eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Bachelor Thesis

Degree in Computer Engineering

Computer Science

3D IDLE Motion Dataset Generation and Analysis

Abdurrahim Ali Ali

Advisors
Jon Vadillo Jueguen
Igor Rodríguez Rodríguez

June 23, 2024

Acknowledgements

I would like to express my deepest gratitude to my director, Jon Vadillo, and my co-director, Igor Rodríguez, for their guidance, support, and encouragement throughout the course of this project. Their expertise and dedication have been instrumental in the successful completion of this project.

I would also like to thank the doctoral researcher, Eneko Atxa, for letting me participate in his research and for allowing me to base my project on his research topic.

Finally, I would like to thank my family and friends for their continuous support and encouragement, not only during this project but also throughout the whole journey. Their unwavering support has been invaluable to me and leads me to where I am today.

Thank you all.

Summary

The demand for natural and immersive experiences in video games and animation movies has fostered advancements in animation techniques. Motion capture (MoCap) allows for realistic character movements by recording human actions and translating them to 3D models. However, capturing subtle, unconscious movements (IDLE animations) is challenging due to limitations of traditional MoCap setups and the use of professional actors.

This thesis aims to reduce this gap and we present a methodology for developing a cost-effective and non-intrusive MoCap recording process specifically designed to capture IDLE animations. This methodology prioritizes affordability, ease of replication, and minimizing the impact on the recorded movements, allowing anyone to generate their own MoCap data.

Furthermore, we introduce the first-ever dataset of non-acted and acted 3D IDLE motions generated using the developed methodology. This dataset encompasses a variety of common IDLE animations for versatility in animation applications. A detailed analysis of the generated dataset is conducted to understand the characteristics of the captured motions. All analysis tools and scripts used will be made publicly available in a repository for further exploration and potential use with other MoCap data.

This works lays a foundation for future research, seeking to enhance realism and immersion in animation applications such as video games, animation movies, and virtual reality experiences.

Resumen

La demanda de experiencias naturales y envolventes en videojuegos y películas de animación ha impulsado avances en las técnicas de animación. La captura de movimiento (MoCap) permite obtener movimientos realistas de los personajes grabando las acciones humanas y traduciéndolas a modelos 3D. Sin embargo, capturar movimientos sutiles e inconscientes (animaciones IDLE) supone un reto debido a las limitaciones de los sistemas MoCap tradicionales y al uso de actores profesionales.

Esta tesis tiene como objetivo reducir esta brecha y presentamos una metodología para desarrollar un proceso de grabación MoCap rentable y no intrusiva diseñada específicamente para capturar animaciones IDLE. Esta metodología prioriza la asequibilidad, la facilidad de replicación y la minimización del impacto en los movimientos grabados, permitiendo a cualquiera generar sus propios datos MoCap.

Además, presentamos el primer conjunto de datos de movimientos IDLE en 3D, actuados y no actuados, generados utilizando la metodología desarrollada. Este conjunto de datos abarca una variedad de animaciones IDLE comunes para una mayor versatilidad en aplicaciones de animación. Se realiza un análisis detallado del conjunto de datos generado para comprender las características de los movimientos capturados. Todas las herramientas de análisis y scripts utilizados se pondrán a disposición del público en un repositorio para su posterior exploración y uso potencial con otros datos MoCap.

Este trabajo sienta las bases para futuras investigaciones, buscando mejorar el realismo y la inmersión en aplicaciones de animación como videojuegos, películas de animación y experiencias de realidad virtual.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Interactive environments, open-world video games or animation movie scenes would not be as rich if it were not for the small details that are often overlooked but essential for the feeling of naturality and immersion. Throughout the years, an increasing need for providing better, more realistic and lifelike experiences in these industries has arisen. The evolution of these characteristics has been a gradual process, driven significantly by the advancements in technology, being the introduction of 3D graphics the central milestone, which revolutionized industries such as the video games industry or animation industry, who often pursue realism and engrossment in their products for the end-user.

When playing a video game or watching an animation movie, an expectation of human-familiar behaviors is created, not only from the main character but also from the characters who compose the scenes. These conducts are induced into the characters by animating their skeletons, which are a set of bones connected to each other that are translated and rotated in order to create motion. Achieving such authentic animations by manually positioning and rotating each bone for eact time step is not only an intensive labor but also a time-consuming task which will frequently fall short of the desired outcome.

Therefore, the question that arises is: which technique is behind to achieve such realistic animations? Both manual animation and motion capture (commonly abbreviated as MoCap) systems are used to achieve this realism. In a nutshell, MoCap is a technique used to record the movement of objects or people, which is then translated into a digital model [1]. Generally, in order to capture precise motions, a complex setup is required, which includes multiple cameras, a suit with markers, and a controlled environment. The subject is then recorded while performing the desired actions, and 3D data is captured and processed to create a virtual skeleton which can be later applied to a 3D model using a capable 3D software such as Blender and will mimic the movements of the subject. Modern workflows often combine these techniques, MoCap systems are used as a foundation and, since manual animation grants more control over the final result, it is used to refine the animations.

But, why motion capture? What does it add to the table? Well, a lot of data is collected from the subject's movements, not only the desired actions but also the small details which are the key ingredient that add the essence of realism in the animation. Consider a character shifting its weight from one leg to another, breathing, adjusting their posture or casually looking around their surroundings. These are the so called IDLE animations [2, 3], which are movements performed when the character is not "doing any action".

In Figure 1.1, an example of an IDLE animation sequence is shown. The character, Nico, from the video game Grand Theft Auto IV [1], is looking around while standing still. The animation is triggered when the player does not interact with the character for a certain amount of time, and it is intended to make the character feel more alive and natural.

Figure 1.1: Grand Theft Auto IV's main character Nico's IDLE animation sequence looking backwards.

Now, these IDLE motions require an actor to perform them, but what if we need to create a scene with a crowd of people? Recording a separate clip for each individual would be immensely labor-intensive, and the task would become daunting. Additionally, recording a clip for a specific actor has its limitations: it captures a sequence that is limited in both time and the variety of gestures. This approach is not easily extensible to other characters or scenarios.

This is where machine learning comes to aid, by automating the process of generating new animations from the motion captured data, utilizing a comprehensive dataset for learning. Through machine learning, it becomes possible to generate a diverse set of animations, reducing the need for extensive manual recording and enabling the creation of more dynamic and varied scenes.

However, there is no publicly available 3D IDLE motion dataset. IDLE movements are subtle and unconscious, making them difficult to capture accurately with the typical motion capture setups. These setups can interfere with the naturalness of these motions due to the presence of equipment or the awareness of being recorded. Additionally, recordings are typically performed by professional actors, which can result in exaggerated or acted behaviors rather than capturing genuine, spontaneous IDLE movements.

[1] https://www.rockstargames.com/games/IV

This thesis aims to address this gap by providing both dataset and motion capture recording methodology. This dataset will provide a valuable resource for further research and application in generating realistic animations through machine learning techniques and the methodology will enable anyone to record their own MoCap data with the topic of their choice, without the need for expensive equipment or professional actors.

1.1 Objectives

The primary aim of this thesis is to fill an existing void in the literature by generating a dataset containing non-acted and acted 3D IDLE motions, minimizing the effect of the setup in the subjects recorded. In order to generate it, a review from the literature of motion capture will be needed and, based on it, a motion capture methodology will be developed, which will be designed to be easily replicable, cost-effective, efficient and as non-intrusive as possible given the nature of the data to be collected. The dataset will be analyzed to provide a detailed insight into the characteristics of the raw IDLE motions and will be made publicly available to the research community or anyone interested in using it.

Therefore, the objectives of the thesis can be summarized as follows:

- **Develop a motion capture methodology:** Develop from scratch a motion capture methodology that is cost-effective, easily reproducible, and, given the nature of the dataset to be recorded, non-intrusive. Mocap setups are known to be often expensive, complex, and conditioning with the intended recordings if not performed by professional actors. The methodology should be designed to be as simple as possible so anyone can replicate it and create their own MoCap data. The fundamental intention for developing this methodology is to contribute to the literature by providing a way to produce high-quality MoCap recordings without the need for expensive equipment.
- **Generate and analyze the 3D IDLE motion dataset:** Generate the first-ever non-acted and acted dataset of 3D IDLE movements using the previously developed motion capture recording methodology. This dataset will include a variety of common motions found in video games and animation movies to ensure its versatility. Conduct a detailed analysis of the generated dataset, offering initial insights and an examination of the non-acted IDLE motions. Provide all tools and scripts used in the analysis to the public, enabling replication with other recorded data.
- **Publish the work:** Make the dataset publicly available to the research community and anyone interested in utilizing it. Publish it in a standard format compatible with any 3D software, accompanied by post-processing scripts. Submit a paper detailing the dataset and the methodology used to a conference or journal, ensuring that the findings and resources reach a broader audience.

CHAPTER 2

Background

In this chapter, the existing literature on motion capture systems and techniques is reviewed. The chapter also discusses the publicly available related 3D datasets. Finally, the existing gap in the literature is identified, and the need for a dataset containing non-acted and unconditioned 3D IDLE motions is highlighted.

2.1 Motion capture techniques

Motion capture (MoCap) techniques have evolved significantly over the years. The primary goal using this systems is to capture the intricate details of human motion with high accuracy, using cameras and specialized software that performs pose estimation to generate realistic animations and simulations. These systems can be classified based on the technology they employ and the nature of their setup. The main categories, marker-based and markerless systems, will be analyzed in more depth below.

2.1.1 Marker-based systems

Marker-based MoCap captures human movement by attaching reflective markers or sensors to key body points. These markers are then tracked by multiple cameras, or the sensors directly measure movement, allowing for reconstruction of the movement in 3D space.

This MoCap-type system is known for its high accuracy and is often used in applications where precise movement data is required, such as realistic animations, biomechanics, and virtual reality. However, despite its high accuracy, marker-based MoCap does have limitations. Applying markers can be time-consuming, and more importantly, subjects have to wear a suit equipped with markers or attached sensors, making it invasive.

Generally, motion capture systems can be further classified into three main types: optical systems, magnetic systems, inertial systems

2.1.1.1 Optical systems

Optical systems[4] are one of the most widely used motion capture systems. These systems use cameras to capture the movements of the subject, which are then processed to create a 3D animation. The cameras are placed around the subject to capture the movements from different angles, providing a comprehensive view of the motion. Optical systems can be further classified into passive and active systems. See Figure 2.1 for an example of a optical marker-based system.

Passive optical systems use markers placed on the subject to track their movements. The markers reflect light emitted by the cameras, allowing the system to capture the position and orientation of the markers in 3D space. These systems are highly accurate and provide detailed motion capture data, making them ideal for capturing complex movements.

Active optical systems, on the other hand, use markers with built-in light sources to track the movements of the subject. The markers emit light that is captured by the cameras, allowing the system to track the position and orientation of the markers. Active optical systems are more robust than passive systems and can capture movements in challenging lighting conditions.

* https://www.vicon.com/

Figure 2.1: Marker-based optical Vicon * motion capture system.

2.1.1.2 Magnetic systems

Magnetic systems, as described in [5], use magnetic sensors attached to the subject to track their movements. The sensors emit a magnetic field that is captured by a receiver, allowing the system to track the position and orientation of the sensors in 3D space. Magnetic systems are portable and easy to set up, making them ideal for capturing movements in outdoor environments or confined spaces.

These systems are less affected by occlusion compared to optical systems and can be used in a variety of environments. However, they are susceptible to magnetic interference from metal objects and electronic devices, which can affect precision. See Figure 2.2 for an example of a magnetic system.

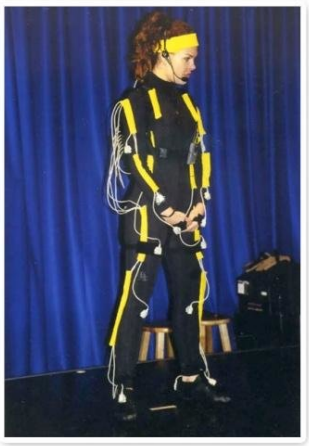


Figure 2.2: Magnetic motion capture system.

2.1.1.3 Inertial systems

Inertial systems [6] use devices called IMU (Inertial Measurement Unit). IMU is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the orientation of the body, using a combination of accelerometers, gyroscopes, and sometimes magnetometers.

Inertial systems are portable and easy to set up, making them ideal for capturing movements in outdoor environments or confined spaces. These systems are less affected by occlusion compared to optical systems and can be used in a variety of environments. However, they are susceptible to drift and noise, which can affect the precision of the captured data. See Figure 2.3 for an example of an inertial system.

Figure 2.3: Inertial motion capture system.

2.1.2 Markerless systems

As alternative to marker-based systems, markerless MoCap systems have gained significant traction in recent years due to their ease of use and non-invasive nature. Unlike marker-based systems that require attaching reflective markers to the body, markerless systems rely on computer vision algorithms that are evolving thanks to deep learning based pose estimation solutions. These algorithms can track body movements from video, eliminating the time-consuming setup process and potential discomfort associated with markers. This makes markerless motion capture well-suited for real-time applications and casual motion capture setups.

Two popular open-source markerless motion capture solutions are MediaPipe [7] and OpenPose [8]. MediaPipe, developed by Google, offers a pipeline for pose estimation, hand tracking, and face detection, providing a comprehensive framework for various motion capture tasks. OpenPose, on the other hand, focuses specifically on body pose estimation, offering real-time performance and support for multiple body instances in a scene. See Figure 2.4 for an example of pose detection using OpenPose and MediaPipe.

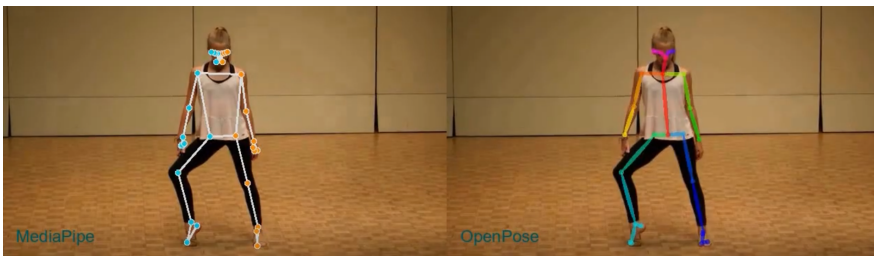


Figure 2.4: Left: Pose detection using MediaPipe. Right: Pose detection using OpenPose.

Though, they are less accurate than marker-based systems and are susceptible to occlusion and noise. Markerless systems are in the early stages of development and are not as widely used as marker-based systems. However, they have the potential to revolutionize the field of motion capture by providing a more natural and non-intrusive way of capturing movements.

Table 2.1 provides a comparison of the different motion capture systems based on their characteristics.

Motion capture system	Differences	Principal Advantage	Disadvantage
Marker-based Systems			
Optical	Use cameras and markers	High accuracy	Affected by occlusion
Magnetic	Use magnetic sensors	Portable	Susceptible to interference
Inertial	Use accelerometers, gyroscopes	Portable, less affected by occlusion	Susceptible to drift, noise
Markerless System			
Camera-based	No markers required	Non-intrusive	Less accurate, susceptible to occlusion

Table 2.1: Comparison of marker-based and markerless motion capture systems

2.2 MoCap-Based 3D motion datasets

Numerous datasets have been recorded and published in the field of 3D character animation. One of the most widely used datasets is the CMU Graphics Lab Motion Capture Database [9], which contains a large number of motion capture sequences covering a variety of actions and movements such as playing sports, runnnin, walking, etc. Another significant dataset is Human3.6M [10], which is one of the largest and most comprehensive motion capture datasets available. AMASS [11] is another relevant dataset that provides a large-scale dataset of human motion capture data, including 3D body scans and detailed annotations. Another worth mentioning dataset is the *TalkingWithHands16.2M* [12] dataset, which is a large-scale dataset of hand gestures and body movements recorded in a controlled environment, it is composed of conversations between two people. The dataset is composed of the animations and the audio of the conversations and has been used in the GENEA challenge [13] which is a challenge that aims to generate realistic human motion from speech.

All of them use advanced motion capture systems and reconstruction techniques to capture the movements of the subjects. Though they are a really valuable resource and contribution for researchers and developers working in the field of 3D character, they do not have a particular focus on the animations of type IDLE. IDLE gestures are subtle, unconscious movements that are specially challenging to capture due to the fact that they are not planned or acted. They go hand in hand with naturalness, recording them in controlled environments and with professional equipment makes them propense to lacking the pursued naturality.

To the best of our knowledge, there are currently no publicly available datasets specifically focused on 3D IDLE animations. While prior research has documented the creation of 3D IDLE animation datasets for machine learning applications [2, 14], these datasets remain unreleased to the public. In [2], Thalmann et al recorded a dataset by placing markers on the body of the subjects and recording their movements. It was later used in a Principal Component Analysis (PCA) to generate new animations. In [14], the recording of the dataset focused on facial expressions and was created by recording the faces of subjects. It had the intention to be a comprehensive collection of facial movements, useful for creating realistic facial animations in 3D characters which had the intention to be applied to avatars or virtual

assistants.

However, in the literature, only a single 2D IDLE gesture dataset has been identified, named IdlePose [3]. This dataset was created by subtly hiding the true intent of the recording from the subjects to make them wait, thus capturing their natural, uninhibited behavior. Ethical considerations were properly addressed, as the subjects were informed about the true nature of the recording after the session. The objective of this strategy was to capture the most natural and realistic IDLE animations possible. Also, the system used to capture it was markerless, which allowed to not condition the subjects with the recording process. Since the dataset is 2D, it has the limitation of not been directly applicable to 3D characters.

Recognizing the need for a comprehensive 3D IDLE motion capture dataset, this thesis aims to fill this gap in the literature by generating one. By using insights from the 2D approach, the goal is to ensure the captured movements are natural and authentic, thus addressing a gap in the current research.

CHAPTER 3

Planification

3.1 Tasks Definition

In this section, the main tasks required to achieve the project's objectives are outlined as defined in Chapter 1. These tasks were essential to ensure progress and successful completion of the project.

3.1.1 Planification

The planning phase involved setting several milestones. This included identifying key tasks, setting timelines, defining the scope of work, and allocating resources effectively to ensure the project stayed on track.

3.1.2 Development

The development phase focused on the actual execution of the planned tasks. This included setting up the motion capture system, recording the 3D IDLE dataset, and performing the data analysis. Regular progress reviews were conducted with the directors of the project to ensure alignment with the project's objectives.

3.1.3 Documentation

Documentation was a continuous process throughout the project since it was going to serve as a support afterwards in the form of a repository. It involved maintaining records of the methodologies, data collection processes, and analyses performed. This also included writing the thesis and compiling results to ensure comprehensive documentation of the project's progress and findings.

3.2 Project duration and timeline

This section details the project's timeline and compares the estimated times with the actual times taken to complete each phase. A Gantt chart was used to visually represent this timeline. The time distribution for each phase was as follows:

- 50% - 70% of the time was allocated to the recording setup and methodology.
- 20% - 30% of the time was allocated to the dataset generation.
- 30% - 40% of the time was allocated to the dataset analysis.

3.2.1 Gantt Chart

The Gantt chart in Figure 3.1 illustrates the project's timeline, showing the start and end dates of each task and the overall project duration.

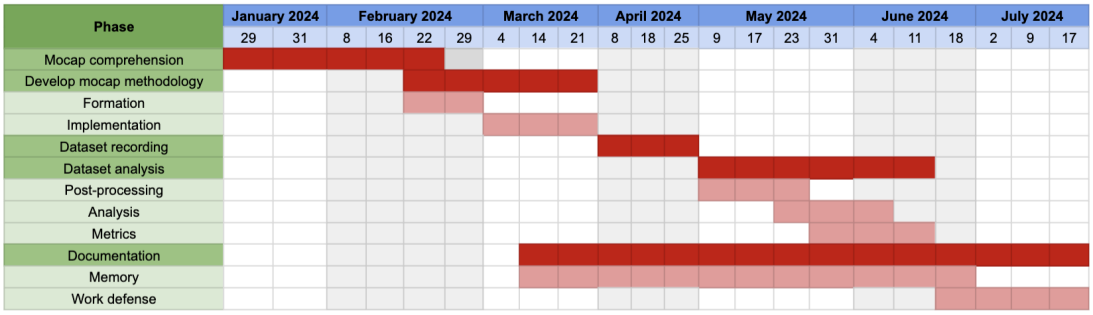


Figure 3.1: Project timeline in a Gantt chart format.

3.2.2 Estimated and Real Time

The Table 3.2 compares the estimated times with the actual times taken for major project tasks. This comparison highlights any deviations from the initial plan.

3.3 Risks

Identifying and mitigating risks was a crucial aspect of the project planning process. This section outlines the key risks identified and the strategies implemented to address them.

3.3.1 Key Risks

Key risks included potential delays in the setup of the motion capture system, unforeseen technical issues, and data integrity or corruption problems. These risks were identified early in the project to develop appropriate mitigation strategies.

Phase	Actual dedication (hours)	Estimated initial dedication (hours)
Mocap comprehension	50	50
Develop mocap methodology	120	100
Formation	55	40
Implementation	65	60
Dataset recording	10	20
Dataset analysis	60	70
Post-processing	25	30
Analysis	25	30
Metrics	10	10
Documentation	80	60
Memory	60	40
Work defense	20	20
Total	320	300

Figure 3.2: Project hours estimated for each phase.

3.3.2 Mitigation Strategies

To minimize the impact of identified risks, several mitigation strategies were put in place. These included developing contingency plans, performing regular checks on the setup, and ensuring data backups. Additionally, an alternative approach was devised in case the primary methodology failed. This involved attempting to extract IDLE animations from other datasets not specifically directed at IDLE and evaluating ML models to set a baseline.

CHAPTER 4

Motion capture methodology

In this chapter, the methodology designed to record motion capture data is explained. The proposed recording methodology aims to grant reproducibility to anyone seeking record high-quality motion capture data, even with limited resources.

Additionally, the chapter details the setup and technologies employed, including the specific hardware and software used, and the configuration of the recording environment. This overview aims to provide a clear and accessible guide for replicating the recording methodology.

4.1 Objectives

Recording motion capture data typically requires complex setups and specialized equipment, including multiple cameras, mocap suits, sensors, and adequate space, which can be impractical for many developers and researchers. The objective when designing this motion capture methodology was to create a way to record high-quality motion capture data while considering the following constraints:

- **Low-resource:** The methodology should be as low-resource as possible, meaning that it should require the least amount of equipment and resources while allowing to record high-quality data.
- **Easily reproducible:** The methodology should be easily reproducible, meaning that it should be simple and easy to follow.
- **Non-intrusive:** The methodology should be as non-intrusive as possible, meaning that the setup should not interfere with the subjects' movements and should not require the subjects to wear any special equipment. When recording certain types of mocap data such as IDLE, it is expected that the subjects to be recorded may not be professional

actors and in order to keep the naturalness of the movements, minimizing the possible interference of the setup on the subjects' movements is crucial.

- **Record extended periods of time:** The methodology was designed to enable the capture of motion data over extended periods, ensuring that any type of motion can be recorded, being the only limiting factor the storage capacity.

4.2 Setup

The setup for the recording methodology is divided into several parts. Each of the following sections will explain in detail the necessary steps to replicate the methodology, always according to the previosly defined objectives.

4.2.1 Hardware

In order to keep it low-resource and easily reproducible, the methodology was designed to accept almost any type of hardware that could be available to the user.

4.2.1.1 Cameras

Any camera that can record video can be used, including commonly available and affordable webcams or smartphone cameras. The only restrictions and requirements the selected cameras should meet are the following:

- **Number of cameras:** At least 2 cameras are required to record 3D motion capture data but 3 or more are recommended to reduce occlusions and improve precision.
- **Resolution:** All cameras should be able to record at the same resolution. Minimal resolution should be 720p.
- **Frame rate:** All the cameras should be able to record at the same frame rate. Minimal frame rate should be 30 frames per second.

In our case, the specific cameras employed were four Logitech C920 webcams. All these cameras are capable of recording at 1080p resolution and 30 frames per second which fulfilled the requirements. The cameras were placed in a way that they could record the subject from different angles in order to be able to capture the subject's movements while minimizing the occlusions. See Figure 4.1 for the camera placement.

As can be appreciated, a recording area measuring 80x90 cm was defined to limit the space in which the subject could move, given the type of dataset to be recorded. This area is sufficiently large for the subject to move freely while remaining constrained to performing IDLE motions. Although there is no strict restriction on the size of the recording area, it is important to ensure that at least two cameras can capture the subject continuously for the 3D reconstruction to be possible.

4.2.1.2 Computer

A computer is needed to manage cameras that require a connection, such as webcams. If the cameras used are smartphone cameras or cameras that posess internal storage and do not require a computer to function, only one computer will be needed later for synchronizing the videos and reconstructing the 3D animations. Otherwise, any computer that can connect to the cameras and store the video data can be used, provided it has sufficient storage capacity.

The approach we used was having one computer per camera. This was done in order to keep it low-resource as any computer with a USB port could be used instead of having to use a more powerful computer with multiple USB ports. In our case, the computers were mixed between Windows and Linux operating systems. See Figure 4.1 for the complete hardware setup.

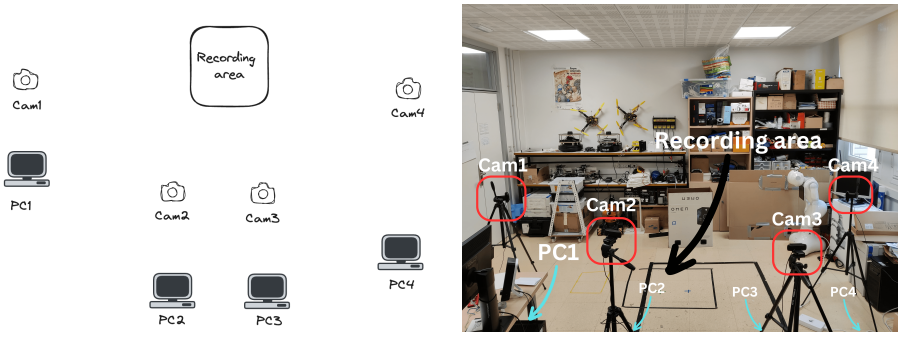


Figure 4.1: Left: scheme of the setup. Right: real image of the setup

4.2.2 Software

The software used for this methodology was chosen to be open-source and free in order to keep it easily reproducible and affordable. This section will provide a step-by-step explanation of the process, introducing the relevant software at each stage.

4.2.2.1 Recording

The very first step to take is to record videos using the cameras. For this purpose, the software used was **OBS Studio** [1]. OBS Studio is a free and open-source software for video recording and live streaming. It is available for the most common operating systems such as Windows, Linux, and macOS.

A great feature of OBS Studio is its integrated websocket server, which allows for remote control of the software and summed with a powerful Python API allows for the automation of the recording process. This served a great purpose for the methodology.

To create a more efficient pipeline for recording and saving time, a script using the OBS Python API was developed. Initially, the start and stop of recording on each camera had to

[1] https://obsproject.com/

be done manually. To address this limitation and ensure efficient data management within a Local Area Network (LAN), the script connected to the OBS Studio websocket server on each computer, enabling all cameras to start recording simultaneously with a single key press. Once the recording session was completed, another key press stopped all cameras from recording and retrieved the videos from each computer to the central computer via Secure Shell (SSH). This automation significantly improved the recording process, eliminating the need for manual operation to start, stop, and retrieve videos. The script can be found in the repository [2] accompanying this thesis.

4.2.2.2 Video synchronization

Though the script allowed for the cameras to start recording at the same time, due to network latency and different hardware, the videos were not perfectly synchronized. To achieve this, a video displayed on a screen was recorded by all the cameras at the beginning or end of the session. This video served as a reference for synchronizing the recordings, if required.

The video used lasted few seconds and consisted of a red screen that changed to green. For a better understanding see Figure 4.2. For reproducibility purposes, the video can also be found on the repository.

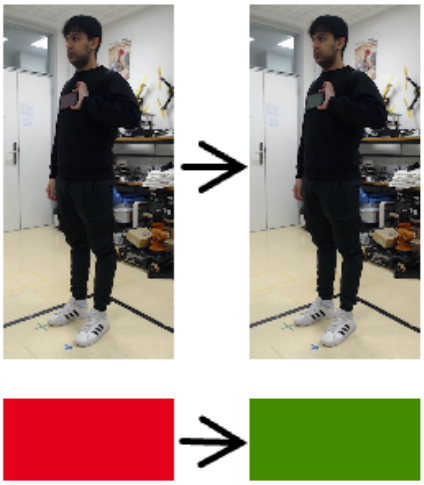


Figure 4.2: Synchronization video shown on the phone screen and recorded by all the cameras.

An approach to automatically synchronize the videos was tried using computer vision tools. It consisted of detecting the frame where the color change happened in the smartphone screen. However, the results were not as expected due to noise in the videos and variable lighting conditions. the manual synchronization was adopted as the best approach. This attempt can be found described in Appendix 7.2.3.

[2] https://github.com/AAAbduu/Thesis-MoCap-4All

To address the manual synchronization of the videos, the tool used was **Kdenlive** [3]. Kdenlive is a free and open-source video editing sofware available for the most common operating systems. The software was used to synchronize the videos by aligning the frame when the synchronization video changed from red to green.

Given the fact of managing multiple cameras, one video per camera will be needed to be processed. After synchronizing the videos, they needed to be rendered in order to save them. Kdenlive was the perfect choice since it allows to enqueue render jobs and render them in the background while the user can, for example, keep working with other videos.

4.2.2.3 3D reconstruction

Once the videos are synchronized and saved, the next step is to reconstruct the 3D motion capture data. For this, the software used was **FreeMoCap** [15]. FreeMoCap is a free, open-source, research-grade markerless motion capture software which was developed to be accesible.

Written in Python, FreeMoCap uses internally libraries such as OpenCV [16], NumPy [17], and MediaPipe [7] to facilitate its operation. Whether capturing data in 2 dimensions with a single camera or in 3 dimensions with multiple cameras, FreeMoCap is capable of providing high-quality results while remaining accessible to users with little to no experience. See Figure 4.3 for a first insight.

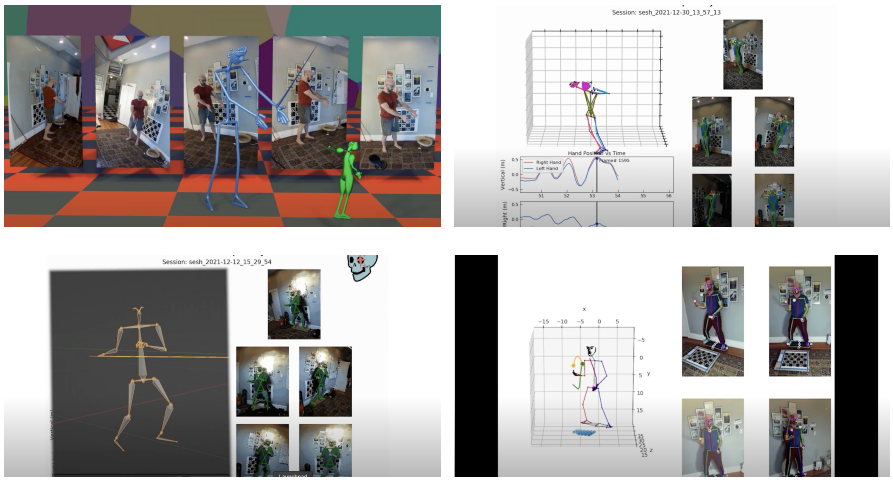


Figure 4.3: A first insight of FreeMoCap

The pipeline used by FreeMoCap starts by calibrating all cameras (see Appendix 7.2.3 for a deep insight of the calibration process) running bodies and landmarks detection using MediaPipe. From this library, FreeMoCap uses the *Pose Landmark Detection* solution which uses a model to detect the presence of bodies similar to [18] and then alongside another model based on Convolutional Neural Network architecture [19] to add a complete mapping of the pose. This data is stored for each frame and for each of the recordings to be later used.

[3] https://kdenlive.org/en/

FreeMoCap runs a triangulation algorithm to calculate the 3D position of the landmarks in the space. Triangulation essentially consists in the process of determining a point in 3D space given its projections onto two or more images. There are multiple ways to solve this problem, FreeMoCap uses Anipose [20] which is a toolkit for robust and efficient 3D markerless motion capture.

Although the software is still under development, it is capable of reconstructing 3D motion capture data from synchronized videos, obtaining high-quality results. It also allows for recording the videos already synchronized and reconstructing the 3D data with them, but currently this method has a limitation regarding the length of the videos that can be recorded since the software stores the videos in RAM memory. We tried in a computer with 32 GB RAM, given our camera setup described in 4.2.1.1, we were able to get around 2:30 minutes of recording. This limitation is not present when the videos are loaded from the disk. That is the reason why the methodology was designed to record the videos with OBS Studio, synchronize them with Kdenlive and then load them into FreeMoCap for the 3D reconstruction.

4.2.2.4 Data export

FreeMoCap has built-in integration with **Blender**[4] for further animation refinement. After reconstructing the 3D motion capture data, FreeMoCap can directly export it into a Blender scene (as seen in Figure 4.4). So far, the only 3D software that is integrated with FreeMoCap is Blender. Blender, a free and open-source 3D creation suite, offers a powerful toolkit for modelling, animation, simulation, rendering, and more. It supports importing motion capture data, allowing to clean, process, and enhance the captured movements.

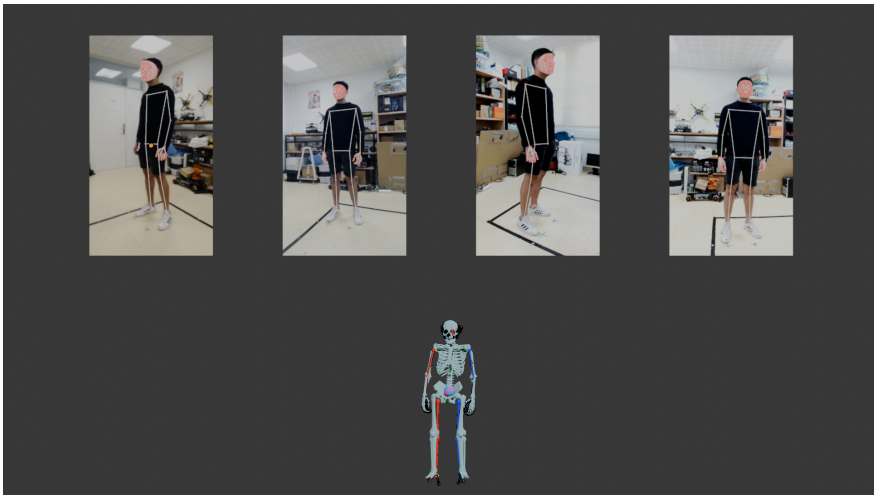


Figure 4.4: Exported motion capture data in Blender

In order to export the motion capture data from Blender, the Biovision Hierarchy (BVH) format was used as it is mostly the standard format for motion capture data, is human-readable,

[4]https://www.blender.org/

and can be easily imported into other software. BVH is a file format that stores hierarchical skeletal animation data. It was developed by Biovision, a motion capture services company, later owned by Motion Analysis Corporation [5], and is now widely used in the motion capture industry.

BVH files encapsulate the components required for animation reconstruction. Within them, the hierarchical structure of joints that compose a skeleton is defined, establishing parent-child relationships. For each frame of animation, BVH records the position and rotation of each bone relative to its parent or the root of the skeleton.

Rotation in BVH files is represented using Euler angles which describe rotations around the local coordinate system's X, Y, and Z axes. In order to translate the joints, at the beginning of the file, the offset of each bone relative to its parent or the root of the skeleton is specified. Later, the translation data is stored for each frame and is applied to the offset to obtain the final position of the joint.

BVH's versatility accommodates various skeletal configurations, from simple biped structures to complex, custom-designed rigs for specialized motion capture applications.

See Figure 4.5 for a first insight of the file and subsequent import into Blender.

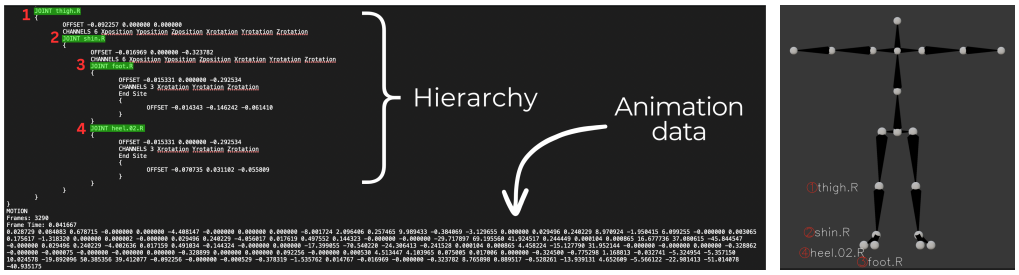


Figure 4.5: On the left, a BVH file opened in a text editor. On the right, the import of the BVH file into Blender.

4.3 Summary

The motion capture recording setup was defined with three primary objectives: to be low-resource, easily reproducible, and non-intrusive. The setup included multiple cameras strategically placed to capture the subject from various angles, each connected to a dedicated computer, and a defined recording area. Open-source and free software was utilized throughout the process, including OBS Studio for recording, Kdenlive for video synchronization, FreeMoCap for 3D reconstruction, and Blender for data export.

This methodology was designed to facilitate extended recording sessions, with storage capacity being the only limiting factor. The aim was to provide a valuable contribution to the

[5] https://www.motionanalysis.com/

field of motion capture, enabling researchers and practitioners to easily capture high-quality mocap data for animation and research purposes.

Supporting files and detailed instructions for this methodology are available in the repository [6] associated with this thesis, and are free to use and modify.

[6] https://github.com/AAAbduu/Thesis-MoCap-4All

CHAPTER 5

Dataset generation

In this chapter, the methodology used to generate the 3D IDLE animation dataset is presented. The chapter begins by outlining the objectives of the dataset and the strategies employed to achieve them. The methodology section then details the guidelines for capturing IDLE behavior, gathering subjects, and the recording process. The results and conclusions of the generated dataset are discussed and presented.

5.1 Objectives

Given the lack of existing 3D IDLE animation datasets and the limitations encountered with previous approaches, the primary objective was to contribute to the field by creating the first dataset of pure 3D IDLE animations.

More precisely, the aim is to define and document the process of creating a comprehensive dataset of pure 3D IDLE animations. This involves capturing authentic, naturalistic idle behaviors of subjects using techniques to discreetly capture IDLE animations, subtle prompts and tricks to ensure subjects remain in an idle state, thereby capturing genuine IDLE animations without their awareness. To achieve this, the recording setup described in Chapter 4.

This dataset aims to support various applications in animation, virtual reality, gaming, and research, offering a valuable resource for developing more lifelike and responsive virtual characters. The dataset will be an initial proposal for future research but it should be extended and improved in the future.

5.2 Methodology

The methodology aimed to capture high-quality IDLE animations with a focus on natural and effortless gestures. Given the lack of existing datasets and the constraints in resources, a well-structured plan was essential. This section outlines the devised plan and the strategies implemented to achieve the objectives efficiently.

5.2.1 Guidelines for Capturing IDLE Behavior

A plan was devised to capture as much data as possible with a specific focus on IDLE animations. The first step involved defining the gestures to be included in the dataset. It was necessary to ensure that these gestures were simple, allowing them to be executed effortlessly by subjects who were not professional actors.

The intended gestures included common idle animations found in video games or animated movies, such as looking around, looking behind, or checking a watch on the wrist. To add variety, the gestures were repeated several times. With the widespread use of smartphones, new idle gestures have emerged, such as looking at a phone while doing nothing else. This modern behavior was relevant and thus included in the dataset. To make the dataset as complete and varied as possible, it did not just focus on pure do-nothing gestures. The opportunity to record different gestures was taken advantage of by including actions that could be used in various situations, enhancing the dataset's completeness and versatility.

Maximizing the naturalness of these gestures was crucial. Internal recording tests revealed that prolonged sessions could lead to tedium and unnatural behavior, as IDLE behavior primarily involves "not doing any particular action". Moreover, the presence of equipment, such as cameras and computers, was found to have the potential to alter participants behavior. To address this, a strategy was developed to sequence actions in a manner that flowed smoothly and dynamically, with each session lasting no longer than 10 minutes.

To further enhance authenticity, participants were instructed to wait for a signal before beginning their actions, under the pretext that synchronization was in progress, while recording had already commenced. This approach allowed for the capture of natural behavior, as participants were unaware that recording had already started. This approach, similar to the one presented in [3], helped capture natural behavior since participants did not know they were being recorded right away. See Figure 5.1 for the defined gestures to be captured and the guidelines used for the recording sessions.

Though this approach raised some ethical considerations, participants were informed about the recording process once the session was over and given the option to withdraw their data if they wished. This ensured that participants were comfortable with the recording process and that their privacy was respected.

5.2.2 Gathering subjects

To recruit participants, a Doodle poll was created, allowing individuals to sign up for the recording sessions conveniently. Additionally, invitations were extended to alumni from the

Time (s)	Action	Directions	Done	Notes
120	Record without the subject knowing (natural IDLE)	"While I calibrate the recording system, wait within the marked area, moving barely"		
120	Acted IDLE	"Stand there and do nothing, without putting your hands in your pockets, until I tell you"		
5	Look at the sky (x1)	"Now, look at the sky until I say so"		
5	Look at the sky (x2)	"Again"		
5	Look at the sky (x3)	"Last time"		
5	Look at the surroundings (x1)	"Look at your surroundings"		
5	Look at the surroundings (x2)	"Again"		
5	Look at the surroundings (x3)	"Last time"		
5	Look at the floor (x1)	"Look at the floor"		
5	Look at the floor (x2)	"Again"		
5	Look at the floor (x3)	"Look at your feet (move them)"		
5	Look behind (left)(x1)	"Look behind from your left"		
5	Look behind (left)(x2)	"Again"		
5	Look behind (right)(x1)	"Look behind from your right"		
5	Look behind (right)(x2)	"Again"		
5	Phone interaction (x1)	"Pull out your phone, check what hour it is and put it back in your pocket"		
5	Phone interaction (x2)	"Again"		
5	Phone interaction (x3)	"Again"		
5	Look at the wrist watch (x1)	"Look at your wrist watch"		
5	Look at the wrist watch (x2)	"Again"		
5	Look at the wrist watch (x3)	"Again"		
180	Use phone	"Pull out your phone and use it as you would, you can do whatever you want, do so until I tell you"		

Figure 5.1: Guideline used for the recording sessions.

computer science degree program, master's students, doctoral researchers, and members from the faculty such as teachers to participate in the recordings. All of them from the Computer Science Faculty of the University of the Basque Country in San Sebastián/Donostia. Of course, beforehand, subjects were not informed of the nature of the recordings, as this could potentially alter their behavior. Instead, they were told that the recordings were for a research project and that they would be required to perform a series of actions to record Moap data.

5.2.3 Recording process

A date and an hour were set for each participant to come to the recording location. The recording setup was prepared in advance and the calibration of the cameras was done.

Before the subject entered the recording room, the cameras had already started recording. The subject then was placed within the recording area and instructed to wait for our signal and instructions to start the recording. By tricking the subject into thinking that the recording was not yet started and some synchronization was going on, allowed to capture the most natural and uninhibited behavior possible.

During the recording, the guideline was followed, and the subject was instructed to perform the actions with a small hint or prompt if needed. Also, notes were taken on the subject's behavior and any issues that arose during the recording.

Once the recording session was finished, the subject was informed about the true nature of the recording and how they were tricked into thinking the recording was not started. They were also given the option to withdraw their recording if they wished which they did not. Participants were asked about their experience, if they felt comfortable and not inhibited

during the recording, and if they had any comments or suggestions for improvement.

5.3 Results and conclusions

A total of 17 subjects, comprising 15 men and 2 women with an average age of 30 years old, participated in the recording sessions, each of whom followed the guidelines and performed the actions as instructed. In order to maintain consistency and structure in the dataset, a naming convention was established for the videos and folder hierarchy. This would be helpful for accessing the desired data and for scripting purposes. To protect the privacy of participants, their names were replaced with numbers, and the final dataset only contained BVH files, deleting the recorded videos of the participants and leaving the naming as: *XXXX_actionPerformed.bvh*.

A dataset comprising 231,506 frames, totaling an estimated duration of 2 hours, is categorized as shown in Table 5.1

Animation Type	**Frames**	**Approximate Duration**
Raw IDLE animations	38,694	21 minutes, 30 seconds
Acted IDLE animations	61,174	33 minutes, 59 seconds
Multiple actions	52,337	29 minutes, 5 seconds
Lookback animations	5,761	3 minutes, 12 seconds
Phone animations	73,540	40 minutes, 51 seconds
Total duration	231,506	2 hours, 8 minutes, 36 seconds

Table 5.1: Dataset composition and duration

The development and recording of the 3D IDLE motion dataset marks a significant step forward in the field of motion capture for animation and video games. The dataset, comprising various IDLE gestures and actions performed by 17 subjects, demonstrates the feasibility of creating high-quality motion capture data using a low-resource and easily reproducible methodology. The established naming conventions and folder hierarchy ensured consistency and ease of access, while measures were taken to cover ethical considerations and protect participant privacy, resulting in a dataset of BVH files ready for public use.

The recorded data holds a wide range of IDLE behaviors, from common animations like looking around and checking a watch to more modern gestures such as interacting with a smartphone. This variety not only adds depth to the dataset but also provides valuable resources for researchers and developers working in animation, virtual reality, and related fields. The dataset's total duration of over two hours offers a robust foundation for further analysis and application.

The dataset is scheduled to be published and made available for public use alongside a paper that will be submitted to a conference.

CHAPTER 6

Dataset analysis

This chapter aims to delve into a comprehensive analysis of the dataset generated in the previous chapter. The primary objective of this analysis is to understand the characteristics and structure, uncover patterns and calculate essential statistics. This analysis serves as a foundational reference for interpreting the data and guiding subsequent research and development activities.

6.1 Data post-processing

The dataset generated following the methodology defined in Section 5.2 comprises of several BVH files, each containing the hierarchical structure of a skeleton and the motion data for each joint across all frames.

After visualizing most of the animations using Blender, it was observed that several joints had no motion or had faulty capture. We believe the main reason lies in the nature of the 3D reconstruction process itself. Which might not be able to capture some joints accurately. To remove those joints, a script using the Blender API was developed to filter out the joints. See Appendix 7.2.3 for a detailed view of the joints that were filtered out. The script is available in the repository and can be easily modified to filter out other joints if needed.

In order to analyze the data effectively, the BVH files were post-processed to extract the time series for each joint and store them in a more structured format that can be easily accessed and analyzed for instance, by ML algorithms. Therefore a script was developed to extract the time series from the BVH files and organize them into structured CSV files. The script was written in Python and is also available in the repository [1] alongside the preprocessed data into CSV files.

[1] https://github.com/AAAbduu/Thesis-MoCap-4All

6.2 Data breakdown

The BVH files across all the dataset posses the same hierarchical relationships, the same number of joints and each joint has the same number of degrees of freedom. After the post-processing of the data, the motion capture files were standardized to include 21 joints, each with either 3 or 6 degrees of freedom (DOF) depending on the joint. See Table 6.1 for all joints and their respective degrees of freedom.

Joint	Pos-X	Pos-Y	Pos-Z	Rot-X	Rot-Y	Rot-Z
pelvis	X	X	X	X	X	X
spine	X	X	X	X	X	X
spine.001				X	X	X
neck	X	X	X	X	X	X
face	X	X	X	X	X	X
shoulder.L	X	X	X	X	X	X
upper_arm.L	X	X	X	X	X	X
forearm.L	X	X	X	X	X	X
shoulder.R	X	X	X	X	X	X
upper_arm.R	X	X	X	X	X	X
forearm.R	X	X	X	X	X	X
pelvis.L	X	X	X	X	X	X
pelvis.R	X	X	X	X	X	X
thigh.L	X	X	X	X	X	X
shin.L	X	X	X	X	X	X
foot.L				X	X	X
heel.02.L				X	X	X
thigh.R	X	X	X	X	X	X
shin.R	X	X	X	X	X	X
foot.R				X	X	X
heel.02.R				X	X	X

Table 6.1: Joints and their respective degrees of freedom

For each existing DOF, a time series can be extracted from the BVH file formed by all the values across all the frames. This time series describes the motion of the joint across the animation. Counting the total number of existing DOF results in 111 time series, which is extended throughout the whole dataset as the hierarchy and number of joints are consistent across all the BVH files.

6.3 Data analysis

With the dataset described and post-processed, the next phase involves delving into detailed analyses to extract valuable insights. Before diving into it, a clarification is needed. Many 3D software programs utilize a right-handed XYZ coordinate system, where X represents the horizontal axis, Y the vertical axis, and Z the depth axis. In this system, the positive Y direction corresponds to UP vector. However, Blender adopts a different left-handed XZY coordinate system, where the positive Z direction defines UP vector. This is important to keep in mind when analyzing the data. See Figure 6.1 for a visual representation of the coordinate systems.

Figure 6.1: On the left: Y-up coordinate system. On the right: Z-up coordinate system.

Given the complexity of the dataset and the challenge of visualizing all the data comprehensively, a structured approach was implemented to enhance visualization and comprehension. For an initial visualization approach, refer to Figure 6.2. The time series depict the animation playback. Each joint's position across all axes is plotted, accompanied by the rotation time series on the right side. Note that the selected joint time series are the most representative and due to symmetry, only the left joint time series are plotted.

6.3.1 Constant time series identification

One of the questions that arise given the dimensions of the dataset is whether all the time series are equally informative. Specifically, with 111 time series per animation, it is important to assess whether all these series contribute equally to the understanding of the motion. An initial inspection of the data plots reveals that some time series are constant across all frames. Although these constant time series might appear redundant, removing them results in the loss of critical information regarding the correct motion of the joints. In the context of BVH files, both rotational and positional data are crucial for accurately representing the movement of each joint. The rotational data defines the orientation of a joint in 3D space, which is essential for capturing the precise angles and directions of movements. Similarly, the positional data specifies the exact location of each joint, ensuring that the spatial relationships between different parts of the body are maintained. Removing these seemingly redundant time series could lead to significant inaccuracies in the animation, such as joints appearing disconnected, limbs moving in an unnatural manner, or the entire skeleton behaving unpredictably.

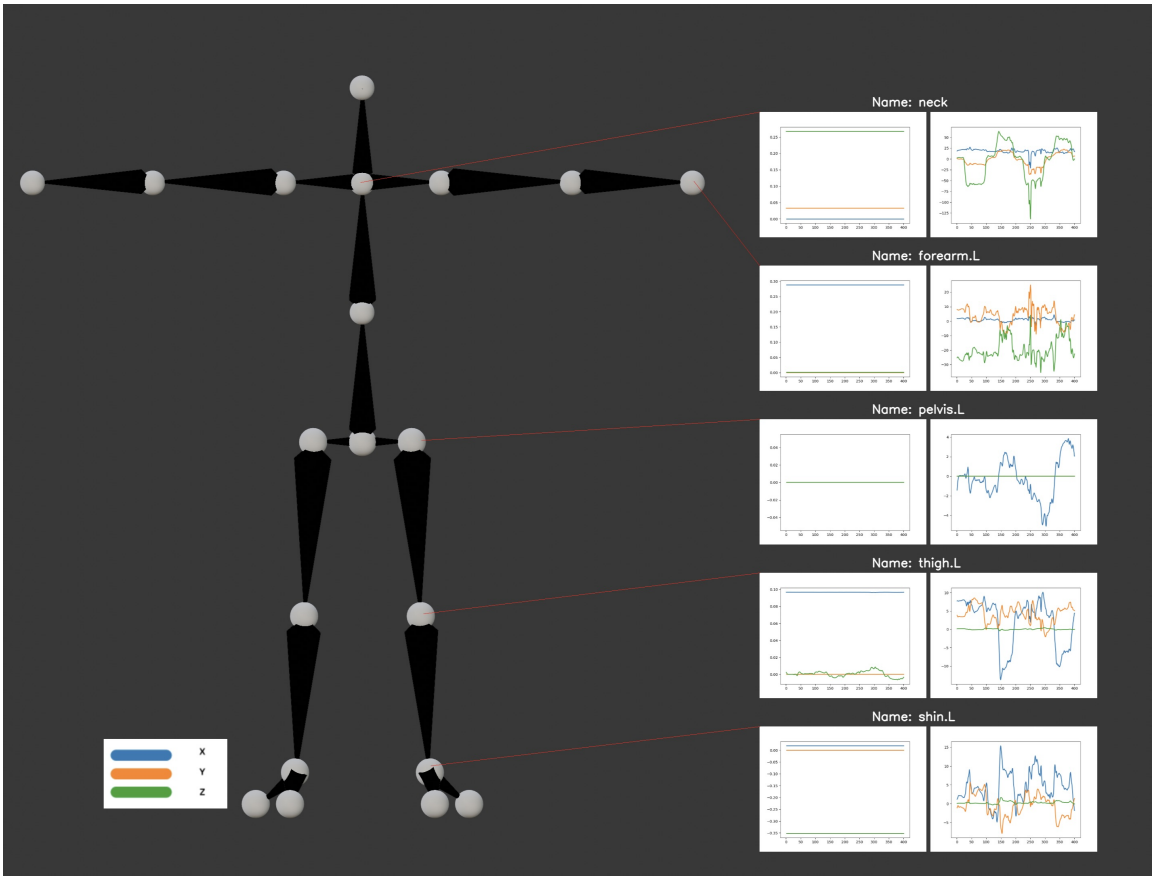


Figure 6.2: A first approach to visualize the data. An skeleton from a BVH file is shown with the time series of the position and rotation of the joints.

Even if certain time series seem to show little to no variation, they play a vital role in preserving the structural integrity

However, can the dimensionality of the dataset be reduced for modeling purposes? To answer this question, the variance of each time series was calculated. The variance of a time series measures how much the values deviate from the mean. By calculating the variance, the aim is to identify the time series that are constant and may not need to be considered, for example, for subsequent forecasting models. The variance was calculated as follows:

$$\mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2 \tag{6.1}$$

Where

- N is the number of frames in the file,

- X_i is the value of the time series at frame i,
- $\bar{X}$ is the mean of the time series.

After calculating the variance of each time series across the entire dataset, those with a variance near zero were identified. A threshold of 10^{-6} was set to consider a time series as constant, based on the precision of the data. Once the constant time series were identified, a search was conducted to find the file with the minimum number of constant series. This file was then used as a reference to check if these series were consistently constant throughout the dataset. The findings are illustrated in the figure 6.3, where the constant joints are highlighted.

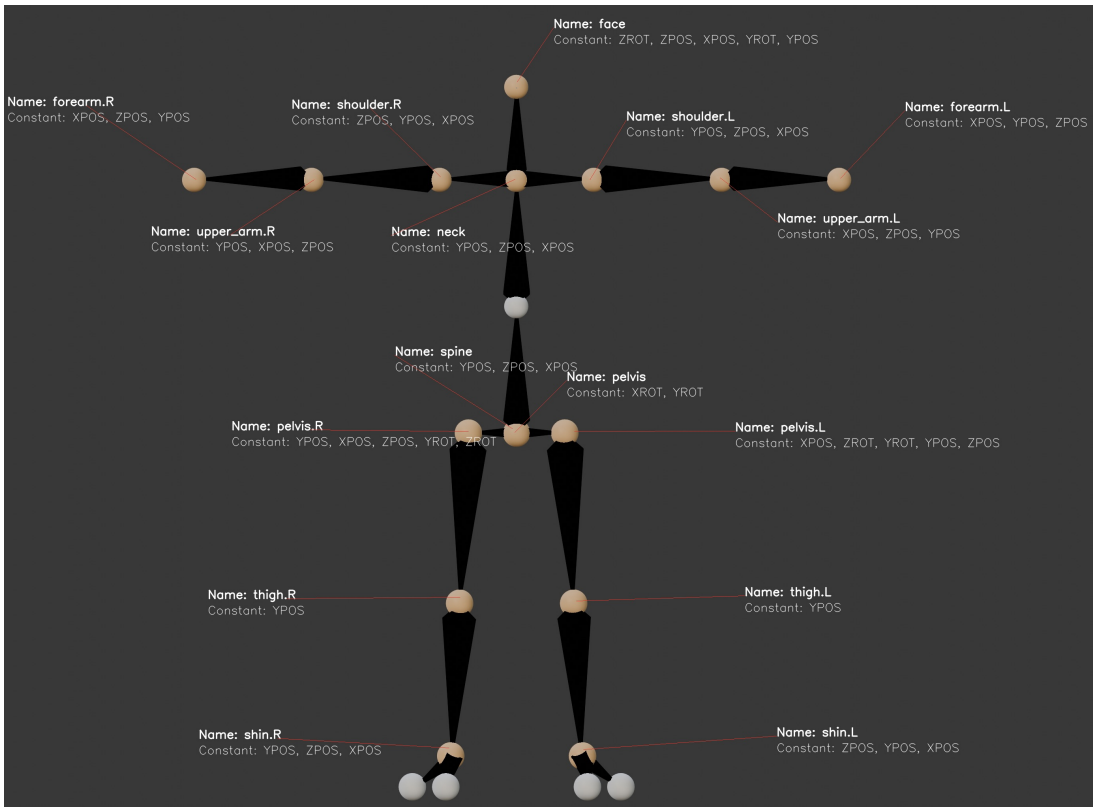


Figure 6.3: Constant time series across the dataset.

A total of 49 constant time series were identified across the dataset. Though these time series are constant, they are essential for the correct motion of the joints, therefore they cannot be removed. Consequently, the modelling might be reduced to 111 - 49 = 62 time series, focusing only on those that exhibit variability and obtaining the constants from the dataset.

6.3.2 Variance analysis of IDLE animations

To further understand the characteristics of the motion data, the variance in the non-acted animations was analyzed, both for rotation and position, and in aggregate form. The IDLE

animations consist of movements where subjects are relatively still, providing a baseline for identifying inherent variability in the data.

The variance of each time series in the IDLE animations was calculated separately for rotational and positional data, as well as combined. This approach helps to highlight which aspects of the motion capture data exhibit the most significant changes and which remain relatively stable.

6.3.2.1 Aggregated motion

The aggregated mean variance of the time series across all the joints was calculated as follows:

$$\text{Joint}_a = \frac{1}{N} \sum_{i=1}^{N} \text{Var}(Rot_{ai}) + \text{Var}(Pos_{ai}) \tag{6.2}$$

Where

- N is the number of files,
- a defines the axis such that $a \in \{X, Y, Z\}$,
- $\text{Var}(Rot_{ai})$ is the variance of the rotational data for axis a in the i-th file,
- $\text{Var}(Pos_{ai})$ is the variance of the positional data for axis a in the i-th file.

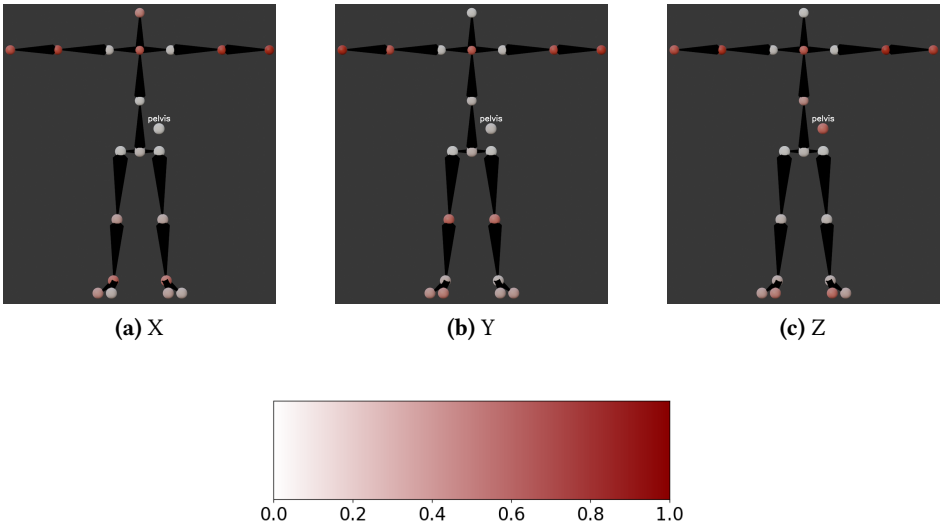
This result answers various questions, which joint has the most variability when speaking of motion, which has the least and in which axis respectively. The results are shown in Figure 6.4.

The Figure 6.4 show the aggregated mean variance of the motion data across all joints for each axis. The colors of the joints in the figures match the colors in the scale, providing a visual representation of the variance distribution. For the sake of clarity, the names of the joints are omitted from the figures, but they can be found in Figure 6.3. Also, the pelvis joint has been translated in order to see it, as otherwise it would be hidden by the spine joint.

The figure reveal that the arms exhibited the most motion variability in all axes, while the pelvis and spine exhibited the least alongside the shoulders. From the X-axis figure, the left leg joints showed less variability than the right leg joints which can indicate subjects mostly leaning on their left leg. Also, the head joint exhibited the most variability in the X-axis, which can be attributed to some tilting when changing equilibrium and looking around. Note that those joints which lack positional degrees of freedom, their variance is completely attributed to the rotational data.

6.3.2.2 Positional motion vs rotational motion

To better understand which aspects of motion provide more informative data, the analysis was conducted by separately evaluating positional and rotational data. This approach aimed



(a) X **(b)** Y **(c)** Z

Figure 6.4: Aggregated mean variance of the motion data across all joints for each axis from left to right (X, Y, Z). The values for the joints have been normalized. The color scale indicates the variance level, where the lightest color represents 0 and the darkest color represents 1.

to identify which degrees of freedom contribute more significant information to the analysis of motion.

Since there are several constant time series and some joints lack positional degrees of freedom, the question to answer is which of the time series contribute more to the IDLE motion.

For calculatig the mean variance of the positional data, the following formula was used:

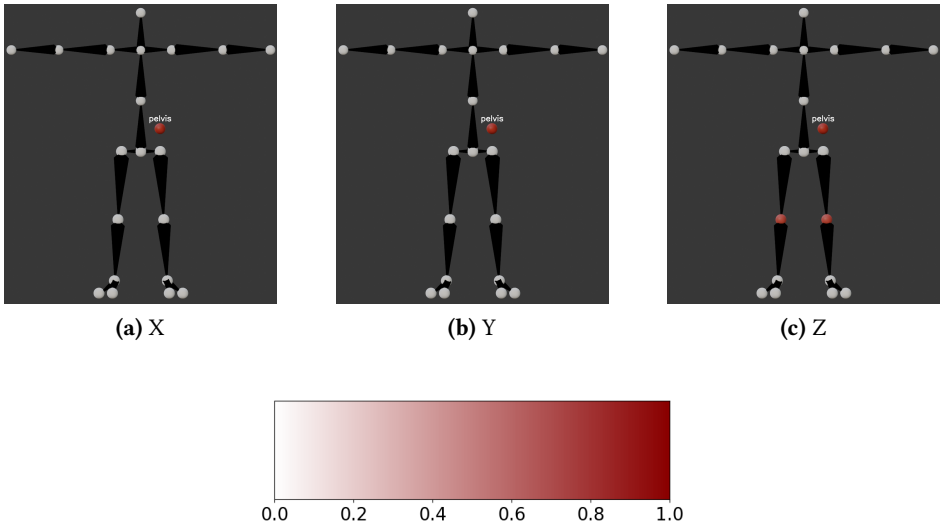
$$\frac{1}{N}\sum_{i=1}^{N}\text{Var}(Pos_{ai}) \tag{6.3}$$

Where

- N is the number of files,
- a defines the axis, that is, $a \in \{X, Y, Z\}$,
- $\text{Var}(Pos_{ai})$ is the variance of the positional data for axis a in the i-th file.

The results are shown in Figure 6.5. Note that for those joints that lack positional degrees of freedom, the variance will be 0.

The Figure 6.5 show the mean variance of the positional data across all joints for each axis. The color scale below indicates the variance level, where the lightest color represents 0 and the darkest color represents 1.



(a) X **(b)** Y **(c)** Z

Figure 6.5: Mean variance of the positional data across all joints for each axis from left to right (X, Y, Z). Note, barely any joint has variance in general. This is due to the fact that, though joints posses positional degrees of freedom, the values mainly remain constant.

The results show that positional data is least variable in general in all axes, the only exception being the pelvis joint in all axes, which has been manually displaced in order to see it, and the thigh joints in the Z-axis. Though it is least variable, positional data is still essential for the correct motion of the joints, as it provides the necessary context for the rotational data. A clear conclusion that can be drawn so far is that rotational data is more variable than positional data and therefore more informative when speaking of motion.

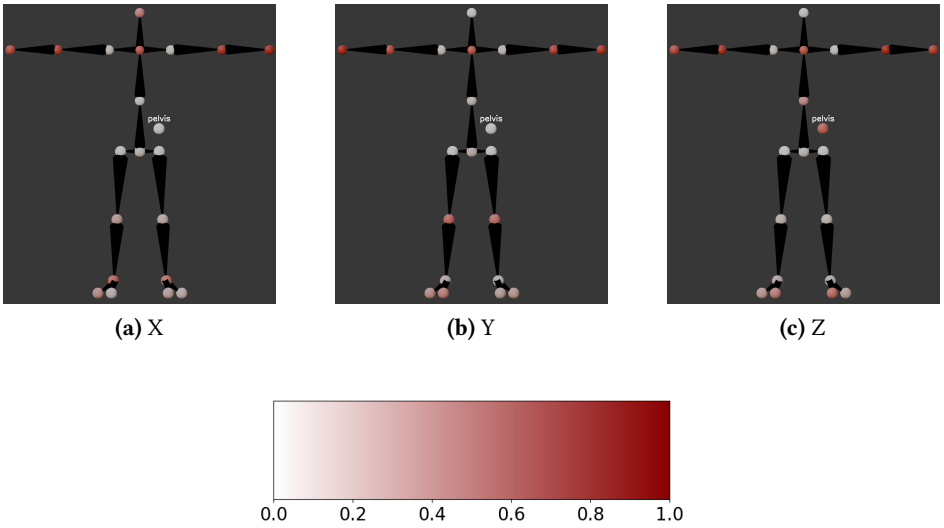
For the rotational data, the mean variance was calculated as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\text{Var}(Rot_{ai}) \tag{6.4}$$

Where

- N is the number of files,
- a defines the axis such that $a \in \{X, Y, Z\}$,
- $\text{Var}(Rot_{ai})$ is the variance of the rotational data for axis a in the i-th file.

In this case, as it can be appreciated in table 6.1, all joints have rotational degrees of freedom, so the variance is calculated for all the joints. As can be seen in Figure 6.6.



(a) X **(b)** Y **(c)** Z

Figure 6.6: Mean variance of the rotational data across all joints for each axis from left to right (X, Y, Z).

The arms and legs exhibit the most variability in axis X and Y, while the spine and pelvis exhibit the least in all of them. The head joint exhibits the most variability in the X-axis which can be attributed to subjects looking up and down whilen looking around.

Moreover, the results show that rotational data is more variable than positional data and more informative when speaking motion-wise, which is consistent with the previous findings. This finding can, for instance, inform to pay closer attention to rotational data when creating realistic IDLE animations for characters or forecasting if speaking of ML.

To support these conclusions, Spearman's rank correlation was applied to check whether these findings applied consistently across all the individual non-acted IDLE animations.

Spearman's correlation is a non-parametric measure of rank correlation that assesses how well the relationship between two variables can be described using a monotonic function. It is particularly useful for identifying and confirming patterns within the dataset. Spearman's correlation requires ranking the data, which consists of assigning a numerical value to each data point according to its relative position in the data series. For instance, the lowest value in the data set would be assigned a rank of 1, the second lowest value would be assigned a rank of 2, and so on.

The objective of this correlation analysis was to determine whether subjects exhibit a consistent mean variance pattern across all joints. By examining this, the aim is to understand if there is a uniform pattern of motion variability among different subjects. The Spearman's rank correlation coefficient is calculated as follows:

$$\rho = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N(N^2 - 1)} \tag{6.5}$$

Where

- N is the number of observations,
- d_i is the difference between the ranks of the two variables for the i-th observation.

The correlation was calculated between the values of each subject that was recorded in the non-acted IDLE animations and the average variance values of the non-acted IDLE data, each containing 111 time series.

With the help of the *spearmanr* and *rank_data* functions from the *scipy* library in Python, the correlation was calculated for each subject. The results of the correlation analysis are shown in Table 6.2.

The results show that the mean correlation across all subjects is 0.983098, with a maximum correlation of 0.992784 found in the file *007_idle.bvh* and a minimum correlation of 0.952282 found in the file *015_idle.bvh*. These results indicate that the mean variance pattern is consistent across all subjects, with a high correlation value. This consistency suggests that the motion variability is uniform across all subjects, which is a critical finding for subsequent modeling and analysis.

A visual representation of the correlation results is shown in Figure 6.7. The figure includes the results for the file with the minimum and maximum correlation, alongside the perfect rank correlation for reference.

Mean correlation	Max correlation	Min correlation
0.983098	0.992784	0.952282

Table 6.2: Spearman's rank correlation results

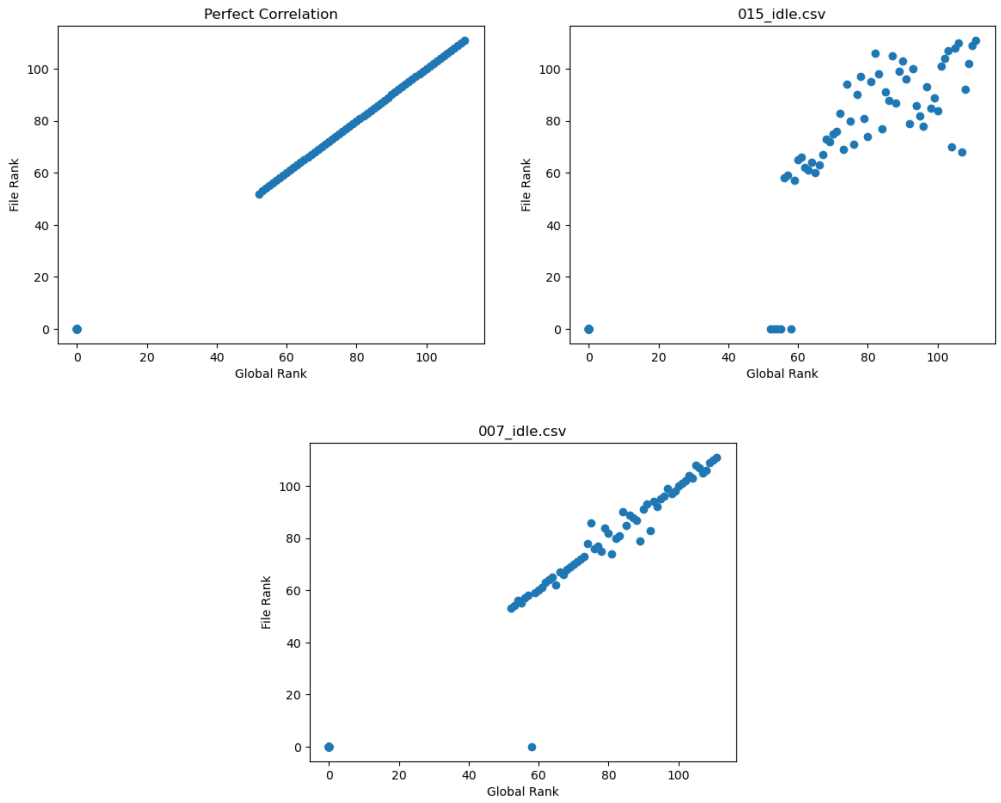


Figure 6.7: Spearman's perfect rank correlation in IDLE analysis, along with the results for the files with the minimum and maximum correlations. The scattered points represent the joints, with their coordinates indicating the ranks assigned to each joint.

6.4 Key findings

The analysis performed aimed to provide a detailed insight into the characteristics of the non-acted motions, uncover patterns, and calculate essential statistics. It was structured to facilitate better visualization and comprehension of the data, with a focus on identifying the most informative time series and understanding the motion variability across all subjects.

The analysis revealed several key findings, including:

- **Constant time series identification:** Out of the 111 time series, 49 were found to be constant across all files. Although these constant series are essential for maintaining correct joint motion, they can be excluded from modeling to reduce dimensionality, leaving 62 variable time series.
- **Variance analysis:** Variance analysis revealed that rotational data is significantly more variable and informative than positional data. This indicates that rotational degrees of freedom play a more critical role in capturing motion characteristics.

- **Positional data:** The positional data exhibited the least variability across all joints, with the pelvis showing the most variability. This finding suggests that positional data is less informative than rotational data when analyzing motion. But it is still essential for providing context to the rotational data.
- **Spearman's rank correlation:** The high mean correlation of 0.983098 across subjects in non-acted IDLE animations suggests a consistent pattern of motion variability. This uniformity in variance patterns supports the reliability of the dataset for further modeling and analysis.

CHAPTER 7

Conclusions and future work

In this chapter, new lines of work are proposed based on the findings and limitations encountered during the project. The following sections detail the proposed future work in the recording methodology, dataset recording, and dataset analysis.

7.1 Conclusions and personal contributions

This thesis addressed the challenge of capturing natural and subtle character movements, known as IDLE animations, for use in animation and video games. Traditional MoCap setups and the use of professional actors often hinder the capture of these unconscious actions.

To overcome this limitation, this work has presented a novel contribution: a cost-effective and non-intrusive MoCap recording methodology specifically designed for capturing IDLE animations. This methodology prioritizes affordability, ease of replication, and minimal impact on the recorded movements, empowering a wider range of users to generate their own MoCap data.

Furthermore, this research has introduced the first-ever publicly available dataset of 3D IDLE motions. This dataset encompasses a variety of common IDLE animations, fostering greater versatility in animation applications. A detailed analysis of the dataset provides valuable insights into the characteristics of the captured motions. All analysis tools and scripts used are publicly available, promoting further exploration and potential use with other MoCap data. The freely available dataset, combined with the user-friendly MoCap methodology, holds significant potential to advance the field of animation.

This combination can promote the development of machine learning techniques for generating diverse and realistic animations, while also enabling MoCap data creation and fostering wider participation in the animation process.

In summary, this thesis has made several key contributions:

- Development of a cost-effective and high-quality 3D MoCap methodology specifically tailored for capturing IDLE animations.
- Creation of a complete 3D MoCap pipeline integrating various technologies and tools for recording and reconstructing 3D animations.
- Establishment and publication of the first publicly available 3D IDLE motion dataset.
- Conducting an initial analysis of the dataset to characterize the variability and attributes of the IDLE animations.

All scripts and tools developed during this research are available on the project's GitHub repository [1], ensuring the reproducibility and extension of this work by the research community.

7.2 Future work

The current thesis has laid a solid groundwork, but there are several avenues for future research and development to enhance both the dataset's quality and analysis, and the motion capture methodology. The following sections detail the proposed future work in the recording methodology, dataset recording, and dataset analysis.

7.2.1 Motion capture methodology

Though the recording setup fits its purpose, it still has some limitations. The recordings need manual synchronization before 3D reconstruction, and faulty detections happen frequently in the reconstruction process due to variable lighting conditions or occlusions. This slows down the pipeline, takes longer to obtain the animation files, and requires manual work to filter and reinterpolate. In order to avoid manual synchronization, speed up the pipeline, and improve data quality, the following lines of work are proposed:

- **Automate the video synchronization:** The video synchronization was done manually by aligning the frames where the color change happened in the phone screen video. Despite significant efforts to record synchronously, network latency and the varying hardware used for recording mean that the videos cannot always be guaranteed to be perfectly synchronized, resulting in videos with different frame durations. Even though the difference was minimal, it is not valid to reconstruct the 3D animation.
- **Increase the number of cameras:** The more information recorded from different angles, the fewer occlusions, and therefore the more accurate the 3D reconstruction will be. Though more data will be needed to be processed by FreeMoCap, the trade-off between time consumed in processing and time consumed manually filtering and reinterpolating the data will be worth it.

[1] https://github.com/AAAbduu/Thesis-MoCap-4All

7.2.2 Dataset recording

The dataset obtained is a good starting point. But compared to other datasets such as the previously mentioned *TalkingWithHands16.2M*, it is quite small and limited in terms of the number of subjects and the variety of gestures. Therefore, future lines of work are explained below:

- **Increase the number of subjects:** In order to enhance the diversity and generalizability of the dataset, recording subjects with different age groups, ethnicities, and physical characteristics would be crucial, as motion capture output does take this into account. Additionally, increasing the number of subjects will lead to a more complete and robust dataset.
- **Add more IDLE gestures:** The dataset contains the most common IDLE gestures, but there are still many more that could be included. IDLE gestures are also influenced by context; therefore, adding some degree of interaction with the environment or driving the gestures with some emotion can result in more realistic animations.

7.2.3 Dataset analysis

Building on the findings from this analysis, several lines of work can be proposed for future research:

- **Dimension reduction:** 49 constant time series were identified across the dataset. Future work could explore the possibility of reducing the dimensionality of the dataset without losing critical information using techniques such as Principal Component Analysis (PCA).
- **Extend the analysis to more gestures:** The analysis focused on the raw IDLE animations. Future work could extend the analysis to other types of animations to understand the motion variability across different scenarios.

Appendix

Automatic synchronization using computer vision tools

In an effort to synchronize the cameras automatically, a script was initially written to start and stop the recordings simultaneously. However, this approach failed due to network latency and differences in the hardware of the connected computers. Despite this, the script was still useful for controlling all cameras from a single computer and retrieving all recordings to the same location. The most time-consuming part of the pipeline, synchronization, remained unsolved and required a manual solution.

Computer vision-based synchronization

Another approach attempted involved using computer vision tools to automate the synchronization process. The idea was to use the video recordings themselves since each recording included a smartphone screen at the beginning or end, which played a reference video. This reference video was used as a synchronization marker. Figure 4.2 illustrates the reference video used for synchronization.

The approach involved defining a Region Of Interest (ROI) around the smartphone screen in each recording. By using OpenCV, changes in each color channel (red, green, and blue) between consecutive frames were computed within this ROI. The goal was to establish a threshold that could detect significant color changes, particularly in the red and green channels, corresponding to specific frames in the reference video.

Process overview

Define Region Of Interest (ROI)

Identify and mark the area surrounding the smartphone screen. See Figure 1 for a visual representation of the ROI.

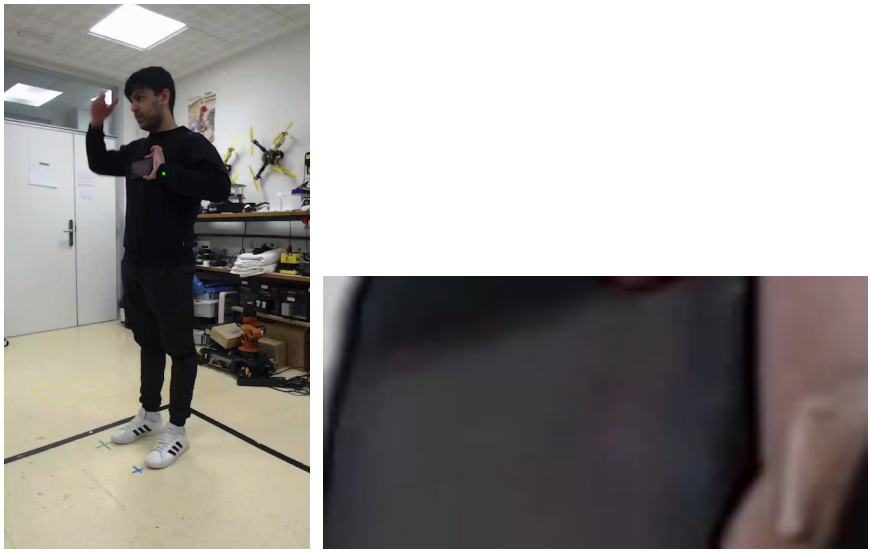


Figure 1: Region Of Interest (ROI) detected in the left image and amplified around the smartphone screen in the right image.

Compute color channel changes

Using OpenCV, analyze the changes in red and green color channels between consecutive frames within the defined ROI. This involves calculating the difference in pixel values for each color channel. The goal is to identify significant changes in color, particularly in the red and green channels. See Figure 2 for a visual representation of the color channel changes.

Figure 2: ROI in the left image and the difference in the red-green channel in the right image.

Infer threshold for synchronization

Determine a threshold to identify significant changes in color, focusing on the red-green channel difference. This threshold would ideally discriminate between frames when the reference video is showing a red screen and when it is showing a green screen. See Figure 3 for a visual representation of the color changes between consecutive frames.

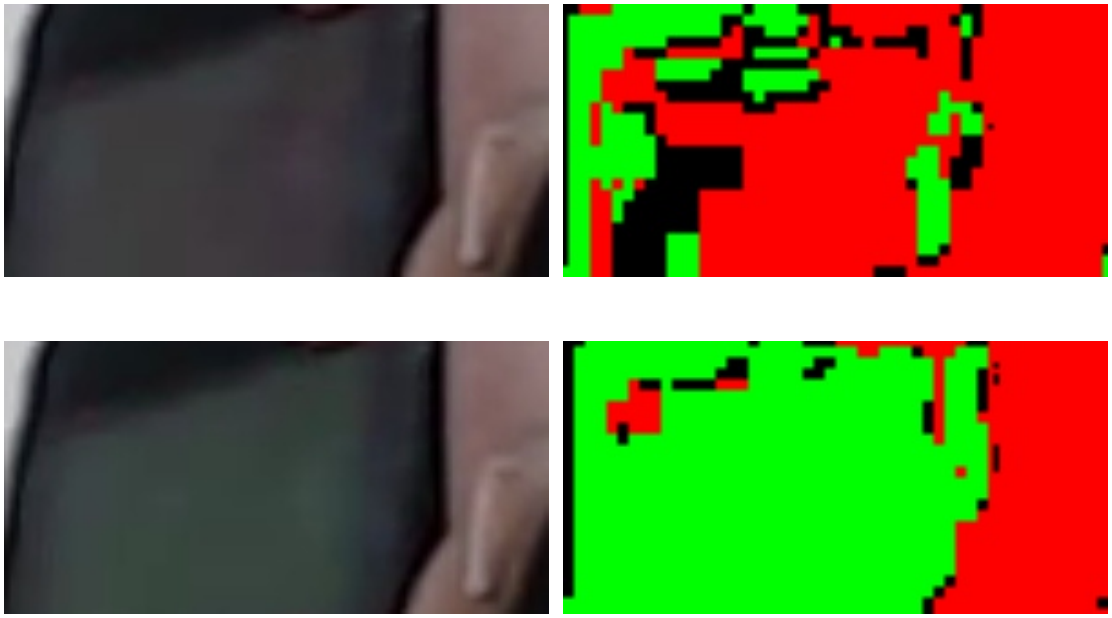


Figure 3: Difference in the red-green channel between consecutive frames where the change of the reference video happens.

Challenges

This approach faced several challenges that ultimately led to its abandonment and failure:

- **Lighting conditions:** Inconsistent lighting conditions across recordings made it difficult to establish a consistent threshold. Also reflections from lighting sources further complicated the color change detection.
- **Noise within the videos:** Variations in video quality and noise levels affected the accuracy of color change detection.
- **Inconsistent backgrounds:** Though the Region Of Interest (ROI) was defined around the smartphone screen, it moved due to the hand movements of the person holding the smartphone. This movement made it challenging to maintain a consistent ROI across all recordings and allowed for the inclusion of irrelevant background information.

Conclusion

Despite these challenges, the approach provides a valuable basis for future work. With improvements in noise reduction, lighting consistency, and background stabilization, this method could potentially become a viable solution for automatic synchronization of video recordings.

Post-processing joint removal

Removed Joints from Face

The following joints were removed from the face during post-processing using a script provided with Blender API. See Table 1 for the list of joints removed from the face.

Joints removed from the face
nose
lip.T.L
lip.B.L
jaw
chin
ear.L
brow.B.L
lid.T.L
lid.B.L
forehead.L
temple.L
jaw.L
chin.L
cheek.B.L
brow.T.L
eye.L
cheek.T.L
nose.L
teeth.B
tongue
jaw.L.001

Table 1: List of joints removed from the face during post-processing. Note that symmetrically the left counterparts were also removed.

Removed Joints from Hands

The following joints were removed from the hands during post-processing using a script provided with Blender API. See Table 2 for the list of joints removed from the hands.

Joints removed from hands
palm.01
f_index.01
f_index.02
f_index.03
palm.02
f_middle.01
f_middle.02
f_middle.03
palm.03
f_ring.01
f_ring.02
f_ring.03
palm.04
f_pinky.01
f_pinky.02
f_pinky.03
thumb.carpal
thumb.01
thumb.02
thumb.03
hand

Table 2: List of right joints removed from the hands during post-processing. Note that these joints were removed from both hands.

See Figure 4 for a visual representation of the joints removed from the face and hands.

The script used to remove the joints from the face and hands is available in the repository of this thesis [2]. The script can easily be modified to remove other joints from the body if needed.

[2]https://github.com/AAAbduu/Thesis-MoCap-4All

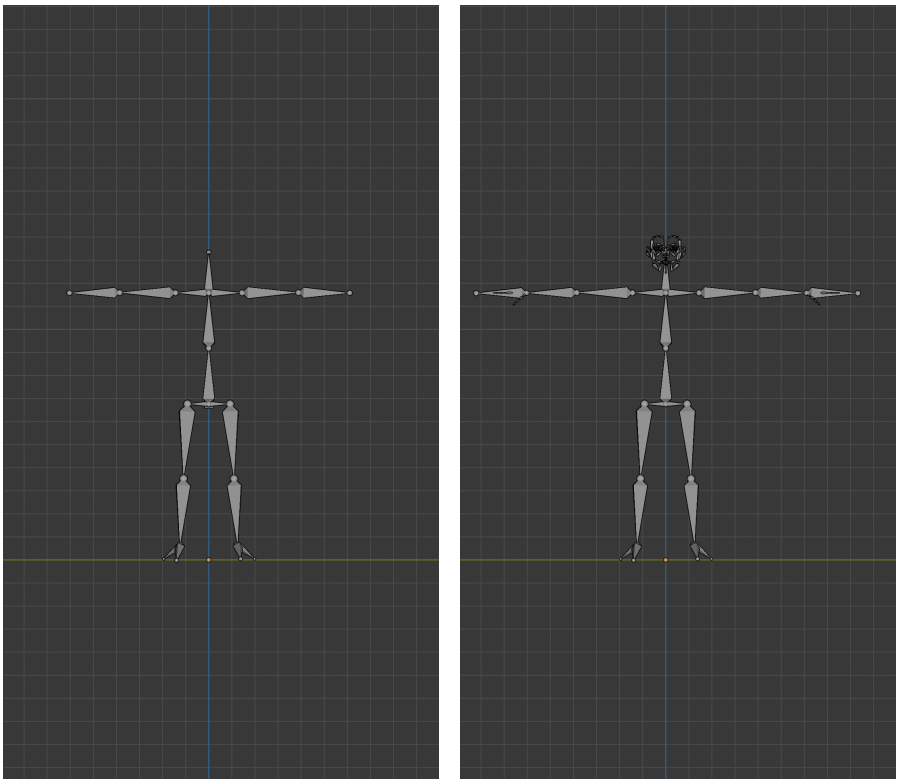


Figure 4: On the right: joints removed from the face and hands. On the left: raw skeleton as exported from FreeMoCap with all joints.

Camera calibration

In order to calibrate all the cameras within the setup, FreeMoCap provides in their website the necessary calibration board and the steps to follow to calibrate the cameras.

The calibration board is a checkerboard pattern that can be printed on a piece of paper. See Figure 5 for a visual representation of the calibration board.

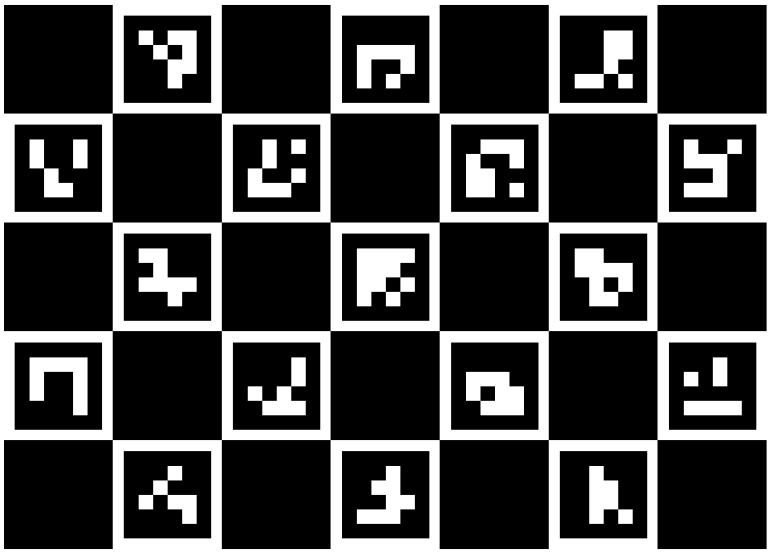


Figure 5: Calibration board used for camera calibration.

The approach we used was to stick the printed board onto a cardboard. Then in order to calibrate the cameras, the board was sustained by someone, moved and rotated in front of the cameras to ensure all corners and patterns were visible for all cameras. See Figure 6 for a visual representation of the calibration process.

Figure 6: Calibration process using the calibration board.

This was done every time the cameras were moved or the setup was changed and also before every recording session to ensure the cameras were correctly calibrated.

After recording and synchronizing the calibration sequences, they were processed using FreeMoCap, which automated the calibration process. FreeMoCap analyzed the recordings to calculate the intrinsic and extrinsic parameters of the cameras. The intrinsic parameters include the focal length, principal point, and lens distortion, while the extrinsic parameters define the position and orientation of the cameras in relation to the calibration board.

The resulting calibration parameters were saved in a file, which was then loaded into FreeMoCap prior to loading the motion capture session recordings. This ensured that the cameras were correctly calibrated and that the motion capture data was accurately processed and synchronized.

Bibliography

[1] Shubham Sharma, Shubhankar Verma, Mohit Kumar, and Lavanya Sharma. Use of Motion Capture in 3D Animation: Motion Capture Systems, Challenges, and Recent Trends. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 289–294, February 2019. See page 1.

[2] Arjan Egges, Tom Molet, and Nadia Thalmann. Personalised Real-Time Idle Motion Synthesis. pages 121–130, January 2004. See pages 2, 9.

[3] Brian Ravenet. IdlePose : A Dataset of Spontaneous Idle Motions. In *Companion Publication of the 2021 International Conference on Multimodal Interaction*, ICMI '21 Companion, pages 164–168, New York, NY, USA, December 2021. Association for Computing Machinery. See pages 2, 10, and 24.

[4] Gutemberg Guerra Filho. Optical motion capture: Theory and implementation. *RITA*, 12:61–90, 01 2005. See page 6.

[5] Rahul M. Review on motion capture technology. *Global Journal of Computer Science and Technology*, 18(F1):23–26, Jan. 2018. See page 6.

[6] Bo Feng, Xianggang Zhang, and Huilong Zhao. The research of motion capture technology based on inertial measurement. In *2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*, pages 238–243, 2013. See page 7.

[7] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines, 2019. See pages 8, 19.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. See page 8.

[9] Carnegie Mellon University - CMU Graphics Lab - motion capture library. See page 9.

[10] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. See page 9.

[11] Jonathan Williams. AMASS. See page 9.

[12] Gilwoo Lee, Zhiwei Deng, Shugao Ma, Takaaki Shiratori, Siddhartha Srinivasa, and Yaser Sheikh. Talking With Hands 16.2M: A Large-Scale Dataset of Synchronized Body-Finger Motion and Audio for Conversational Motion Analysis and Synthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 763–772, October 2019. ISSN: 2380-7504. See page 9.

[13] Taras Kucherenko, Rajmund Nagy, Youngwoo Yoon, Jieyeon Woo, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. The GENEA Challenge 2023: A large-scale evaluation of gesture generation models in monadic and dyadic settings. In *Proceedings of the ACM International Conference on Multimodal Interaction*, ICMI '23. ACM, 2023. See page 9.

[14] Maja Kocoń. Idle Motion Synthesis of Human Head and Face in Virtual Reality Environment. In Minhua Ma, Manuel Fradinho Oliveira, Sobah Petersen, and Jannicke Baalsrud Hauge, editors, *Serious Games Development and Applications*, pages 299–306, Berlin, Heidelberg, 2013. Springer. See page 9.

[15] Philip Queen, Aaron Cherian, Wirth Trent, Idehen Endurance, and Jonathan Samir Matthis. Freemocap: A free, open source markerless motion capture system, 2023. See page 19.

[16] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. See page 19.

[17] Charles R. Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. See page 19.

[18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. See page 19.

[19] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking, 2020. See page 19.

[20] Pierre Karashchuk, Katie L. Rupp, Evyn S. Dickinson, Sarah Walling-Bell, Elischa Sanders, Eiman Azim, Bingni W. Brunton, and John C. Tuthill. Anipose: A toolkit for robust markerless 3D pose estimation. *Cell Reports*, 36(13):109730, September 2021. See page 20.