algorithms: Mediapipe Blazepose CNNs and Anipose Camera Calibration & 3D Reconstruction.
ii. Extended functionality methods/algorithms: Dynamic Programming Algorithm & OOP.

4.5.3. Overall System Modules Structure

The system modules are divided into 2 categories: **Existing modules & Additional modules**.

i. Existing modules: Camera Calibration & Synchronization, Image/Frame Acquisition, Motion Tracking Module, Image/Frame Processing, and 3D Reconstruction.
ii. Additional modules: Joint Angles Calculation, REBA Posture Score Calculation, and File Export.

4.5.4. Design Overview

Although my initial plan was to develop the system from scratch using *'Freemocap'* as a reference, after a thorough discussion with my supervisor and considering suggestions from my examiner, I decided to extend the existing system's functionalities instead. They recommended adding a few new modules to enhance the system by displaying a new type of information. This approach not only leverages the robust foundation of '*Freemocap'* but also allows for a more efficient development process, enabling us to introduce advanced features and improve the system's overall capabilities without reinventing the wheel.

5 SYSTEM TESTING & EVALUATION

5.1. Testing Strategy

5.1.1. Unit Testing

"Unit testing is meant to test individual units, components, or individual modules of the software to ensure they are functioning as intended."

src: https://www.geeksforgeeks.org/software-testing-strategies/

Given the nature of my project, it is crucial to first test my additional modules individually before integrating them into the main system to ensure overall system stability. To achieve this, I require a dummy input that matches the type and parameters of the main system modules. Implementing my modules using Object-Oriented Programming (OOP) has streamlined the testing process, as I can easily call the relevant classes and functions, facilitating efficient and accurate module verification.

Objective: To verify each additional modules correctness in isolation

Approach:

- Modules Tested: Joint Angles Calculation, REBA Posture Score Calculation, and Export Data.
- Process:
    - Get raw data (*mediapipe_body_3d_xyz.csv*) path directly from the local directory of an active session.
    - Set direct path to read the .csv file and proceed to do joint angles calculation modules.
    - Process joint angles calculation for each frame.
    - Process REBA posture score for each frame.
    - Export data to local folder.
- Execution: Isolated environment/directory from the active recording system.

5.1.2. Integration Testing

"Integration testing is meant to test the integration of different components of the software to ensure they work together as a system."

src: https://www.geeksforgeeks.org/software-testing-strategies/

After verifying that each of my individual modules operates correctly with various dummy inputs and developing responses to handle unexpected errors, my next step is to test the interaction between my modules using actual data. This involves sending data to one module to simulate how the main system transmits data to and from my modules. This integrated testing is crucial to ensure that the relationships between my modules and the main system modules are correctly established. It allows me to identify any relational errors and confirm that each module functions properly in different scenarios, thereby ensuring seamless integration and system reliability.

Objective: Ensure that different modules work together as intended when integrated.

Approach:

- Modules Tested: Joint Angles Calculation, REBA Posture Score Calculation, and Export Data.
- Process:
    - Get raw data (*mediapipe_body_3d_xyz.csv*) path directly from the local directory of an active session.
    - Set direct path to read the .csv file and proceed to do joint angles calculation modules.
    - Process joint angles calculation for each frame.
    - Process REBA posture score for each frame from joint angles data.
    - Export data to local folder.

Execution: Isolated environment/directory from the active recording system.

5.1.3. Functional Testing

"Functional testing is meant to test the functional requirements of the software to ensure they are met."

src: https://www.geeksforgeeks.org/software-testing-strategies/

After completing the initial tests, I then proceed to verify that the modules are working as intended and fulfilling the planned functions to meet the requirements of the extended software. This involves tracking the process, inputs, and outputs of each module and their interactions. I will systematically check each module's values and outputs to ensure they align with expectations. If any module's output deviates from the expected results, the functional testing will be considered unsuccessful, as the module does not function as intended.

Objective: Ensure that each modules functions as intended and generates the expected output and response.

Approach:

- Modules Tested: Joint Angles Calculation, REBA Posture Score Calculation, and Export Data.
- Process:
  - Get raw data (*mediapipe_body_3d_xyz.csv*) path directly from the local directory of an active session.
  - Set direct path to read the .csv file and proceed to do joint angles calculation modules.
  - Process joint angles calculation for each frame.
  - Process REBA posture score for each frame from joint angles data.
  - Export data to local folder.
  - Check whether each module functions as intended.

Execution: Isolated environment/directory from the active recording system.

5.1.4. System Testing

"System testing is meant to test the complete extended software system to ensure it meets the specified requirements of the extended system."

src: https://www.geeksforgeeks.org/software-testing-strategies/

At this stage, my modules should have passed all the initial tests and are ready for integration with the main system. All the modules function smoothly and meet the project's requirements without any errors. The integration process involves thorough analysis and verification to identify which parts of the main system produce the necessary inputs for my modules to function as intended. This careful approach ensures seamless integration and optimal performance.

Objective: To validate the overall behavior and functionality of the integration of the main system modules and the additional modules as a whole, ensuring that it meets the specified requirements and functions correctly in different scenarios.

Approach:

- Modules Tested: Joint Angles Calculation, REBA Posture Score Calculation, and Export Data.
- Process:
    - Get the latest path data if it exists in memory.
    - Get raw data (*mediapipe_body_3d_xyz.csv*) path from textbox (user input)
    - Connect the modules to the path to read the .csv file and proceed to do joint angles calculation modules.
    - Process joint angles calculation for each frame.
    - Process REBA posture score for each frame from joint angles data.
    - Export data to local folder.

Execution: Integrated with the main system.

5.1.5. Regression Testing

"Regression testing is meant to test the software after changes or modifications have been made to ensure the changes have not introduced new defects."

src: https://www.geeksforgeeks.org/software-testing-strategies/

Finally, after successfully integrating my modules with the main system and connecting the necessary inputs to produce the expected outputs, the final test involves a comprehensive check for potential errors or defects introduced by the modifications. This thorough testing ensures that the addition of new modules does not negatively impact the overall functionality and stability of the main system.

Objective: To verify whether there are any unexpected error or defects when applying modifications to the existing system

Approach:

- Modules Tested: Joint Angles Calculation, REBA Posture Score Calculation, and Export Data.
- Process:
  - Get the latest path data if it exists in memory.
  - Get raw data (*mediapipe_body_3d_xyz.csv*) path from textbox (user input)
  - Connect the modules to the path to read the .csv file and proceed to do joint angles calculation modules.
  - Process joint angles calculation for each frame.
  - Process REBA posture score for each frame from joint angles data.
  - Export data to local folder.
  - Check for any error or defects through the terminal or error log.

Execution: Integrated with the main system.

5.2. Test Cases

Table 14; Test Case Table 1

Test Case ID	TC_001
Test Case Name	CSV Read Test
Test Scenario	To test whether the csv reader method can read the content of the .csv file and output a value in the console.
Steps	- Set absolute path to a .csv file. - Create a csv reader method. - Read a specific row and column of the .csv file. - Output the value on the terminal.
Expected Result	The referred value will be displayed on the terminal/GUI.
Actual Result	Same as expected result.
Test Status	Pass

Table 15; Test Case Table 2

Test Case ID	TC_002
Test Case Name	Angle Calculation Method
Test Scenario	To test whether the method can generate a precise angle with the given x, y, z coordinates.
Steps	- Get a specific coordinate data from .csv file. - Get the data. - Calculate using the specified method. - Print a result on the terminal.
Expected Result	The referred angle value will be displayed on the terminal/GUI.

Actual Result	Same as expected result.
Test Status	Pass

Table 16; Test Case Table 3

Test Case ID	TC_003
Test Case Name	Verify File Path in Textbox
Test Scenario	To test whether it can verify the given path exist or not
Steps	- User input a path in the text box. - A method will verify if the file location referred by the path exist or not
Expected Result	If the file wither exists or not, print the status in terminal/GUI.
Actual Result	Same as expected result.
Test Status	Pass

Table 17; Test Case Table 4

Test Case ID	TC_004
Test Case Name	File Export
Test Scenario	To test whether it will export the calculated joint angles data as an external .csv file in the current directory.
Steps	- Call the function to export the joint angles to an .csv file in the calculate joint angles class. - Call and run the method.
Expected Result	Create a .csv file containing the value of the calculated joint

	angles.
Actual Result	Same as expected result.
Test Status	Pass

Table 18; Test Case Table 5

Test Case ID	TC_005
Test Case Name	Calculate REBA posture score
Test Scenario	To test whether it will generate an accurate score based on the joint angles data and its built-in functions.
Steps	- Link the class and method to be able to run in calculate joint angles class. - Check the boxes for the conditions. - Call and run the method.
Expected Result	Generate an accurate REBA posture score.
Actual Result	Partial score (specific REBA posture score for specific body parts only e.g neck)
Test Status	Partially pass

Table 19; Test Case Table 6

Test Case ID	TC_006
Test Case Name	File Export REBA
Test Scenario	To test whether it will add another column for the overall REBA posture score.
Steps	- Modify the joint angles calculation so that it also

	includes the REBA posture score calculation. - Call the function to export the joint angles & REBA posture score to an .csv file in the calculate joint angles class. - Call and run the method.
Expected Result	Generate an accurate joint angles & REBA posture score.
Actual Result	Partial result (only joint angles for now)
Test Status	Partially pass

Table 20; Test Case Table 7

Test Case ID	TC_007
Test Case Name	Generate Calibration File
Test Scenario	To test whether it will generate a suitable calibration file.
Steps	- Subject will record themselves holding a charuco board. - Move the board around slowly until at least the board is visible in 2 cameras. - Do it for a few seconds.
Expected Result	Generate a suitable calibration file for that recording.
Actual Result	Same as expected result
Test Status	Pass

Table 21; Test Case Table 8

Test Case ID	TC_008
Test Case Name	Motion Capture

Test Scenario	To test whether does the system still retains the original motion capture process after the modules addition.
Steps	- Subject will repeat the calibration procedure. - After so, the subject can put the charuco board down and start moving and posing. - Do so for a few seconds or minutes
Expected Result	Generated an accurate 3D representation of the subject alongside the annotated videos and the output files.
Actual Result	Same as expected result
Test Status	Pass

5.3. Test Results Summary

Table 22; Test Case Results Summary Table

Test Case ID	Test Case Name	Test Status	Test Outcome
TC_001	CSV Read Test	Yes	Pass
TC_002	Angle Calculation Method	Yes	Pass
TC_003	Verify File Path in Textbox	Yes	Pass
TC_004	File Export	Yes	Pass
TC_005	Calculate REBA posture score	Not Yet	Not Yet
TC_006	File Export REBA	Not Yet	Not Yet
TC_007	Generate Calibration File	Yes	Pass
TC_008	Motion Capture	Yes	Pass

**for the test that hasn't been verified to pass yet or partially pass. Due to time mismanagement and other issues, it is not possible to present it right now. But will surely finish implementing the REBA calculation module and other related modules later on during the FYP presentation.*

5.4. Evidence of the Works

Table 23; Test Cases Evidence Reference Table

Test Case ID	Test Outcome	Evidence
TC_001	Pass	*Figure 48; TC 001 Output*
TC_002	Pass	*Figure 48; TC_002 Output/Evidence*
TC_003	Pass	*Figure 49 & figure 50; TC_003 File Validation a) & b)*
TC_004	Pass	*Figure 51; TC_004 Output/Evidence*
TC_005	Not Yet	-
TC_006	Not Yet	-
TC_007	Pass	*Figure 52; TC_008 Output/Evidence*
TC_008	Pass	*Figure 53 & figure 54; TC_009 Calibration Evidence/Configuration & 3D Representation*

5.5. Critical Evaluation of the System and Comparison with Other Systems

Freemocap++ certainly has some advantages as well as some disadvantages. Below I will discuss the current extended system critical evaluation analysis.

5.5.1. Advantages:

- Markerless technology: increase user comfort and convenience and reduce time and effort required for a setup.
- Multi camera configuration: increases accuracy of the coordinates captured as well as the joint angles calculated.
- REBA posture calculation: in addition to the joint angles' calculation, it also has a functionality that can calculate the subject REBA score, providing feedback on ergonomic risks.

5.5.2. Disadvantages:

- Complexity of setup: calibration process requires a careful and thorough approach to ensure an accurate 3D reconstruction and the system's performance can be affected by environmental factors such as lighting conditions and background clutter.
- Cost: Requires multiple high-resolution cameras and powerful computing resources, making it potentially more expensive than single-camera systems or simpler motion capture technologies.
- Data processing: Processing data from multiple cameras to calculate joint angles and REBA scores requires significant computational power, which may limit the use of it with using an old computer.

5.5.3. Strengths:

- Non-Invasive Measurement: Enhances user experience by avoiding physical markers and allowing natural movement, which is crucial for accurate motion analysis in various applications.

- Enhanced Accuracy & Precision: The use of multiple cameras significantly improves the precision of joint angle measurements and the reliability of posture assessments.
- Integration of Ergonomic Assessment: Combining motion capture with ergonomic evaluation (REBA scores) provides a comprehensive tool for analyzing both movement quality and ergonomic risk, offering a dual benefit in a single system.

5.5.4. Limitations:

- Environmental Constraints: Requires a controlled environment for optimal performance. Variations in lighting, presence of reflective surfaces, and background activity can interfere with data capture.
- Scalability: While effective for smaller, controlled spaces, scaling the system for larger environments or multiple simultaneous subjects can be challenging and resource intensive.

5.5.5. Comparison with other systems:

- Marker-Based Systems: While marker-based systems may be less comfortable and more intrusive, they often provide higher accuracy in joint tracking due to the direct measurement of marker positions. However, marker-less systems like the one evaluated here offer a more natural and less intrusive user experience.
- Single-Camera Marker-Less Systems: Single-camera systems are easier to set up and less expensive but generally offer lower accuracy and are more prone to occlusion issues. The multi-camera approach mitigates these issues but at a higher cost and complexity.
- Wearable Sensor Systems: Wearable systems using inertial sensors can provide accurate motion data with a simple setup but lack the detailed spatial information that a multi-camera system can capture. Additionally, wearables still involve attaching devices to the body, which can be less comfortable.

5.6. Summary of Important Findings

This report evaluates the capabilities and performance of a "*Freemocap++*: Marker-Less Multi-Camera Motion Capture System with Joint Angles & REBA Calculation." The findings indicate that the multi-camera configuration significantly enhances the accuracy and precision of joint angle measurements. By providing multiple perspectives, the system effectively reduces occlusion issues and improves depth perception, resulting in more reliable motion capture data.

The marker-less design eliminates the need for physical markers, offering a more comfortable and natural experience for users. This feature is particularly beneficial for applications requiring unrestricted movement, such as sports science, rehabilitation, and ergonomic studies. Additionally, the integration of the REBA (Rapid Entire Body Assessment) posture scoring system allows for an effective ergonomic evaluation. This dual functionality enables immediate feedback on ergonomic risks, making the system valuable for workplace safety assessments and ergonomic research.

However, the system's performance is influenced by environmental factors such as lighting conditions and background clutter. Optimal results are achieved in controlled environments, highlighting the need for careful setup and calibration. Furthermore, while the system offers advanced capabilities, it also involves significant complexity and cost. The calibration process for multiple cameras is intricate and time-consuming, and the requirement for multiple high-resolution cameras and powerful computing resources contributes to higher overall expenses.

Compared to traditional marker-based systems, the marker-less approach offers greater user comfort and convenience, though marker-based systems may still provide slightly higher accuracy due to direct marker tracking. In contrast to single-camera marker-less systems, the multi-camera setup provides superior accuracy and reduces occlusion issues, albeit at the cost of increased setup complexity and expense. When compared to wearable sensor systems, the multi-camera system captures more detailed spatial information without the need for attached sensors, although wearables offer simpler setup and lower cost.

The system's versatility makes it suitable for a wide range of applications, including motion analysis in sports, clinical rehabilitation, ergonomic evaluations, and animation. Its ability to provide detailed joint angles and ergonomic scores in real-time adds significant value across these domains. However, scaling the system for larger spaces or multiple subjects presents challenges. The need for extensive calibration and increased computational load may limit its scalability.

6 CONCLUSION & FUTURE WORK

6.1. Summary of Results

In summary, the "*Freemocap++*: an AI-Based Multi-Camera Marker-Less Motion Capture System with Joint Angles & REBA Posture Score Calculation" demonstrated significant advancements in motion capture accuracy and ergonomic assessment capabilities. The system successfully utilized multiple cameras to achieve a high degree of precision in joint angle measurements. This multi-perspective approach effectively mitigated common issues related to occlusions and depth perception, providing a reliable and detailed 3D reconstruction of human motion.

During the testing phase, the system was able to capture motion data with a high level of accuracy across various scenarios and movements. The integration of the REBA posture scoring algorithm allowed for a quick assessment of ergonomic risks, which was validated against standard ergonomic evaluation methods. The REBA scores generated by the system correlated well with manual assessments, confirming the system's reliability in ergonomic evaluation.

The marker-less design proved to be highly beneficial in terms of user experience. Subjects reported greater comfort and freedom of movement compared to traditional marker-based systems. This non-invasive approach also reduced setup time and complexity, making the system more user-friendly and accessible for repeated use in different environments.

However, the system's performance was found to be sensitive to environmental conditions. Optimal results were achieved in controlled settings with consistent lighting and minimal background clutter. Variations in these factors sometimes led to minor inaccuracies in the captured data, indicating the need for further refinement in the system's environmental adaptability.

In terms of computational requirements, the system demonstrated efficient processing capabilities with my current setup, handling real-time data capture and analysis with reasonable processing time. The use of sophisticated algorithms for joint angle calculation and REBA scoring ensured that the system provided immediate and actionable feedback. Nevertheless, the computational load was significant, necessitating the use of high-performance computing resources to maintain its high performance.

In conclusion, *'Freemocap++'* delivers a robust solution for accurate motion capture and ergonomic assessment. The system's strengths in non-invasive design, real-time feedback, and precise data capture make it a valuable tool across various applications, from sports science to workplace ergonomics. Despite certain environmental sensitivities and computational demands, the system's overall performance and user experience highlight its potential for wide adoption and further development.

6.2. Critical Evaluation of The Results of The Project

6.2.1. Achievement of Goals & Outlines of the Success of The Project

Goals & Objectives Met:

- Successfully implemented additional data processing capability to the existing system *'Freemocap'*.
- Successfully implemented a frame-by-frame data visualization by showing the processed data through the GUI frame by frame as a better method to show efficient information to the user.
- Successfully integrated additional modules and new functionalities to the existing system modules.
- High accuracy in joint angle measurements was achieved by multiple cameras by a reliable AI tracking model.
- Integration of REBA posture scoring allowed for a quick ergonomic assessment with minimal user interaction.

Unmet Goals & Objectives:

- I believe I have achieved all the goals & objectives that I set initially when starting this project, so there are no unmet goals.

6.2.2. Software Fitness for a Purpose

Intended Purpose: The software is mainly designed for animation since there is a function that can export the motion capture data to Blender, but as of now I see that it has the potential to have various applications in in sports science, rehabilitation, and workplace ergonomics, providing detailed motion analysis and ergonomic risk assessments.

Practical Application: Feedback from supervisor and other lecturer confirmed that the software meets its intended purpose effectively, and even has a practical application for ergonomic assessors to lessen their work by using this system.

6.2.3. Design and Implementation Practices

Design Practices:

- The design leveraged an existing system that utilizes advanced deep learning algorithms (Mediapipe Blazepose CNNs and Anipose) for a markerless motion capture system and ergonomic assessment, ensuring high accuracy and performance.

Implementation Practices:

- The implementation involved the use of action cameras and a fairly powerful computing resources and successfully overcoming the challenges of processing large volumes of data using OOP.

6.2.4. Strengths of the Extended System

Non-Invasive Measurement: Enhances user experience by avoiding physical markers and allowing natural movement, which is crucial for accurate motion analysis in various applications.

Enhanced Accuracy & Precision: The use of multiple cameras significantly improves the precision of joint angle measurements and the reliability of posture assessments.

Integration of Ergonomic Assessment: Combining motion capture with ergonomic evaluation (REBA scores) provides a comprehensive tool for analyzing both movement quality and ergonomic risk, offering a dual benefit in a single system.

6.2.5. Limitations and Areas for Improvement/Future Works

Identified Limitations:

- Sensitivity to environmental conditions such as lighting and background clutter.
- Still involves user interaction in my modules.

Potential Improvements:

- Enhancing the system's adaptability to different environments through improved calibration and algorithmic adjustments.
- Automate everything and try to lessen user interaction as little as possible (ex; no need for the text box and condition checkboxes)
- Implement a computer vision method that can automatically recognize the subject body parts condition as stated in the REBA worksheet (ex; is the neck twisted or side bending?)

REFERENCES

CMU-Perceptual-Computing-Lab. (n.d.). *CMU-Perceptual-Computing-Lab/openpose: OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation*. GitHub. https://github.com/CMU-Perceptual-Computing-Lab/openpose

GeeksforGeeks. (2023a, June 20). *Software engineering: Classification of software requirements*. GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/

GeeksforGeeks. (2023b, August 1). *R-CNN: Region Based CNNS*. GeeksforGeeks. https://www.geeksforgeeks.org/r-cnn-region-based-cnns/

Google. (n.d.). *Google/Mediapipe: Cross-platform, customizable ML solutions for live and streaming media.* GitHub. https://github.com/google/mediapipe

Gupta, R. (2019, October 11). *6 deep learning models-when should you use them?*. Medium. https://towardsdatascience.com/6-deep-learning-models-10d20afec175

Markerless Motion Capture. iPi Soft. (n.d.). https://ipisoft.com/

Nakano, N., Sakura, T., Ueda, K., Omura, L., Kimura, A., Iino, Y., Fukashiro, S., & Yoshioka, S. (2020, April 14). *Evaluation of 3D markerless motion capture accuracy using openpose with multiple video cameras*. Frontiers. https://www.frontiersin.org/articles/10.3389/fspor.2020.00050/full

UML class diagram - javatpoint. www.javatpoint.com. (n.d.). https://www.javatpoint.com/uml-class-diagram

What is Class Diagram?. What is class diagram? (n.d.). https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

GeeksforGeeks. (2023, December 14). *Functional vs Non Functional Requirements*. GeeksforGeeks. https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/

YouTube. (2021a, April 6). *Latest pose estimation realtime (24 fps) using CPU | computer vision | opencv python*. YouTube. https://www.youtube.com/watch?v=brwgBf6VB0I

YouTube. (2021b, August 2). *Real-time 3D pose detection & pose classification | mediapipe | opencv | python*. YouTube. https://www.youtube.com/watch?v=aySurynUNAw&list=LL&index=1&t=2070s&pp=gAQBiAQB

Anipose. (n.d.). https://anipose.readthedocs.io/en/latest/#fromHistory

Freemocap. (n.d.). *Freemocap/freemocap: Free motion capture for everyone 💀✨*. GitHub. https://github.com/freemocap/freemocap?tab=AGPL-3.0-1-ov-file

Google. (n.d.). *Pose landmark detection guide | google ai edge | google for developers*. Google. https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

Model card blazepose ghum 3D (June 2021). (n.d.-a). https://developers.google.com/ml-kit/images/vision/pose-detection/pose_model_card.pdf

Rapid entire body assessment (reba). (n.d.-b). http://ergo-plus.com/wp-content/uploads/REBA-A-Step-by-Step-Guide.pdf

Simplilearn. (2024, April 18). *Top down approach vs. bottom up approach: Understanding the differences*. Simplilearn.com. https://www.simplilearn.com/top-down-approach-vs-bottom-up-approach-article#:~:text=Top%2DDown%3A%20More%20effective%20in,quick%20decision%2Dmaking%20are%20needed.

Google. (n.d.-a). *Face landmark detection guide | google ai edge | google for developers*. Google. https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker

YouTube. (2021, September 29). *Anipose: A toolkit for robust markerless 3D pose estimation*. YouTube. https://www.youtube.com/watch?v=hQAFI5o2-F4

APPENDICES

```
PS C:\Users\Arya\freemocap> & C:/Users/Arya/anaconda3/envs/FYP/python.exe c:/Users/Arya/freemocap/scripts/read_csv.py
[-238.94, -223.85, -219.08, -225.99, -237.42, -246.68, -253.83, -262.09, -271.71, -280.36, -287.73, -295.57, -303.77,
57.62, -364.5, -369.25, -373.61, -377.61, -380.95, -383.53, -384.75, -384.15, -382.56, -381.24, -380.3, -378.59, -374.
25, -324.04, -322.28, -319.48, -316.4, -314.22, -313.78, -315.15, -317.33, -318.9, -319.53, -320.38, -322.52, -325.32,
 -362.25, -364.3, -364.39, -364.16, -364.48, -365.16, -366.02, -367.26, -368.84, -370.32, -371.38, -372.21, -373.39, -
361.74, -357.33, -352.32, -346.92, -341.5, -336.8, -333.4, -331.07, -328.84, -325.6, -321.03, -315.92, -311.28, -307.6
3, -296.05, -296.05, -296.73, -297.73, -298.31, -298.22, -297.92, -297.53, -296.37, -293.92, -290.79, -287.96, -285.4,
-256.34, -256.07, -256.68, -258.91, -264.5, -274.77, -289.26, -305.9, -322.18, -336.39, -348.07, -358.2, -369.29, -383
19, -523.57, -525.71, -523.54, -517.62, -514.09, -517.02, -523.21, -527.18, -527.86, -526.86, -523.96, -518.61, -513.7
, -451.24, -445.33, -446.03, -451.6, -457.69, -461.89, -464.02, -464.58, -464.13, -462.29, -457.63, -450.07, -442.5, -
9.64, -405.05, -399.46, -390.35, -379.49, -370.72, -365.69, -362.59, -358.61, -352.8, -346.26, -340.48, -335.84, -331.
-279.7, -279.54, -280.66, -283.88, -288.61, -293.03, -297.04, -302.37, -309.14, -314.73, -316.96, -316.22, -314.05, -3
0.38, -265.37, -257.07, -249.91, -247.76, -249.2, -249.55, -245.84, -238.99, -232.2, -227.84, -225.77, -224.36, -222.4
 -211.54, -210.56, -209.65, -209.63, -210.02, -210.37, -211.26, -212.8, -213.39, -211.7, -209.18, -208.55, -209.9, -21
```

Figure 27; TC 001 Output

neck	trunk	upper_left	upper_rigl
10.99846	19.68788	28.16834	18.00321
14.99886	18.05106	27.96148	19.06823
18.43931	16.83665	29.27927	21.20737
20.88123	15.81062	31.28496	23.92167
22.18354	14.78108	31.97316	25.79605
22.90063	14.02434	30.85198	26.32606
23.25751	13.62714	29.29832	26.26363
22.86905	13.11502	28.6579	26.0699
21.83048	12.11888	28.75334	25.54278
20.90915	10.80209	28.42729	24.80507

Figure 28; TC_002 Output/Evidence

C:\Users\Arya\freemocap_data\recording_sessions\sample2\output_data\angles.csv

File exists: C:\Users\Arya\freemocap_data\recording_sessions\sample2\output_data\angles.csv

Usually located in 'C:\Users\%USER%\freemocap_data\recording_sessions\%SESSION%\output_data\angles.csv'

Submit

Figure 29; TC_003 File Validation a)

C:\Users\Arya\freemocap_data\recording_sessions\sample2\output_data\angless.csv

File does not exist: C:\Users\Arya\freemocap_data\recording_sessions\sample2\output_data\angless.csv

Usually located in 'C:\Users\%USER%\freemocap_data\recording_sessions\%SESSION%\output_data\angles.csv'

Submit

Figure 30; TC_003 File Validation b)

	A	B	C	D	E	F	G	H	I	J
1	neck	trunk	upper_left	upper_rigl	upper_left	upper_rigl	lower_left	lower_rigl	lower_left	lower_righ
2	10.99846	19.68788	28.16834	18.00321	29.72814	21.83622	36.27795	99.90589	29.35166	22.54226
3	14.99886	18.05106	27.96148	19.06823	28.66529	21.10668	30.02002	101.5352	24.62749	23.50935
4	18.43931	16.83665	29.27927	21.20737	28.75312	21.21755	25.79346	102.6702	21.24605	24.80066
5	20.88123	15.81062	31.28496	23.92167	29.37198	22.32104	22.60397	103.0142	18.94493	26.04336
6	22.18354	14.78108	31.97316	25.79605	29.30588	23.7869	20.24686	102.5273	17.26113	26.56798
7	22.90063	14.02434	30.85198	26.32606	27.98701	24.37747	19.77496	101.5634	16.28686	26.34827
8	23.25751	13.62714	29.29832	26.26363	26.05256	23.38562	22.10639	100.3986	16.01147	26.30368
9	22.86905	13.11502	28.6579	26.0699	24.44881	21.39578	28.18135	98.95069	15.82993	26.42441
10	21.83048	12.11888	28.75334	25.54278	23.28633	19.33916	37.354	97.41985	15.16042	25.47901
11	20.90915	10.80209	28.42729	24.80507	22.22919	17.33403	44.64941	96.29003	14.13378	24.46137

Figure 31; TC_004 Output/Evidence

Name	Size	Type
annotated_videos		File Folder
sesh_2022-09-19_16_16_50_in_cl...	64.77 MiB	MPEG-4 video
sesh_2022-09-19_16_16_50_in_cl...	75.75 MiB	MPEG-4 video
sesh_2022-09-19_16_16_50_in_cl...	69.97 MiB	MPEG-4 video
output_data		File Folder
center_of_mass		File Folder
segmentCOM_frame_joint_xyz....	363.69 KiB	unknown
total_body_center_of_mass_x...	26.09 KiB	unknown
raw_data		File Folder
mediapipe2dData_numCams_n...	42.07 MiB	unknown
mediapipe3dData_numCams_n...	14.02 MiB	unknown
mediapipe3dData_numFrames_...	4.67 MiB	unknown
mediapipe3dData_numFrames_...	14.02 MiB	unknown
angles.csv	199.99 KiB	CSV document
mediapipe_body_3d_xyz.csv	1.96 MiB	CSV document
mediapipe_body_3d_xyz.npy	857.09 KiB	unknown
mediapipe_face_3d_xyz.csv	28.25 MiB	CSV document
mediapipe_face_3d_xyz.npy	12.12 MiB	unknown
mediapipe_left_hand_3d_xyz.csv	1.24 MiB	CSV document
mediapipe_left_hand_3d_xyz.npy	545.47 KiB	unknown
mediapipe_names_and_connectio...	17.33 KiB	JSON document
mediapipe_right_hand_3d_xyz.csv	1.26 MiB	CSV document
mediapipe_right_hand_3d_xyz.npy	545.47 KiB	unknown
mediaPipeSkel_3d_body_hands_fa...	14.02 MiB	unknown
recording_parameters.json	1.07 KiB	JSON document
synchronized_videos		File Folder
sesh_2022-09-19_16_16_50_in_cl...	51.10 MiB	MPEG-4 video
sesh_2022-09-19_16_16_50_in_cl...	56.72 MiB	MPEG-4 video
sesh_2022-09-19_16_16_50_in_cl...	51.66 MiB	MPEG-4 video
2022-09-19_16_16_50_in_class_jsm_...	1.59 KiB	TOML document
sample2_by_frame.json	113.51 MiB	JSON document
sample2_by_trajectory.csv	33.40 MiB	CSV document
sample2_frame_name_xyz.npy	14.02 MiB	unknown

Figure 32; TC_008 Output/Evidence

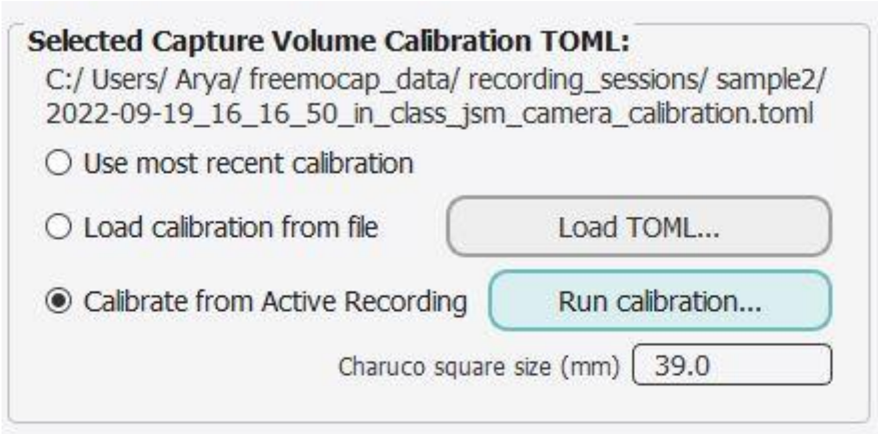


Figure 33; TC_009 Calibration Evidence/Configuration

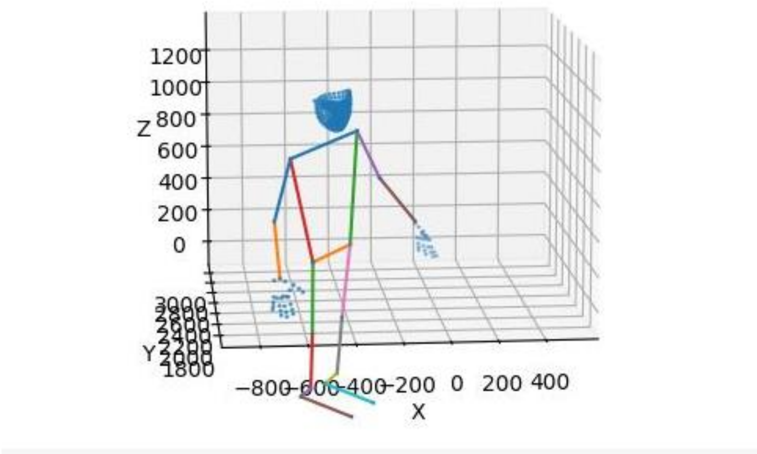
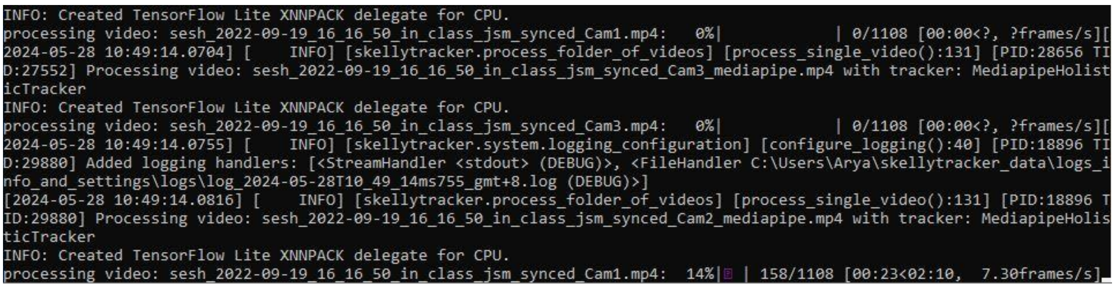


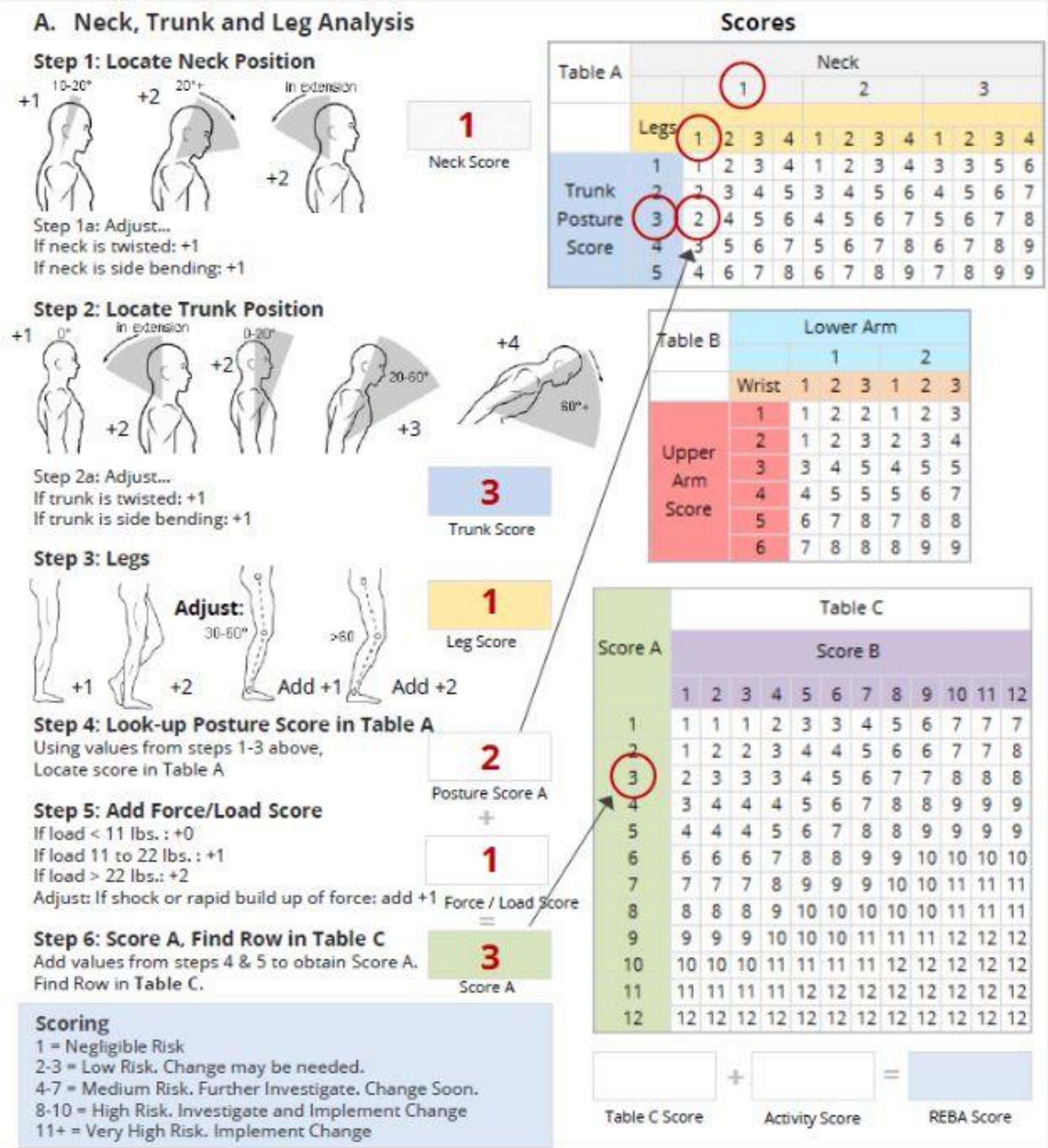
Figure 34; TC_009 3D Representation



```
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
processing video: sesh_2022-09-19_16_16_50_in_class_jsm_synced_Cam1.mp4:   0%|            | 0/1108 [00:00<?, ?frames/s][
2024-05-28 10:49:14.0704] [    INFO] [skellytracker.process_folder_of_videos] [process_single_video():131] [PID:28656 TI
D:27552] Processing video: sesh_2022-09-19_16_16_50_in_class_jsm_synced_Cam3_mediapipe.mp4 with tracker: MediapipeHolist
icTracker
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
processing video: sesh_2022-09-19_16_16_50_in_class_jsm_synced_Cam3.mp4:   0%|            | 0/1108 [00:00<?, ?frames/s][
2024-05-28 10:49:14.0755] [    INFO] [skellytracker.system.logging_configuration] [configure_logging():40] [PID:18896 TI
D:29880] Added logging handlers: [<StreamHandler <stdout> (DEBUG)>, <FileHandler C:\Users\Arya\skellytracker_data\logs_i
nfo_and_settings\logs\log_2024-05-28T10_49_14ms755_gmt+8.log (DEBUG)>]
[2024-05-28 10:49:14.0816] [    INFO] [skellytracker.process_folder_of_videos] [process_single_video():131] [PID:18896 T
ID:29880] Processing video: sesh_2022-09-19_16_16_50_in_class_jsm_synced_Cam2_mediapipe.mp4 with tracker: MediapipeHolis
ticTracker
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
processing video: sesh_2022-09-19_16_16_50_in_class_jsm_synced_Cam1.mp4:  14%|▏ | 158/1108 [00:23<02:10,  7.30frames/s]
```

Figure 35; Motion Capture Process (Terminal)

Figure 36; Video Recording Attempt



A. Neck, Trunk and Leg Analysis

Step 1: Locate Neck Position

+1 10-20° +2 20°+ in extension +2

1 Neck Score

Step 1a: Adjust...
If neck is twisted: +1
If neck is side bending: +1

Step 2: Locate Trunk Position

+1 0° in extension +2 0-20° +2 20-60° +3 +4 60°+

Step 2a: Adjust...
If trunk is twisted: +1
If trunk is side bending: +1

3 Trunk Score

Step 3: Legs

+1 +2 Adjust: 30-60° Add +1 >60 Add +2

1 Leg Score

Step 4: Look-up Posture Score in Table A
Using values from steps 1-3 above,
Locate score in Table A

2 Posture Score A

Step 5: Add Force/Load Score
If load < 11 lbs. : +0
If load 11 to 22 lbs. : +1
If load > 22 lbs.: +2
Adjust: If shock or rapid build up of force: add +1

1 Force / Load Score

Step 6: Score A, Find Row in Table C
Add values from steps 4 & 5 to obtain Score A.
Find Row in Table C.

3 Score A

Scoring
1 = Negligible Risk
2-3 = Low Risk. Change may be needed.
4-7 = Medium Risk. Further Investigate. Change Soon.
8-10 = High Risk. Investigate and Implement Change
11+ = Very High Risk. Implement Change

Scores

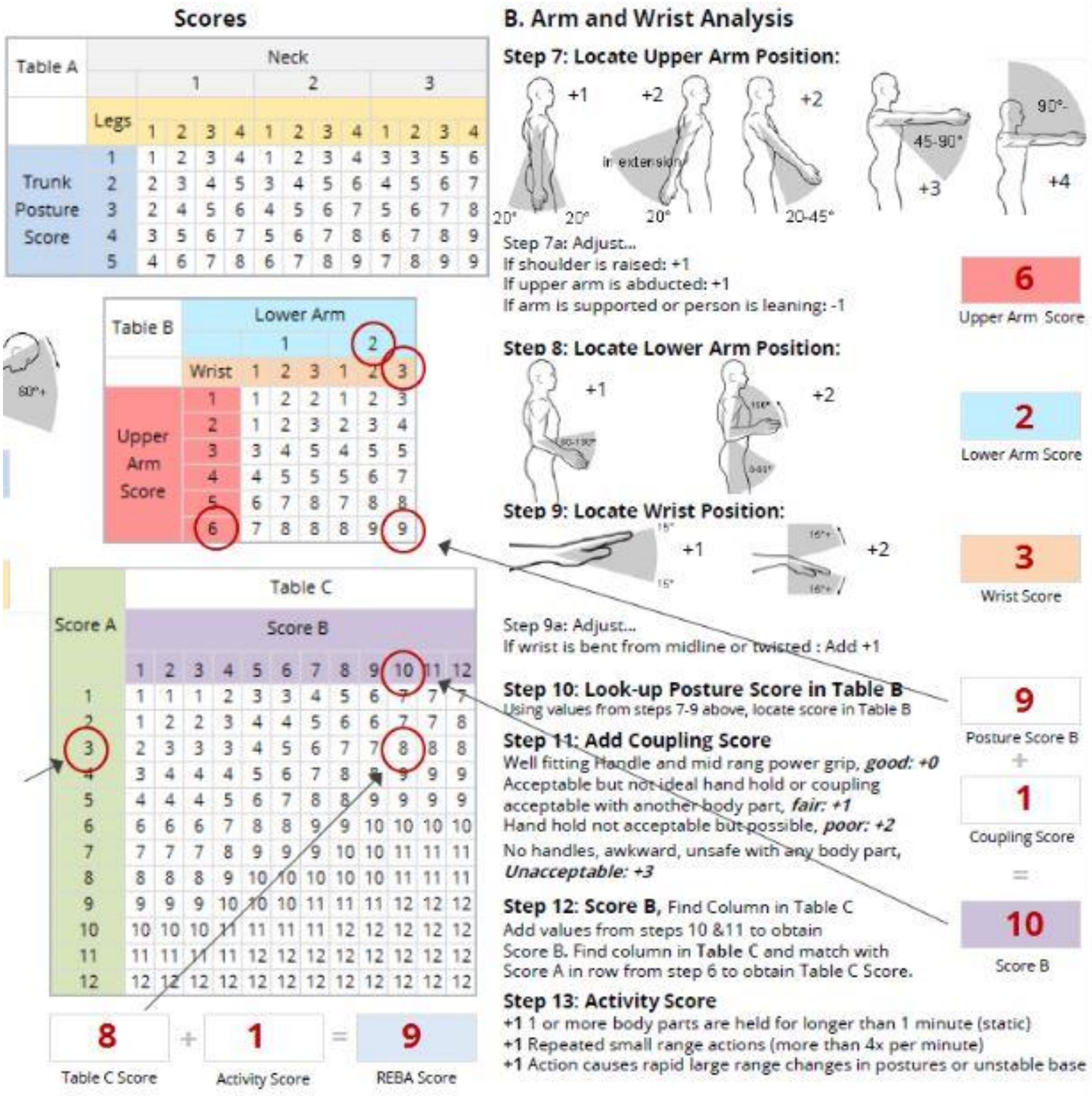
Table A	Neck	1				2				3			
	Legs	1	2	3	4	1	2	3	4	1	2	3	4
Trunk Posture Score	1	1	2	3	4	1	2	3	4	3	3	5	6
	2	2	3	4	5	3	4	5	6	4	5	6	7
	3	2	4	5	6	4	5	6	7	5	6	7	8
	4	3	5	6	7	5	6	7	8	6	7	8	9
	5	4	6	7	8	6	7	8	9	7	8	9	9

Table B	Lower Arm	1			2		
	Wrist	1	2	3	1	2	3
Upper Arm Score	1	1	2	2	1	2	3
	2	1	2	3	2	3	4
	3	3	4	5	4	5	5
	4	4	5	5	5	6	7
	5	6	7	8	7	8	8
	6	7	8	8	8	9	9

Table C	Score B											
Score A	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	2	3	3	4	5	6	7	7	7
2	1	2	2	3	4	4	5	6	6	7	7	8
3	2	3	3	3	4	5	6	7	7	8	8	8
4	3	4	4	4	5	6	7	8	8	9	9	9
5	4	4	4	5	6	7	8	8	9	9	9	9
6	6	6	6	7	8	8	9	9	10	10	10	10
7	7	7	7	8	9	9	9	10	10	11	11	11
8	8	8	8	9	10	10	10	10	10	11	11	11
9	9	9	9	10	10	10	11	11	11	12	12	12
10	10	10	10	11	11	11	11	12	12	12	12	12
11	11	11	11	11	12	12	12	12	12	12	12	12
12	12	12	12	12	12	12	12	12	12	12	12	12

Table C Score + Activity Score = REBA Score

Figure 37; REBA Sheet A



Scores

Table A		Neck											
		1				2				3			
	Legs	1	2	3	4	1	2	3	4	1	2	3	4
Trunk Posture Score	1	1	2	3	4	1	2	3	4	3	3	5	6
	2	2	3	4	5	3	4	5	6	4	5	6	7
	3	2	4	5	6	4	5	6	7	5	6	7	8
	4	3	5	6	7	5	6	7	8	6	7	8	9
	5	4	6	7	8	6	7	8	9	7	8	9	9

Table B		Lower Arm					
		1			2		
	Wrist	1	2	3	1	2	3
Upper Arm Score	1	1	2	2	1	2	3
	2	1	2	3	2	3	4
	3	3	4	5	4	5	5
	4	4	5	5	5	6	7
	5	6	7	8	7	8	8
	6	7	8	8	8	9	9

Score A	Table C											
	Score B											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	2	3	3	4	5	6	7	7	7
2	1	2	2	3	4	4	5	6	6	7	7	8
3	2	3	3	3	4	5	6	7	7	8	8	8
4	3	4	4	4	5	6	7	8	8	9	9	9
5	4	4	4	5	6	7	8	8	9	9	9	9
6	6	6	6	7	8	8	9	9	10	10	10	10
7	7	7	7	8	9	9	9	10	10	11	11	11
8	8	8	8	9	10	10	10	10	10	11	11	11
9	9	9	9	10	10	10	11	11	11	12	12	12
10	10	10	10	11	11	11	11	12	12	12	12	12
11	11	11	11	11	12	12	12	12	12	12	12	12
12	12	12	12	12	12	12	12	12	12	12	12	12

8 + 1 = 9

Table C Score, Activity Score, REBA Score

B. Arm and Wrist Analysis

Step 7: Locate Upper Arm Position:

Step 7a: Adjust...
If shoulder is raised: +1
If upper arm is abducted: +1
If arm is supported or person is leaning: -1

6
Upper Arm Score

Step 8: Locate Lower Arm Position:

2
Lower Arm Score

Step 9: Locate Wrist Position:

3
Wrist Score

Step 9a: Adjust...
If wrist is bent from midline or twisted : Add +1

Step 10: Look-up Posture Score in Table B
Using values from steps 7-9 above, locate score in Table B

9
Posture Score B
+

Step 11: Add Coupling Score
Well fitting Handle and mid rang power grip, ***good: +0***
Acceptable but not ideal hand hold or coupling acceptable with another body part, ***fair: +1***
Hand hold not acceptable but possible, ***poor: +2***
No handles, awkward, unsafe with any body part, ***Unacceptable: +3***

1
Coupling Score
=

Step 12: Score B, Find Column in Table C
Add values from steps 10 &11 to obtain Score B. Find column in **Table** C and match with Score A in row from step 6 to obtain Table C Score.

10
Score B

Step 13: Activity Score
+1 1 or more body parts are held for longer than 1 minute (static)
+1 Repeated small range actions (more than 4x per minute)
+1 Action causes rapid large range changes in postures or unstable base

Figure 38; REBA Sheet B