



SAPIENZA
UNIVERSITÀ DI ROMA

Detection and segmentation of locomotor cycle in mice movement using processed data from marker-based 3D motion capture on voluntary treadmill running

Faculty of Information Engineering, Informatics, and Statistics
Master Course in Computer Science

Federico Barreca
ID number 1736423

Advisor
Prof. Maria De Marsico

A handwritten signature in black ink that reads "Maria De Marsico".

Co-Advisor
Lakshmipriya Swaminathan

Academic Year 2023/2024

**Detection and segmentation of locomotor cycle in mice movement using processed
data from marker-based 3D motion capture on voluntary treadmill running**

Master Thesis. Sapienza University of Rome

© 2024 Federico Barreca. All rights reserved

This thesis has been typeset by L^AT_EX and the Sapthesis class.

Author's email: federico.barreca97@gmail.com

Abstract

Locomotion and the central nervous system are connected through a mapping that can be identified by analyzing the movement of an animal. By studying a complex and articulated movement produced by the coordination of the whole body of a mouse, distinct repetitions of circular patterns representing different types of gaits can be observed. This research aims to detect and isolate these patterns, called "Unitary Movements". To achieve this, we process data collected from 10 different parts of a mouse's body using 3D markers and state-of-the-art motion capture technology specifically adjusted for mice. The acquisition of high-quality mice locomotion data makes a significant contribution to the existing literature. Additionally, the 30-dimensional trajectory of the mouse body serves as the input for an innovative method to extract unitary movements, utilizing a sophisticated Recurrence Quantification Analysis. Furthermore, the study includes a decomposition analysis of the movement using Principal Component Analysis to enhance the understanding of the changes in body configuration during locomotion. My contributions to this work include essential data processing tasks such as labeling and gap-filling the motion capture data. To determine a threshold for an accurate and automatic application of delay embedding techniques, I proposed and implemented a vision-based method to compute optimum delay embedding parameter by quantifying the unfolding of the underlying attractor. Finally, I manipulated raw data to create a portable, efficient, and user-friendly visualization tool that will also be useful for future studies on mice and other animals.

Acknowledgments

I would like to thank all the people who made it possible for me to develop this thesis, first and foremost Professor Maria De Marsico, for accepting me as a thesis student and supervising my work.

I would like to express my gratitude to Lakshmipriya Swaminathan for allowing me to base my master's thesis on her project, to Professor Marylka Yoe Uusisaari for giving me the opportunity to join the research internship, and to all the members of the Neuronal Rhythms in Movement unit for their kindness.

The research was supported by Japan Society for Promotion of Science Kakenhi Grant for Scientific Research (21K06399) awarded to Bogna Ignatowska-Jankowska.

My research internship at the Okinawa Institute of Science and Technology was supported by full intramural funding.

A special thanks goes to my family, Natsuki, and my friends for their support throughout this long-distance experience. I also want to thank my colleagues for the work we did together and the experiences we shared over the past two years.



Contents

1	Introduction	1
2	Background	3
2.1	Neural Manifolds	3
2.2	Tracking Animal Movement	4
2.2.1	Muybridge's Horse	4
2.2.2	Mouse Tracking on Plexiglass	4
2.2.3	Motion Capture Techniques in 2D	5
2.2.4	Motion Capture Techniques in 3D	5
2.3	Principal Component Analysis of Animal Movement Data	5
2.3.1	Data Visualization	6
2.3.2	Noise Reduction	6
2.3.3	Feature Extraction	6
2.3.4	Movement Decomposition	6
2.4	About Recurrence Quantification Analysis	6
2.4.1	Applications of RQA	7
2.5	Why Work with Mice?	7
3	Data Collection	8
3.1	Equipment Specifications	8
3.2	Environment Setup for Mice	8
3.2.1	Marker Engineering	9
3.3	Qualisys Track Manager Software	11
3.4	Experiments	12
3.5	Known Limitations	13
4	Data Processing	14
4.1	Data Labeling	14
4.2	Automatic Gap-Filling	15
4.3	Manual Gap-Filling	15
4.3.1	Linear Interpolation	15
4.3.2	Polynomial Interpolation	15
4.3.3	Relational Interpolation	16
4.3.4	Interpolation Choice and Usage Criteria	16
4.4	Data Exportation	17
4.5	Understanding 30-dimensional Data	17

4.6	Egocentric Coordinates Transform	18
5	Principal Component Analysis	19
5.1	Data Pre-processing for PCA	19
5.2	Evaluation of the Principal Components	20
5.2.1	Heuristics	20
5.2.2	Modes of Deformation	22
6	Recurrence Quantification Analysis	23
6.1	Recurrence Plot	23
6.2	Dimensionality Reduction	24
6.2.1	UMAP	25
6.2.2	ISOMAP	25
6.3	Cycle Detection Methods	26
6.3.1	Frequency Analysis	26
6.3.2	Unitary Movements	27
6.4	Delay Embedding	28
6.5	Threshold Criteria for Delay Embedding	30
6.5.1	Recurrence Plot Points Correlation	31
6.5.2	Quantification using Blur Detection	32
7	Web Application for Data Visualization	34
7.1	Aim of Mouse Movement	34
7.2	Software Design and Use Cases	34
7.3	Software Architecture	39
7.4	Implementation	40
7.5	Known Issues and Limitations	43
7.6	Possible Applications	43
8	Ethics and Animal Care	44
9	Conclusions	45
9.1	Future remarks	45
9.1.1	Evaluation of Performance	47
9.1.2	OIST Research Internship Certificate of Completion	48
	Bibliography	49

Chapter 1

Introduction

I developed most of my master's thesis during a research internship that took place from July 18 to October 1, 2024 at the "Neuronal Rhythms in Movement" unit, led by Professor Marylka Yoe Uusisaari, at the Okinawa Institute of Science and Technology (OIST). I had the opportunity to work in an authentic research environment in Japan, where I gained initial insights into the research world by interacting with other students and colleagues. I actively participated in work-in-progress meetings and discussed materials for these sessions. Additionally, I assisted PhD student Lakshmipriya Swaminathan and actively contributed to her research project titled "Of Movement and Mice: A Case Study of Movement Variability Using Marker-Based 3D Motion Capture and Mathematical Representations of Voluntary Treadmill Locomotion". The unit provides a range of cutting-edge equipment and professional expertise for managing it. For example, it includes a dedicated motion capture room where data involving animals, such as mice in the case of my thesis, can be meticulously collected, all respecting international standards of safety and ethical protocols. Animal experiments were coordinated by Dr. Bogna Ignatowska-Jankowska, the principal investigator and supervisor. Tara Helmi Turkki was responsible for animal handling, implantation procedures, and behavioral tasks, while Lakshmipriya Swaminathan managed the operation of the motion capture system and data acquisition. With such advanced technology available, it is possible to thoroughly investigate neurological aspects of mice in movement, where an approach combining computing and data analysis is particularly effective. The questions addressed by this project range from "What does the whole body do when a mouse takes a step?" to "What is the role of movement variability?". By addressing these questions, we succeeded in isolating locomotor cycles, for the first time defined as "Unitary Movements", which represent different gaits. This allowed us to identify the frames that delimit their occurrence. The workflow to achieve this goal involves the following steps:

1. Data collection using 3D marker-based motion capture from mice;
2. Data processing, which consists of labeling data, gap-filling, and applying the egocentric transform;
3. Decomposing movement using Principal Component Analysis on the processed data from step 2 to identify Modes of Deformation and study the gaits;

4. Detecting and segmenting Unitary Movements using Recurrence Quantification Analysis on data from step 2;
5. Comparing gaits from step 3 with the animation using a custom visualization tool, considering the Unitary Movements obtained from step 4.

I contributed significantly to step 2 by labeling data and performing manual gap-filling. I assisted in the segmentation phase of step 4 by proposing and implementing a vision-based selection criterion to identify the Unitary Movements, and I handled step 5 in its entirety. This paper, currently in preparation, will feature me as a co-author due to my aforementioned contributions.

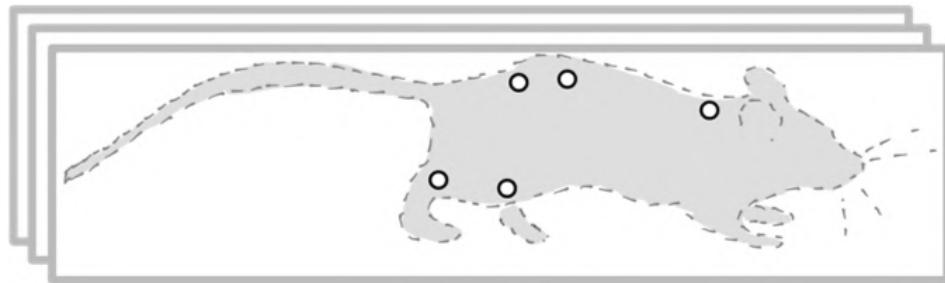


Figure 1.1. Half of the mouse markers configuration as viewed from the side. Image credit: Lakshmipriya Swaminathan (LS)

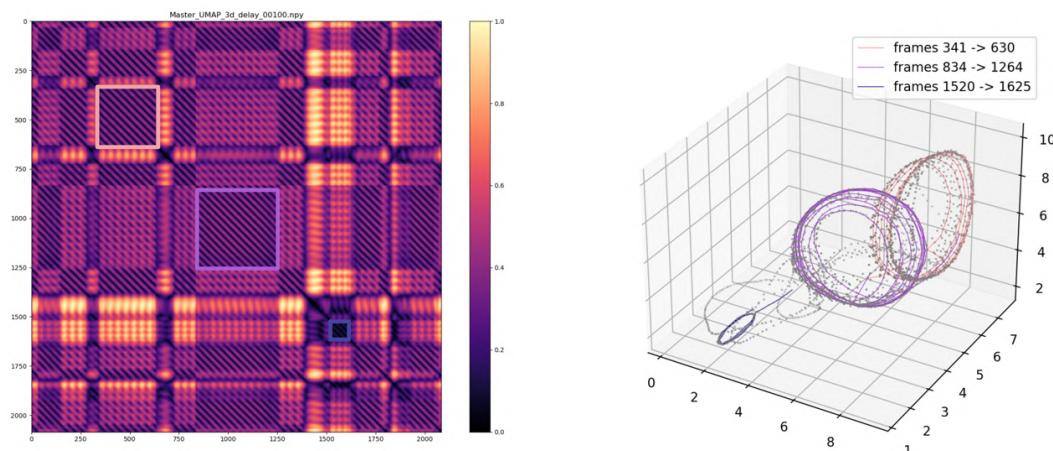


Figure 1.2. On the left, a Recurrence plot computed from the distances of the motion captured time series with itself, where the axes represent the timestamps, or frames, in milliseconds. On the right, a visualization of Unitary Movements derived from the same time series after applying delay embedding and reducing dimensionality using UMAP to three dimensions. The plot displays a set of 3D points on axes X, Y and Z representing the time series for each frame. Image credit: LS

Chapter 2

Background

2.1 Neural Manifolds

The main focus of the project is on neural manifolds, which represent the neurons involved in movement control. Specifically, a definition is provided by Sadler et al. [1], who state that neural activity is inherently constrained by the properties of the physical network circuitry itself. These constraints result in neural activity patterns that form a low-dimensional subspace, the manifold, within the larger high-dimensional neural space (Figure 2.2). Historically, in the 1990s, spinal transection on adult cats [2] revealed that by severing the connection between the brain and the spinal cord, a decerebrate cat could continue walking [3]. This demonstrated that movement is controlled by the spinal cord, and not only by the brain. Furthermore, by stimulating a frog limb and mapping the movement according to the position and intensity of the stimulus, the concept of vector muscle force fields was introduced [4]. Muscle force fields are inherent ways in which the body can move, as shown in Figure 2.1. Consequently, an initial insight was to identify a mapping between muscle force fields, i.e., how movement can be performed, and neural manifolds [5]. Thus, research began on reverse-engineering locomotion data due to the significant difficulty in directly recording from neurons.

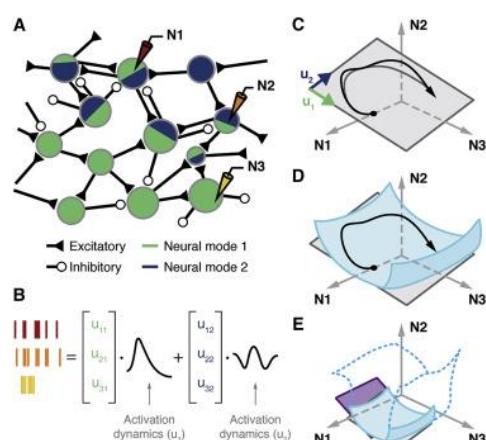
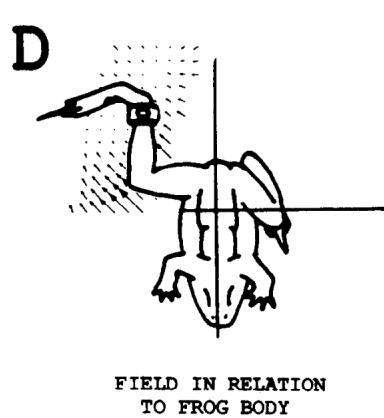


Figure 2.1. Frog limb stimulation generating a force field. Image credit: [4]

Figure 2.2. Neural manifolds. Image credit: [5]

2.2 Tracking Animal Movement

Data collected by tracking animals with high-quality equipment can be used to build a detailed description of their movements. In particular, locomotion involves the coordinated movement of the entire body.

2.2.1 Muybridge's Horse

In 1877, one first attempt to record animal movement was made by Eadweard Muybridge with horses to produce cabinet cards [6]. A horse was made to run along a track equipped with cables and camera triggers. Each time the horse stepped on a trigger, a photo of the horse's body configuration was taken, capturing spatiotemporal information (Figure 2.3). This technique is known as chronophotography of animal locomotion.

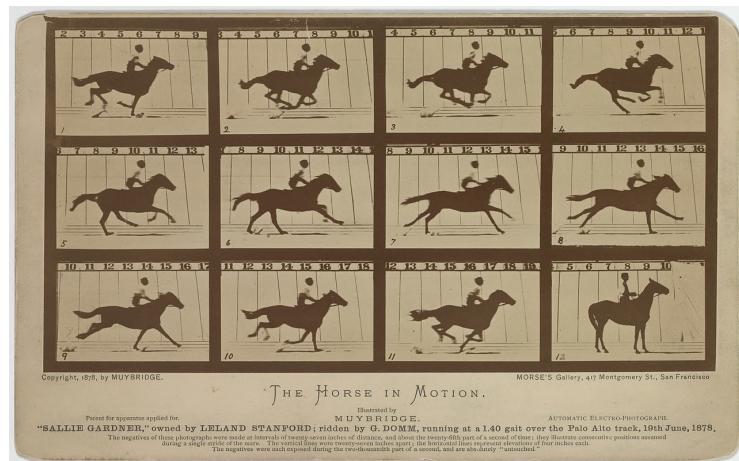


Figure 2.3. "Sallie Gardner", owned by Leland Stanford; ridden by G. Domm, running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

2.2.2 Mouse Tracking on Plexiglass

A simpler yet effective technique for capturing mouse movement involves applying ink to the mouse's paws before it walks on a sheet of plexiglass. The transparency of the plexiglass allows researchers to study the animal's behavior and gather various details, such as gait direction, speed, and step width (Figure 2.4).

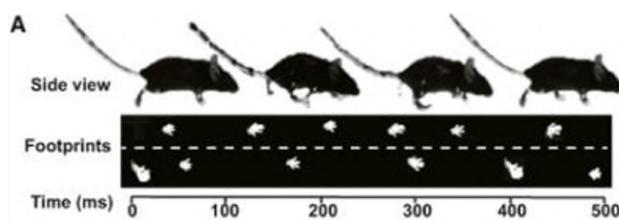


Figure 2.4. Mouse gait printed on plexiglass. Image credit: [7]

2.2.3 Motion Capture Techniques in 2D

In the literature, there is a wide variety of computer vision tools available for tracking objects and animals in motion using 2D motion capture. Among these, some of the most popular are DeepLabCut, a markerless pose estimation system for user-defined body parts using deep learning [8], and SLEAP, a deep learning system for multi-animal pose tracking [9], both of which are commonly used to study mouse behavior. Additionally, DeepLabCut has recently been used to develop a system, called PyMouseTracks (Figure 2.5) for multiple rodent tracking and behavior assessment that can be set up within minutes in any user-defined arena at minimal cost [10].

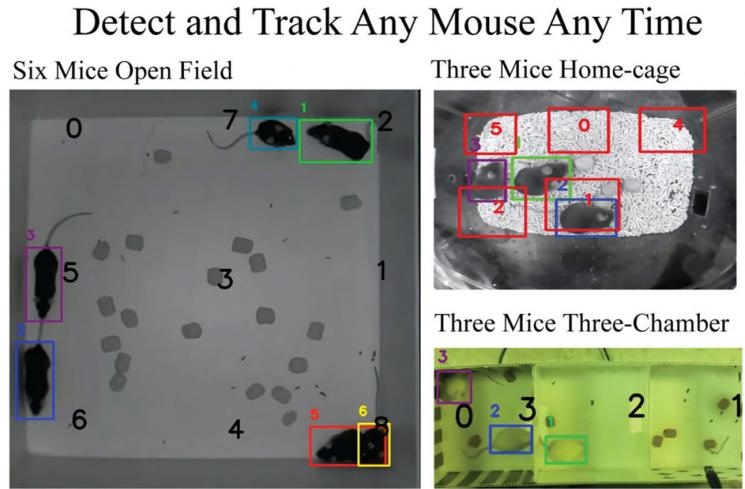


Figure 2.5. PyMouseTracks detecting mice. Image credit: [10]

2.2.4 Motion Capture Techniques in 3D

Performing 3D motion capture is more challenging but provides a greater amount of information. In a movement study, having complete spatial information enables a more consistent data analysis. In this context, we consider Anipose, an open-source toolkit for robust markerless 3D pose estimation [11] and Qualisys Track Manager (QTM) made by Qualisys [12], the software we used for this research.

2.3 Principal Component Analysis of Animal Movement Data

Principal Component Analysis (PCA) is a statistical self-supervised technique used for dimensionality reduction while preserving as much variability in the data as possible. Developed by the mathematician Herman Hotelling [13] in the 1930s, PCA is widely used in various fields including machine learning, image processing, and data visualization. We can find various applications:

2.3.1 Data Visualization

PCA reduces the dimensionality of data to two or three principal components, making it possible to visualize high-dimensional data in lower-dimensional space.

2.3.2 Noise Reduction

By focusing on the principal components that capture the most variance, PCA can help reduce noise and enhance the signal in data.

2.3.3 Feature Extraction

PCA helps in extracting the most informative features from large datasets, which can be beneficial for further analysis or modeling.

2.3.4 Movement Decomposition

Analyzing animal behavior often involves simplifying complex motor actions obtaining the principal components that are linear combinations of the original features. Stephens et al. [14] show that the movement patterns of the nematode *Caenorhabditis elegans* can be effectively described using just four principal components from Principal Component Analysis, which account for 95% of the shape variance. These dimensions help reconstruct the "equations of motion" and show that its behavior is influenced by several different attractors in this space (Figure 2.6).

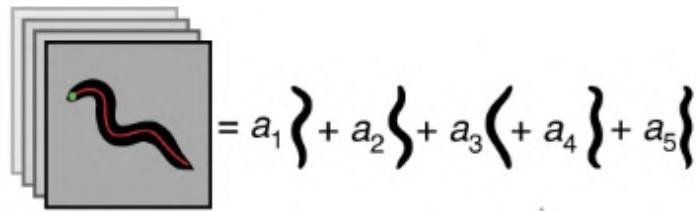


Figure 2.6. Movement of a *C. elegans* represented as a linear combination of deformations computed using PCA. Image credit: [15].

2.4 About Recurrence Quantification Analysis

Recurrence Quantification Analysis (RQA) [16] is a technique used to analyze time series data by examining the patterns of recurrence within the data. It is particularly useful for studying complex and dynamic systems, revealing insights into their behavior by quantifying recurring patterns or states. In the context of dynamic systems, "recurrence" refers to the idea that a system may return to similar states or configurations at different times. RQA is based on the concept that certain states of a system repeat or show regular behavior over time. To analyze recurrences, RQA constructs a recurrence matrix, which represents the presence of recurrences of states in the system's phase space. Each element of the matrix indicates whether a given state at one time is similar to a state at another time. Recurrence plots are distance

matrices which visualise the recurrence behaviour of the phase space trajectory $\vec{x}(i)$ of dynamical systems:

$$\mathcal{R}(i, j) = \Theta(\epsilon - \|\vec{x}(i) - \vec{x}(j)\|) \quad (2.1)$$

where:

- $\Theta : R \rightarrow \{0, 1\}$ is the Heaviside function;
- ϵ is a predefined tolerance.

2.4.1 Applications of RQA

In physiology, it helps identify anomalies in heartbeats by analyzing electrocardiogram (ECG) signals [17]. In finance, it reveals recurring patterns in stock market data [18]. In meteorology and climatology, RQA is employed to study climate data and identify cycles or patterns over time, enhancing weather prediction and climate modeling [19]. It's also useful in neuroscience to understand brain activity patterns [20].

2.5 Why Work with Mice?

The animal chosen for this research is the mouse, as it is a modal organism widely used to study various behavioral tasks and paradigms. Additionally, its physiological properties can be extended to humans due to similarities in the musculoskeletal structure. Furthermore, mice represent a homogeneous population [21], and multiple genetic manipulation tools can be used to test specific model conditions.

Chapter 3

Data Collection

The Qualisys system, renowned for its leading role in studying biomechanics in both humans and animals, is widely regarded as one of the best in the world. In this laboratory at OIST, it has been adapted and installed for the first time for use with mice [22]. In this chapter, I will provide an overview of this complex setup. The toolset used to collect data on the rodents' trajectories is based on Qualisys, a provider of motion capture and 3D tracking systems that offers both software and hardware, as well as guidelines for setup to maximize efficiency, given the high stability and compatibility of its components.

3.1 Equipment Specifications

The equipment includes:

- 6 Qualisys **Oqus** infrared (IR) cameras with a frame rate of up to 300 Hz and a sensor resolution of 12 MP, used for motion capture and quantification purposes;
- 1 Qualisys **Miqus** RGB camera with a frame rate of 85 Hz, 4 MP resolution, and a wide field of view, used only for qualification purposes;
- A treadmill with a speed of up to $100 \frac{m}{min}$;
- 10 passive custom markers covered in retro-reflective material.

3.2 Environment Setup for Mice

Given the small size of the mice, a custom setup was required. On a perforated metal-surface table, axes and angles were marked with insulating tape, serving as a reference for the central positioning of the treadmill. Around the setup, the 6 infrared cameras were installed at equidistant positions, with an angle of 60° between each camera. This configuration is not specifically optimized for treadmill use, but it is designed to be usable with other tasks, such as open field or climbing activities. A RGB camera is also installed next to an IR camera for a perspective view.



Figure 3.1. Environment setup. Image credit: Federico Barreca (FB)

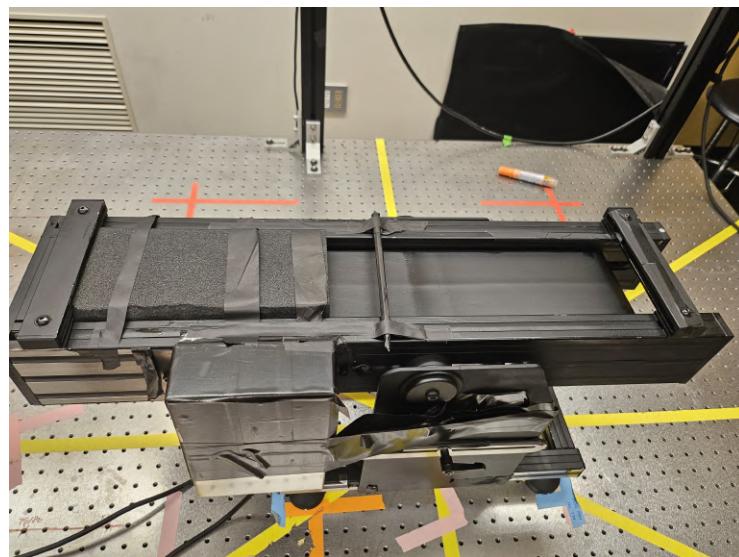


Figure 3.2. Treadmill closeup. Image credit: FB

3.2.1 Marker Engineering

We decided to apply 10 markers to the rodents' bodies at key locations where movement is most noticeable. These markers were placed symmetrically on spots such as the shoulders, pelvis (named coordinates), hips, knees, and ankles, as shown in Figure 3.3.

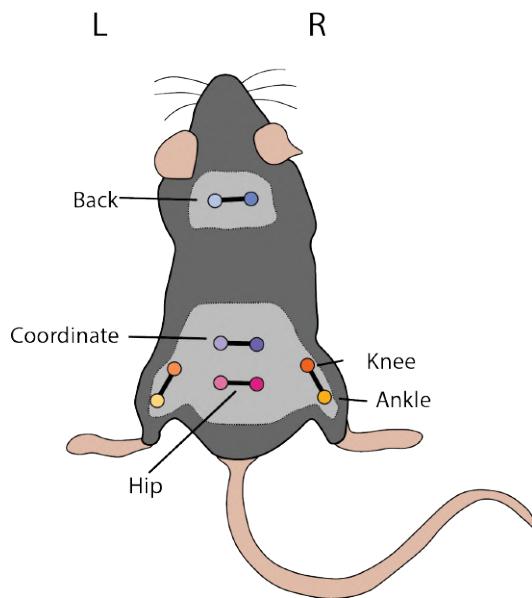


Figure 3.3. Marker locations on the body of the mouse. Image credit: LS

However, due to the small size of the mice and their fur, it was quite challenging to find an effective way to attach the markers securely. We spent effort trying different methods and eventually had to manually adjust the size of the markers as well as the way they were applied. After experimenting, we found that using a removable skin implant method worked better than using adhesive markers, as it provided a more reliable attachment. Custom markers were manufactured using stainless steel balls from piercings. We used balls with diameters between 3 mm and 4 mm . Each ball was wrapped with retro-reflective tape, covered with a clear film, and then coated with UV hardening glue for protection. These layers created strong, reflective markers that could be attached to the mice without breaking easily, even if the mice tried to groom or bite them (Figure 3.5). It was not possible to apply markers to the front legs due to their small size, which resulted in a significant loss of data quality. In addition, two reference markers were attached to one end of the treadmill support, and some adhesive markers were placed on the belt to provide a reference for the speed.

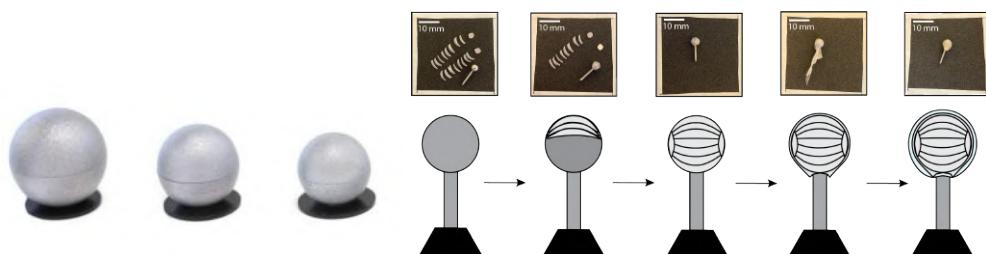


Figure 3.4. Qualisys default markers. Image credit: [23]

Figure 3.5. The process of creating custom markers begins with a steel ball, which is then attached with retroreflective tape and covered with a clear film. Image credit: LS

3.3 Qualisys Track Manager Software

The whole camera setup is orchestrated by Qualisys Track Manager, version 2022.2, running on Windows 11. Qualisys cameras emit rings of infrared light asynchronously for very short periods. To prevent cameras from self-capturing and causing interference, a delayed exposure of $50\ \mu s$ is set. The software builds a camera mapping model by grouping cameras in pairs, with each camera capturing images every $50\ \mu s$, ensuring a total shooting time of 3.3 ms and a consistent frame rate of 300 *fps*. There is a break time between each shot (Figure 3.6).

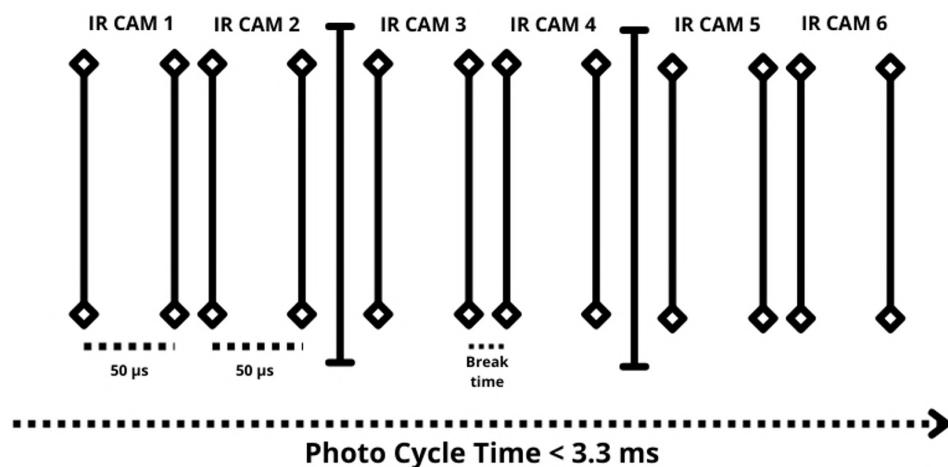


Figure 3.6. IR cameras photo cycle. Image credit: FB

Marker capture and triangulation algorithms are performed directly on the camera's integrated hardware, which reduces the load on the server. Finally, for proper software use, calibration is necessary. This involves placing a support surface at the treadmill's height and simulating mouse movement with specific markers to cover as much volume as possible as shown in Figure 3.7. The cameras were calibrated before recordings and re-calibrated every 4 hours. Qualisys Track Manager also offers an editor for data visualization and data processing (Figure 3.8).



Figure 3.7. Calibration process. Image credit: FB

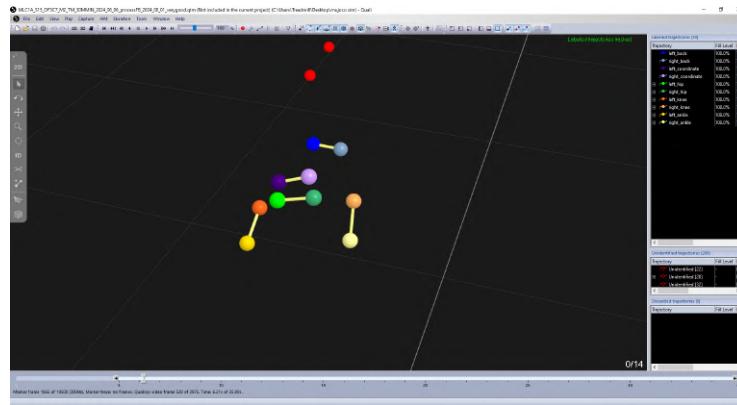


Figure 3.8. Qualisys Track Manager editor showing marker trajectories and bones. Image credit: FB

3.4 Experiments

The experiments conducted followed a well-defined structure on which the file naming convention is based:

- <experiment name>;
- <sample number>;
- <cage ID>, which identifies the mouse's cage;
- <mouse number>, which identifies a mouse from a specific cage;
- <task ID>, which can be: "OF" for Open-Field, "TM" for TreadMill, "CLB" for Climbing, and "MRKR_CLBRTN" for Marker Calibration;
- <detail of task>, if applicable (for example, treadmill speed value);
- <acquisition date>, in "YYYY_MM_DD" format;
- <name of processing operator_processing date>, if processed.

The concatenation of these parameters using underscores ("_") generates the file name (Figure 3.9).

MLC1A_S15_OF3C7_M2_TM_30MMIN_2024_06_06_processFB_2024_08_01

Figure 3.9. Example of file name.

Note that for this study, only the treadmill task data was considered. Every experimental trial (or acquisition) corresponds to a recording of about 35 seconds and, in any case, no more than one minute. Finally, each task was performed for each of the following belt speeds: 10, 15, 20, 30 (Figure 3.10), 40 meters per minute.



Figure 3.10. Mouse running on treadmill at $30 \frac{m}{min}$. Image credits: [24], [25]

3.5 Known Limitations

Three major challenges were predicted: mice training to be conducted due to the absence of barriers, the need for marker adaptation for mice, and the necessary implants. Difficulties were expected during data acquisition, some of which were directly related to the motion capture system itself. Tracking loss was observed due to occlusions such as fur and other body parts, as well as the high precision required by the system to detect small-sized markers. Some trials had to be discarded due to lack of cooperation from the mice. Additionally, we observed mistracking of markers and false positives caused by poor calibration and damaged cables.

Chapter 4

Data Processing

One of my main tasks and contributions was processing the data collected via Qualisys. I initially identified the markers in the time series and, when necessary due to incompleteness, performed manual gap-filling using various types of interpolation based on the duration of the gaps.

4.1 Data Labeling

For a complete and consistent analysis, it is necessary to identify and label the tracked parts of the mouse. Initially, the first 5 seconds of the time series are excluded, considering a total of 30 seconds. Next, I manually inspect the animation frame-by-frame, assigning the corresponding label to each marker. This work requires a lot of concentration and care because there can be visually overlapping mistracked markers in the 3D environment. Sometimes, it is also necessary to intervene in the automatic gap-filling system, which can be susceptible to ambiguity if two markers are too close to each other. During the data review, I came across some trials where the mouse was removed from the treadmill due to insufficient cooperation. In these cases, the trial is discarded by adding "BADTRIAL" to the end of the file name.

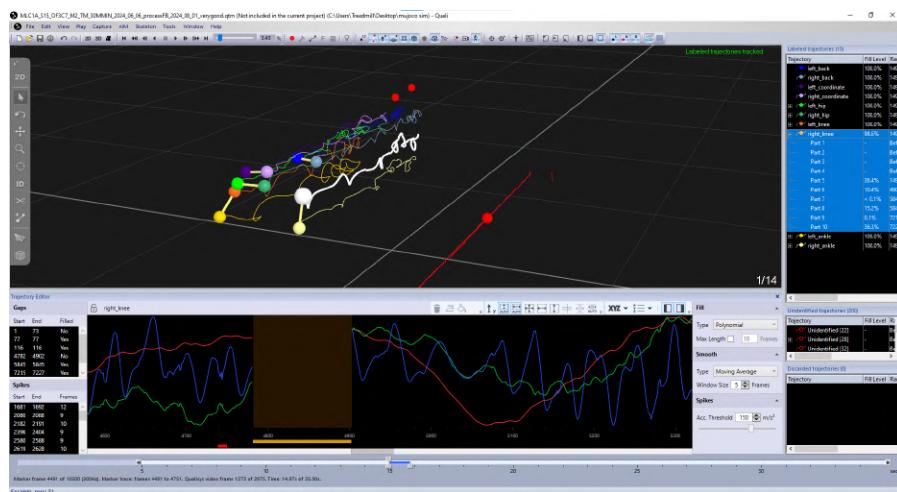


Figure 4.1. Trajectory analysis on Qualisys Track Manager. Image credit: FB

4.2 Automatic Gap-Filling

Even though Qualisys Track Manager provides an automatic gap-filling functionality, we set the range threshold to 30 ms , about 3-4 frames, because as the gap length increases, the number of incorrect labels also increases.

4.3 Manual Gap-Filling

Given the limitations of automatic gap-filling, Qualisys Track Manager provides a manual system (Figure 4.2) in which the user can choose the type of interpolation, such as static, linear, polynomial, relational, virtual, or kinematic. For this task, I only used linear, polynomial, and relational interpolation methods, whose functionalities are briefly described in the Qualisys documentation [26].

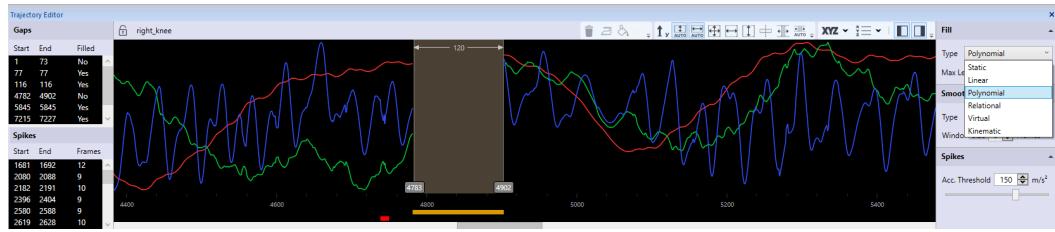


Figure 4.2. Manual gap-filling on Qualisys Track Manager. Image credit: FB

4.3.1 Linear Interpolation

The linear interpolation method addresses gaps by drawing a straight line (Figure 4.3) between the coordinates on either side of the gap, in the X, Y, and Z dimensions. This technique is especially useful for bridging small gaps or for tracking objects that are moving at a constant speed. If a gap occurs at the beginning of a trajectory, it is filled with the value of the first data point following the gap. Conversely, when a gap is found at the end of a trajectory, it is filled with the value of the last data point recorded before the gap.

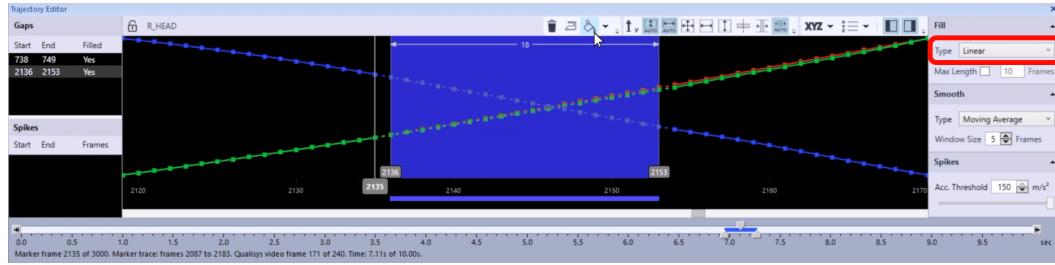


Figure 4.3. Linear fill. Image credit: [26]

4.3.2 Polynomial Interpolation

The polynomial interpolation method employs an algorithm to smoothly connect the X, Y, and Z trajectories across the gap. Polynomial gap-filling requires trajectory

data on both sides of the gap, as it depends on this surrounding data to generate the interpolated curves (Figure 4.4). Since this method interpolates data, caution is advised when using it for large gaps. To avoid excessive interpolation, it may be prudent to set a maximum frame limit for which gaps can be filled. If a gap is too extensive for Polynomial interpolation and the trajectory is part of a cluster of markers that could serve as reference points, it is worth considering relational gap-filling method instead.

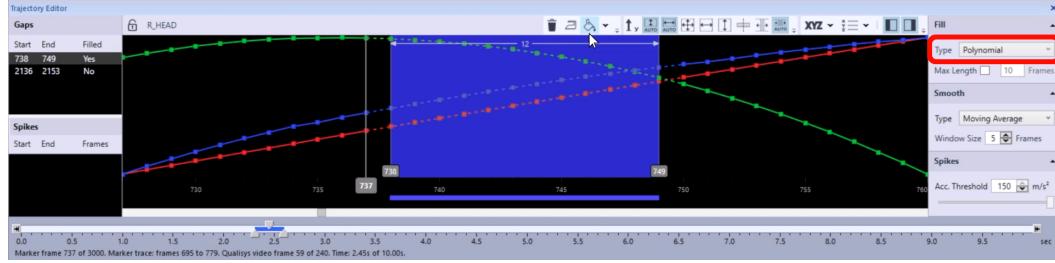


Figure 4.4. Polynomial fill. Image credit: [26]

4.3.3 Relational Interpolation

The relational interpolation method links the X, Y, and Z trajectories across a gap based on the movement of adjacent markers. This approach is particularly effective when tracking a group of markers that maintain a consistent relative position, such as a cluster of markers (Figure 4.5). To use this method, I select up to three context markers, which will establish a local coordinate system for the interpolation of the missing data. These markers can be chosen from drop-down menus or dragged from the 3D viewer or the trajectory information pane into the settings sidebar. One of the selected markers will serve as the origin for the local coordinate system. Additionally, I choose a marker to define the X-axis and another to establish the XY plane. For example, filling the gap of the left ankle could be based on the movement of the left knee, or the hips could be determined by the pelvis coordinates.

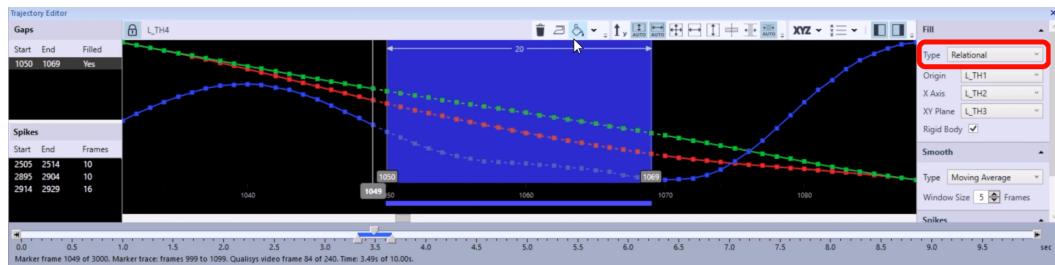


Figure 4.5. Relational fill. Image credit: [26]

4.3.4 Interpolation Choice and Usage Criteria

The approach I used for gap-filling exclusively employs the three aforementioned interpolation methods. To choose the interpolation method, I start with linear

fill, which is typically used for short gaps, and qualitatively assess the result by observing the animation and the trajectory path. If the marker does not follow the expected trajectory, I then move on to polynomial interpolation. For very long gaps, polynomial fill often generates unacceptable trajectories because of overfitting, so I fall back on relational interpolation. In this case, I associate the marker to be labeled with a nearby labeled marker that performs a similar movement, such as the ankle and knee.

4.4 Data Exportation

For further processing, it is necessary to move the data from Qualisys Track Manager, which allows exporting time series in various formats, including .txt, .csv, and .mat. The choice of format depends on how gaps are interpreted and converted. While .csv represents gaps as the origin in machine coordinates, leading to misrepresentations in later analysis, .mat assigns gaps as NaNs, making it the preferred format. All subsequent processing and analysis are implemented using Python.

4.5 Understanding 30-dimensional Data

An essential aspect of this project is the choice of the data structure shape to work with. The input data points correspond to 10 markers across 3 dimensions over a duration of approximately 10000 frames. As a result, the initial data structure has dimensions of (10, 3, 10000), which can be cumbersome to manage and visualize. It is assumed that the three coordinates X, Y, and Z are equally informative, with no dominant dimension. To make the structure more suitable for analysis methods such as PCA and RQA, it is reshaped to (30, 10000). This explains why we refer to the data as 30-dimensional. The 30-dimensional position trajectories from motion capture recordings are used as a proxy for whole-body muscle activity during treadmill locomotion. Although the movements of the body are highly complex and could theoretically require many dimensions to represent, the use of markers simplifies the problem, reducing it to 30 dimensions. By analyzing the relationships between the different markers, the dimensionality can be reduced even further. One key challenge is that the data may be non-linear and incomplete without a proper gap-filling, requiring a careful choice of dimensionality reduction technique. Gap-filling remains a significant task, as gaps result in a loss of information. Several dimensionality reduction methods exist, including non-linear techniques like UMAP and auto-encoders, as well as linear methods such as Principal Component Analysis and Linear Discriminant Analysis. Furthermore, the 30-dimensional data corresponds to the configuration of the markers, which in turn corresponds to the mean body configuration of the mouse.

4.6 Egocentric Coordinates Transform

To highlight how different parts of the body move relatively to each other, the trajectories were converted from the machine's coordinate system to an egocentric one (Figure 4.6). In this new system, the origin is set at the midpoint between the two hip markers. The x_e axis is defined by the vector connecting these hip markers, while the y_e axis is established as perpendicular to x_e , using the markers as reference points. The z_e axis, in turn, is perpendicular to the plane formed by both x_e and y_e , with its negative direction determined by the ankle markers. This transformation removes the effects of body translation and rotation.

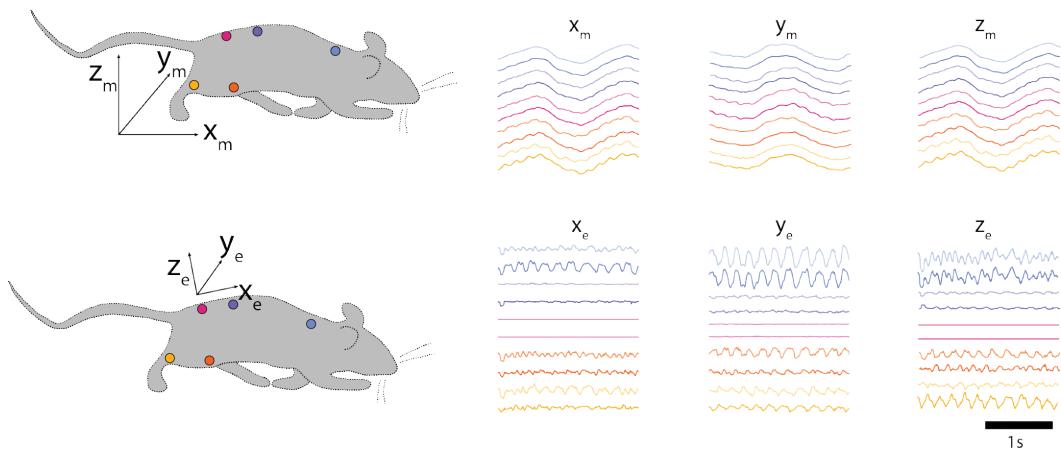


Figure 4.6. Comparison between machine's coordinate system and egocentric coordinate system. Image credit: LS

Chapter 5

Principal Component Analysis

This research aims to study the parts of the mouse's body involved in a detected gait adopted during movement, which can initially be decomposed using Principal Component Analysis. We chose PCA because it is simple, transparent, and effective at capturing the key patterns in the data without discarding important variability or non-linearities. Each experiment involved mice running at different speeds. We applied PCA to the pooled treadmill data within each experiment, keeping the data from different experiments separate. The goal of using PCA is to identify the different body configurations the mouse adopts while running at various speeds. Pooling the data helps capture the overall structure of these configurations, even if there are significant differences between speeds. Although some may argue that pooling could skew the results, the purpose here is to obtain a lower-dimensional representation that explains the main variations in the data. PCA achieves this by diagonalizing the covariance matrix of the pooled data, providing a general framework to compare movements across different speeds.

5.1 Data Pre-processing for PCA

Some of the typical PCA pre-processing steps were applied to the pooled 30-dimensional data, such as:

Mean Subtraction Subtracting the mean ensures that PCA does not treat the mean as the dominant feature, as without centering, the first principal component would simply be the vector that centers the data.

Dataset Shuffle Shuffling the dataset is important because it helps to prevent any unintended patterns or correlations that may exist due to the order of the data. If the data points are ordered by time, category, or some other factor, PCA might pick up on these patterns instead of focusing on the actual variance in the data.

Parallel Analysis Parallel Analysis involves generating a surrogate dataset composed of uncorrelated random variables that match the size of the original dataset. This process involves conducting PCA on several versions of these

surrogate datasets and calculating the average eigenvalues from these analyses. The resulting mean values serve as lower bounds for determining which Principal Components (PCs) are statistically significant [27].

Another common practice, though not used in this project, is standardization, which involves scaling the variance of the variables to 1. This is typically done when the data comes from different sensors or is measured in different units, to make the variables comparable. However, since the data in this case comes from the same source and is measured in the same units, we did not standardize it. Additionally, standardization would have caused the loss of relative variations across different parts of the body.

5.2 Evaluation of the Principal Components

Principal Component Analysis was performed on all acquisitions (trials) belonging to all experiments, considering only the tasks at $20\frac{m}{min}$.

5.2.1 Heuristics

Parallel Analysis was used to determine the threshold, and shuffled data were utilized to simulate noise, resulting in a noisy dataset on which PCA was initially performed. As the variance of these Principal Components converges around a similar value, an average threshold was computed based on 1,000 iterations; this number was chosen due to limitations in the random number generator (which is not purely random). The average variance of the PCs coming from the noisy dataset is 3.33%. The threshold is set at 3.33%, meaning that the first six principal components are considered significant, while the subsequent components likely represent noise, as shown in Figure 5.1.

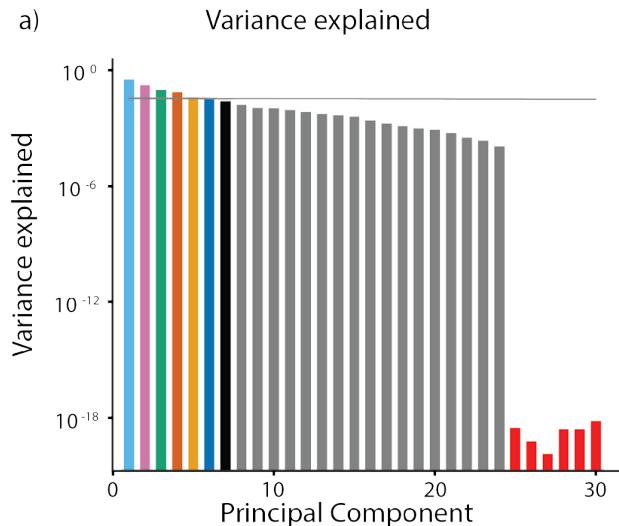


Figure 5.1. Explained Variance Ratio of Principal Components in logarithmic scale. Image credit: LS

From Figure 5.2, we can observe that applying PCA on each mouse reveals different contributions to the principal components and that a highly significant PC does not necessarily correspond to the largest contribution for a specific movement. Consequently, the different ordering of the PCs may reflect different types of running.

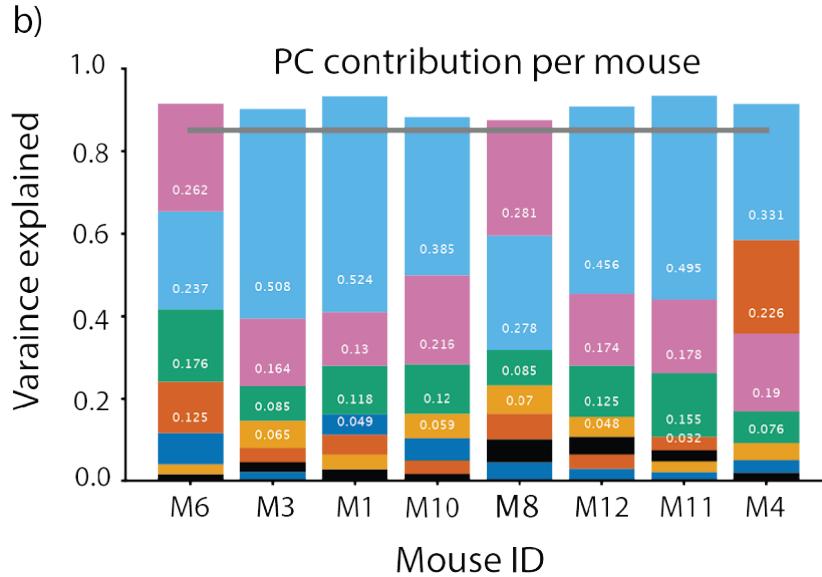


Figure 5.2. Variance Contribution of the first 6 PCs for each mouse. Every color is associated with a PC of the same color from Figure 5.1 while numbers inside bars indicate the variance of the corresponding PC. Image credit: LS

The graph in Figure 5.3 shows the number of principal components that, when summed starting from the most dominant from Figure 5.2, reach an 85% variance for each mouse. This threshold was obtained by summing the variances of the first six PCs from the complete mice dataset. This indicates that each mouse has a different minimum number of PCs that describe a distinct gait.

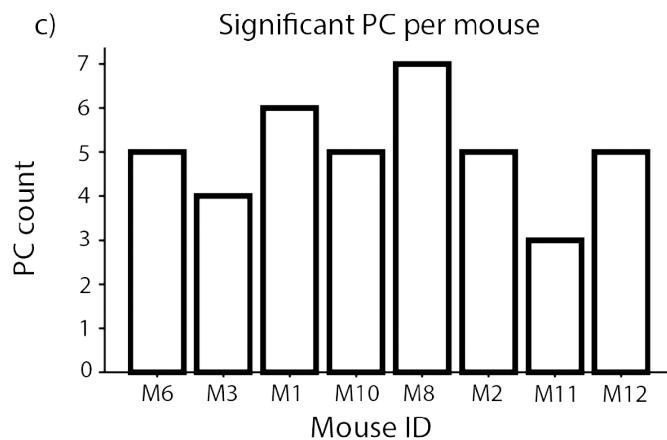


Figure 5.3. Number of PCs necessary to reach 85% variance threshold from the most dominant for each mouse. Image credit: LS

5.2.2 Modes of Deformation

The Principal Components computed from the data can be viewed as Modes of Deformation (MDs) around the mean body configuration of the mouse [14]. Although some studies interpret these principal components as muscle synergies [28], this thesis will refer to them simply as synergies [29]. The reconstruction is achieved by taking the mean body configuration and adding the contribution of each Principal Component (PC), where each PC's contribution is determined by multiplying its corresponding projection by its weight at the given time point. This process allows for an accurate representation of the original movement using only the most significant principal components. When applying deformation using a single Principal Component, the resulting trajectory tends to be flat or linear. This is because the deformation reflects only the variations along that specific direction in the data space. Consequently, using just one PC loses the complexity of the original movement, which can only be captured by considering multiple PCs simultaneously. The six most significant MDs, as shown in Figure 5.4, illustrate the mouse's leg alternation and hopping patterns. Even though we only tracked the hind limbs, the tracking of shoulder markers provides some insight into the movement of the front limbs. This allows us to compare the obtained MDs with traditionally defined gaits. PC1, PC3, PC5 are modes characterised by alternation of the hindlimbs, and PC2, PC4, PC6 are characterised by in phase coordination of the hindlimbs. In particular PC1 corresponds to the traditionally defined gait of trot in mice and PC2 to the bound gait.

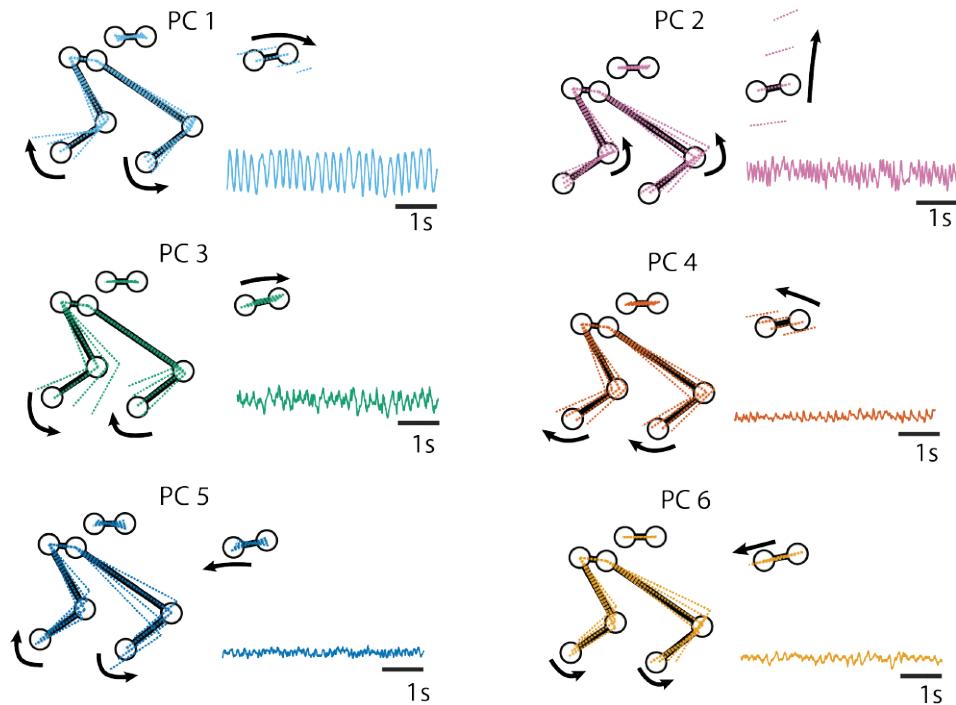


Figure 5.4. Static reconstructions of the different MDs picked out by PCA, along with the projection of the 30-dimensional trajectory along each PC axes is shown. Image credit: LS

Chapter 6

Recurrence Quantification Analysis

Alongside PCA, we used a method based on Recurrence Quantification Analysis to identify periodicities and uncover hidden structures, aiming to find coordination patterns in the processed time series. A primary objective of utilizing RQA is to detect cycles from 30-dimensional data, specifically identifying when the mean body configuration changes from one state to the same state over time during locomotion.

6.1 Recurrence Plot

The first fundamental step of RQA involves calculating a distance matrix, referred to as a recurrence plot. For example, by taking a sinusoidal signal, we can select a point and observe when it returns to a neighborhood around that point (Figure 6.1).

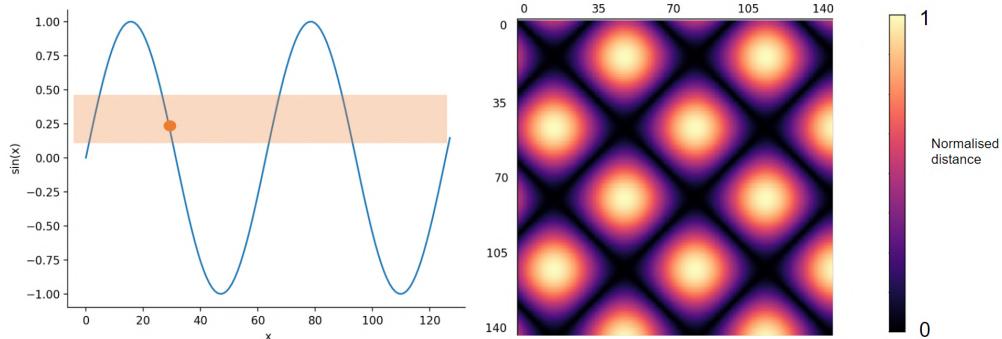


Figure 6.1. 2D sinusoidal signal highlighting a colored neighborhood where the distance from the orange point is close to zero, indicating its periodicity. Image credit: LS

Figure 6.2. Recurrence Plot computed using the Euclidean distance between the signal in Figure 6.1 and itself. The parallel pattern indicates the periodicity of the points. Image credit: LS

Since working with 30-dimensional data is computationally demanding and Euclidean distance is susceptible to outliers, dimensionality reduction to 3 dimensions was applied using UMAP [30]. The recurrence plot was then constructed by computing the Euclidean distance, specifically the L^2 norm, of the 3D time series with itself:

$$\mathcal{D}_{i,j} = \|\vec{a}_i - \vec{a}_j\|_2 = \sqrt{(x_{a_j} - x_{a_i})^2 + (y_{a_j} - y_{a_i})^2 + (z_{a_j} - z_{a_i})^2} \quad (6.1)$$

where:

- $\mathcal{D}_{i,j}$ is the value at i -th row and j -th column of the distance matrix \mathcal{D} ;
- \vec{a}_i and \vec{a}_j are respectively the i -th and j -th 3D points of the vector \vec{a} .

Despite the time series being approximated as a non-periodic signal, the recurrence plot reveals periodicities that appear as parallel patterns and clusters (Figure 6.3).

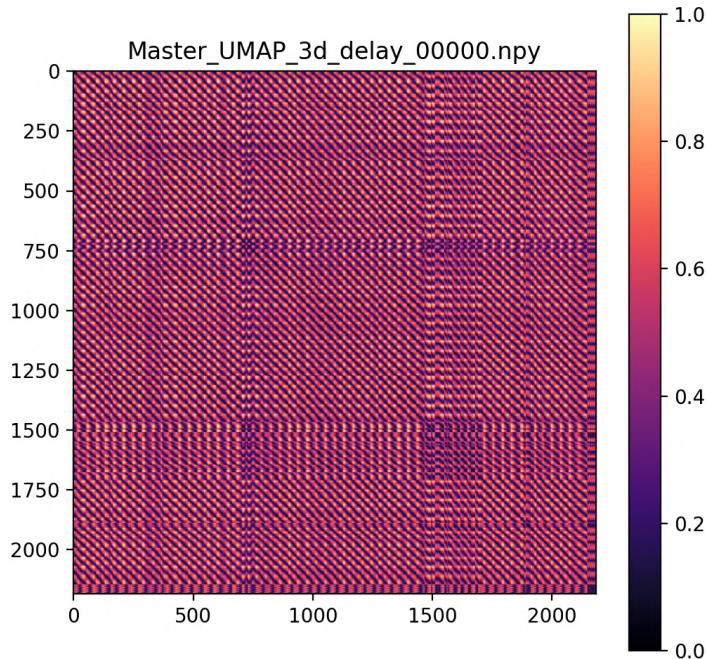


Figure 6.3. Recurrence Plot of pre-processed time series 3D UMAP-reduced from a $40 \frac{m}{min}$ trial. Image credit: LS

6.2 Dimensionality Reduction

There are few yet key reasons why it is necessary to reduce the dimensionality of the data in this project:

Curse of Dimensionality As the number of dimensions increases, the volume of the data space grows exponentially, making data analysis more difficult and leading to sparsity of data, which can degrade the performance of algorithms [31].

Computational Feasibility High-dimensional data often requires significant computational resources. Reducing the dimensionality helps decrease the time and memory needed for computations, making the analysis more practical.

Visualization It is difficult to visualize and interpret data in more than three dimensions. By reducing the data to lower dimensions, such as 2D or 3D, it becomes easier to visualize patterns, trends, and relationships within the data.

In this project, UMAP was used to reduce the dimensionality of the 30-dimensional time series to 3 dimensions, in order to ease the computation of recurrence plots and to visualize the data points in 3D (Figure 6.4). PCA was used only to decompose the movement before RQA and to compare it with the RQA output. ISOMAP was used as a comparative dimensionality reduction algorithm but was discarded in favor of UMAP due to UMAP's clustering behavior.

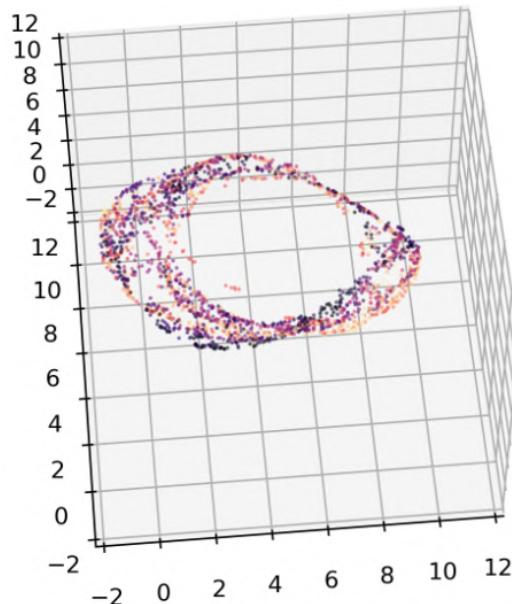


Figure 6.4. Plot of the 30-dimensional time series reduced to three dimensions using UMAP. The values of the axes represent the 3D coordinates of the points in the time series. Image credit: LS

6.2.1 UMAP

Uniform Manifold Approximation and Projection is a dimension reduction technique based on Riemannian geometry and algebraic topology. It is fast, scalable, and preserves more global structure compared to t-SNE, making it ideal for visualizing data. Unlike other methods, UMAP has no limitations on the number of dimensions, making it useful for general machine learning tasks [30].

6.2.2 ISOMAP

Isometric Mapping is a nonlinear dimensionality reduction technique introduced by Tenenbaum et al. in 2000. It uncovers low-dimensional structures in high-dimensional

data by preserving geodesic distances between points. ISOMAP constructs a neighborhood graph based on local distances and then applies classical multidimensional scaling to obtain a low-dimensional embedding. Unlike traditional methods like PCA, ISOMAP effectively captures nonlinear relationships and converges to the true data structure for certain manifolds [32].

6.3 Cycle Detection Methods

In the literature, various algorithms for cycle detection can be found, with frequency analysis being one of the most popular. However, in this project, a different method is proposed to identify groups of cycles, which are introduced for the first time as "Unitary Movements".

6.3.1 Frequency Analysis

Frequency analysis using Fourier analysis is a powerful technique for identifying cycles in time series data. In particular one method entails focusing on finding peak frequencies with the application of a threshold:

Fourier Transform The Fourier Transform converts a time-domain signal, such as a time series, into its frequency-domain representation. This transformation decomposes the signal into its constituent frequencies, allowing us to analyze how much of each frequency is present in the original signal.

Identifying Peaks After applying the Fourier Transform, the output is a spectrum that shows the amplitude of various frequencies. Peaks in this spectrum correspond to dominant frequencies in the original signal, which are indicators of periodic behavior or cycles.

Applying a Threshold To identify significant peaks, a threshold can be applied. This threshold helps in distinguishing meaningful frequencies from noise. Frequencies with amplitudes above this threshold are considered indicative of cycles in the data.

Even though this method can detect a full stride of a mouse (Figure 6.5), it has a limitation: if some steps do not meet the threshold, such as gait changes, they will be ignored and lost. Another limitation concerns what one intends to observe in the mouse's locomotion. This research aims to study the entire body movement and the interrelation of its discrete parts, which cannot be analyzed because the peak frequency method operates only through separate computations, such as analyzing one limb at a time.

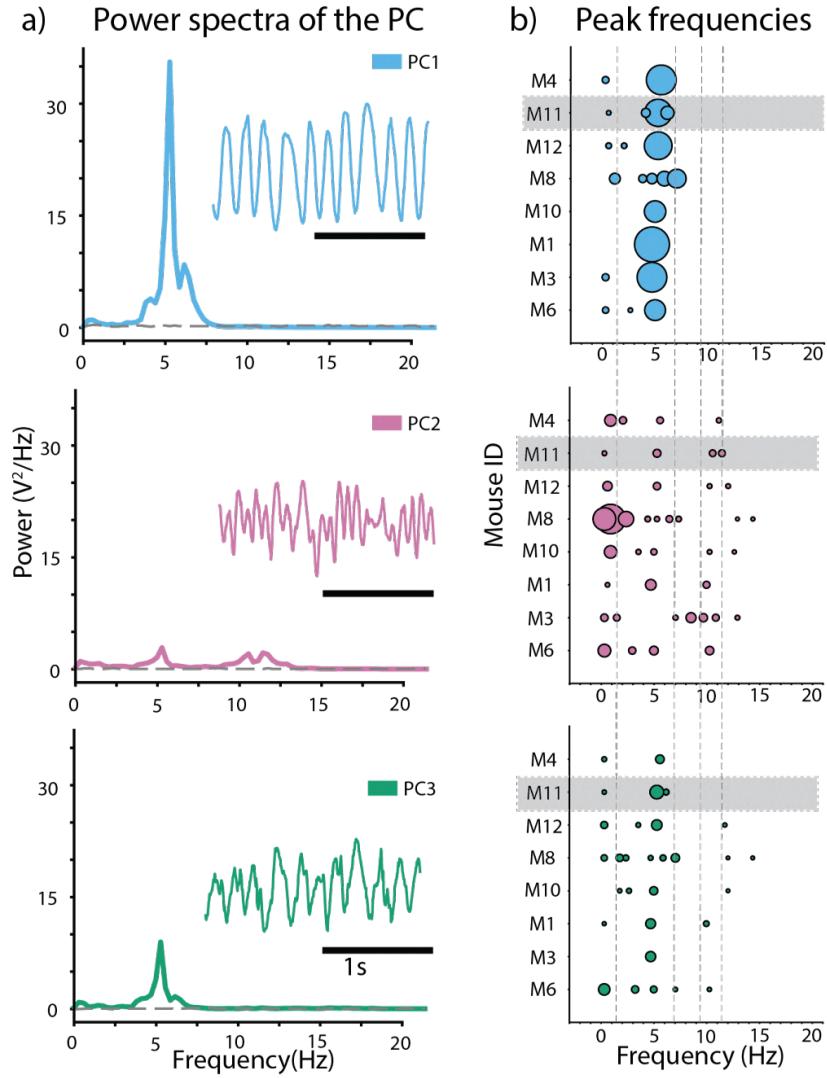


Figure 6.5. Peak frequencies computed from the first three principal components for each mouse. Image credit: LS

6.3.2 Unitary Movements

The study of four-limb coordination in a moving mouse remains challenging, even in recent literature [33]. A muscle movement, even when repeated with extreme precision, does not follow the exact same trajectory in each iteration, as demonstrated in the famous study by Nikolai Bernstein. His research demonstrated that most actions, like striking a chisel with a hammer, are composed of smaller movements. A change in any of these smaller components impacts the overall motion [34]. The limited tolerance to these variations by the frequency analysis makes it an unsuitable method for this project.

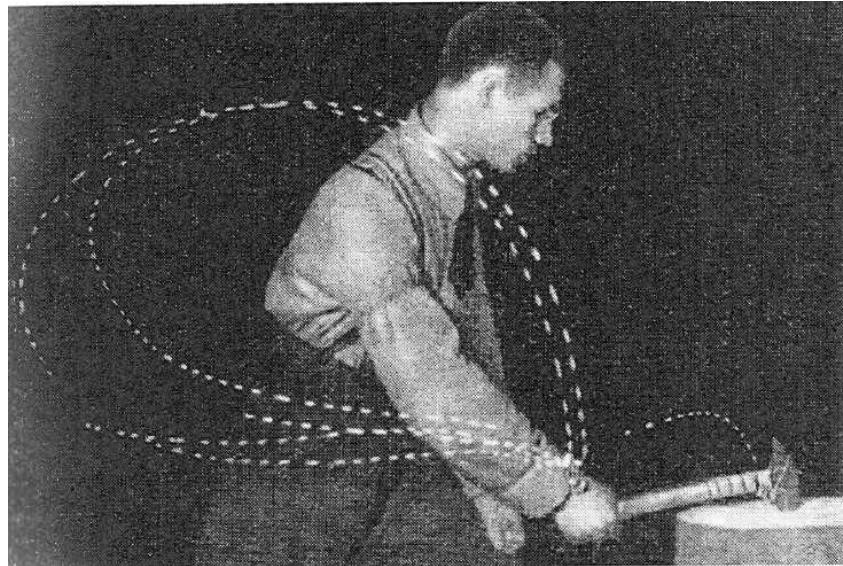


Figure 6.6. A cyclogram of cutting metal with a chisel and hammer. Aleksei Gastev in the laboratory of the Central Institute of Labor.

It can be observed in the Figure 6.6 that the movement is not defined by a single trajectory, but rather by a set of cycles. This comes from the hypothesis that these cycles are all outputs of the same command from the central nervous system. From this, we have defined these groups of cycles, in the context of locomotion, as "Unitary Movements". This term was chosen because these cycles represent coordination patterns (movements) and are repeated with each unit of locomotion (unitary), such as a step cycle.

6.4 Delay Embedding

The technique of delay embedding is employed to increase the dimensionality of a time series, facilitating the reconstruction of the dynamics of complex systems. This method involves creating a vector that combines measurements of a signal at regular time intervals, known as delays (Figure 6.7). By transforming a uni-dimensional time series into a multidimensional space, delay embedding allows for the identification of hidden structures and dynamics within the system's behavior [35]. This enhanced representation enables researchers to uncover coordination patterns, cyclicity, and other complex phenomena that may not be apparent in a one-dimensional view.

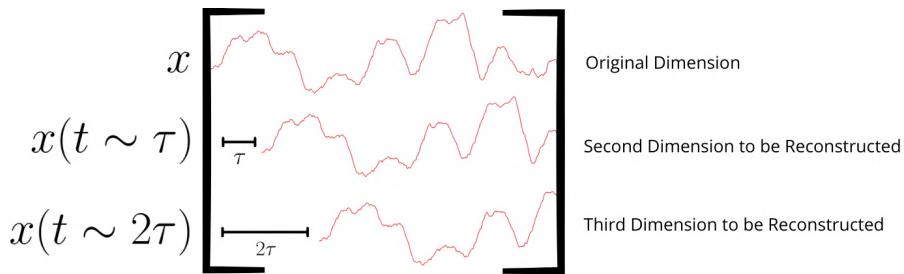


Figure 6.7. Example of reconstruction on two delay embeddings from a 1-dimensional time series. It was applied a 2τ delay. Image credit: FB

The reconstruction of other dimensions consists of duplicating the original time series n times, in its original dimension, each shifted by τ such that:

$$\tau = 1 \text{ frame} = \frac{1}{300} \text{ ms} \quad (6.2)$$

For example, as shown in the Figure 6.8, applying a $167 \text{ ms} = 50\tau$ delay results in reconstructing $n = 50$ delay embeddings, thereby increasing the dimensionality to $30 \times 50 = 150$ dimensions.

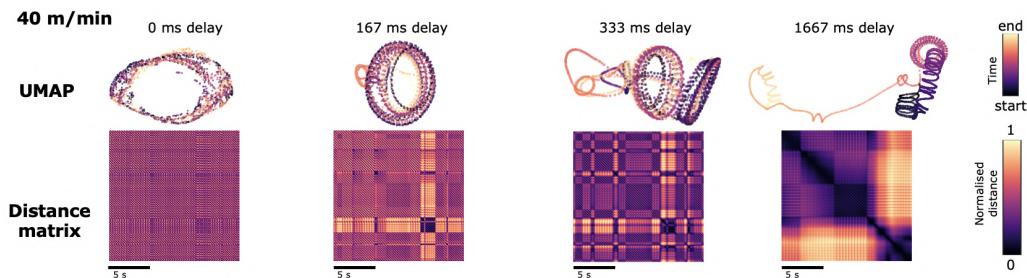


Figure 6.8. Recurrence plots and 3D UMAP-reduced visualization of the same time series at different delays. Image credit: LS

As expected, the delay embedding procedure reveals the untangled hidden structures of the dynamics by unfolding the underlying attractor. An attractor in delay embedding is defined as a set of points in phase space that represents the behavior of a dynamic system. This technique reconstructs phase space from time series data. However, it is important to note that applying too many delay embeddings can lead to noise reconstruction [36], as shown in Figure 6.8 by applying a 1667 ms delay. Furthermore, the recurrence plots gradually appear blurrier, but with more pronounced clusters. In the case of a 333 ms delay, groups of cycles, specifically the unitary movements, become clearly visible. Finally, the recurrence plot allows for the identification of the exact frames that delineate the unitary movements, as shown in the Figure 6.9.

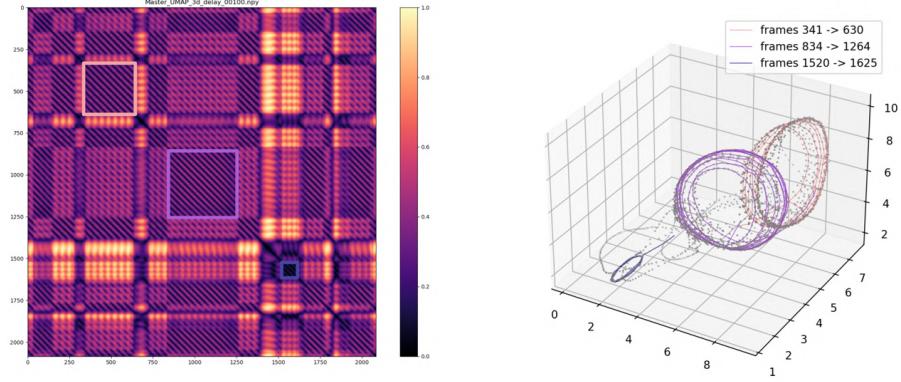


Figure 6.9. Plots from a correct delay embeddedding with 333 ms delay. On the left, a Recurrence Plot highlighting well-defined clusters formed by the frames along the X and Y axes. On the right, segmented Unitary Movements from the time series are shown for specific frame ranges corresponding to the clusters in the Recurrence Plot. Dimensionality reduced with UMAP. Image credit: LS

It is also evident the difference between UMAP and ISOMAP, with UMAP applying clustering techniques and ISOMAP maintaining the exact distances between points when reducing dimensionality. From the Figure 6.10, it can be concluded that ISOMAP is not a suitable method for this task, as it does not capture information about the clusters.

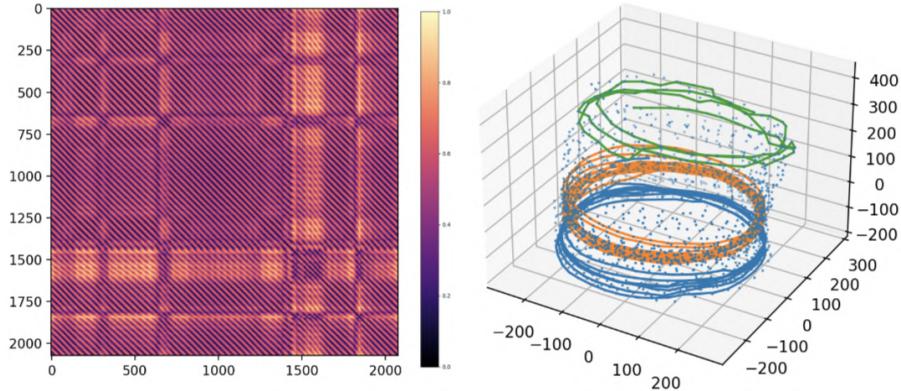


Figure 6.10. Recurrence plot and 3D visualization of the same time series. Dimensionality reduced with ISOMAP. Image credit: LS

6.5 Threshold Criteria for Delay Embedding

To avoid overembedding, a stopping method is required to identify the optimal number of delay embeddings. We propose two approaches: a computation-based method aimed at detecting overembedding, and vision-based approach, proposed and implemented by me, that aims to provide a metric for determining a threshold compatible with any time series.

6.5.1 Recurrence Plot Points Correlation

This method involves observing a row of the distance matrix as a 2D signal (Figure 6.11), selecting two points that are temporally distant, and calculating their Euclidean distance.

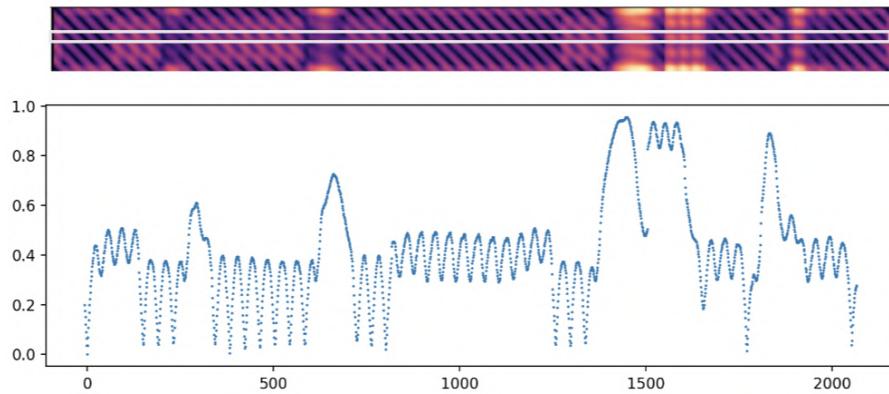


Figure 6.11. 2D signal coming from a row of a correctly delay embedded time series. Image credit: LS

It is assumed that two temporally distant points are uncorrelated; however, in the case of overembedding, they appear to be spatially close over a certain threshold. Furthermore, it is noted that the periodic behavior is lost due to overembedding, as shown in the Figure 6.13.

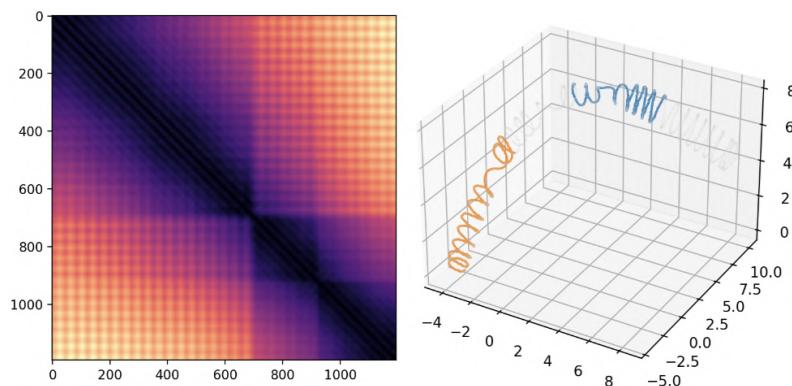


Figure 6.12. Recurrence plot and 3D UMAP-reduced visualization of the same overembedded time series. Image credit: LS

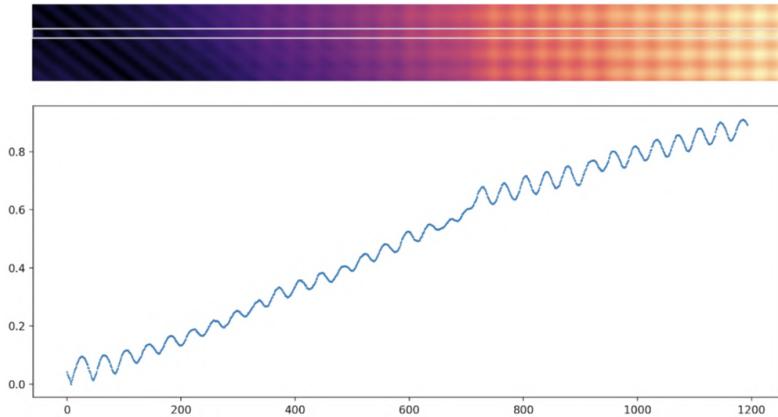


Figure 6.13. 2D signal coming from a row of a overembedded recurrence plot. Image credit: LS

6.5.2 Quantification using Blur Detection

Starting from the intuition that as the number of delay embeddings increases, the blur of the recurrence plot also increases, I experimentally observed that this intuition holds true. By calculating a metric called the focus measure, which quantifies the blur of an image and consequently the unfolding of the underlying attractor, I found that it exhibits a decreasing trend as the delay increases. The focus measure is obtained by applying the Laplacian operator and then calculating its variance. The Laplacian is indeed an operator commonly used for edge detection, as it involves calculating the second partial derivatives of an image, thereby identifying inflection points that correspond to the edges of shapes within the images. If a large number of edges is detected, it indicates that the image is sharp, resulting in a high variance. For a two-dimensional image $I(x, y)$, the Laplacian is defined as:

$$\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (6.3)$$

where:

- ∇^2 is the Laplacian operator;
- $I(x, y)$ is the intensity of the pixel at coordinates (x, y) ;
- $\frac{\partial^2 I}{\partial x^2}$ is the second partial derivative of the image in the horizontal direction;
- $\frac{\partial^2 I}{\partial y^2}$ is the second partial derivative of the image in the vertical direction.

For an image with N pixel values p_1, p_2, \dots, p_N , the variance σ^2 can be calculated as follows:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (p_i - \mu)^2 \quad (6.4)$$

where μ is the mean pixel value of the image.

I computed the variance of the Laplacian, referred to as the focus measure, of a recurrence plot using the OpenCV library in Python. I converted the image to grayscale and computed the Laplacian by convolving it with the following 3×3 kernel:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (6.5)$$

The convolution of an image I with a kernel K is given by:

$$(I * K)(x, y) = \sum_{m=-M}^M \sum_{n=-N}^N I(x + m, y + n) \cdot K(m, n) \quad (6.6)$$

where:

- $(I * K)(x, y)$ is the convolved value at (x, y) ;
- $I(x + m, y + n)$ is the value of the pixel in the original image;
- $K(m, n)$ is the value of the kernel;
- M, N are the dimensions of the kernel.

Subsequently, I took the focus measure as the variance of the response from the convolution. Finally, a threshold based on the focus measure can be empirically determined, with a value around 53, as shown in the Figure 6.14. It still needs to be verified whether the same focus measure can be fully generalized for every time series.

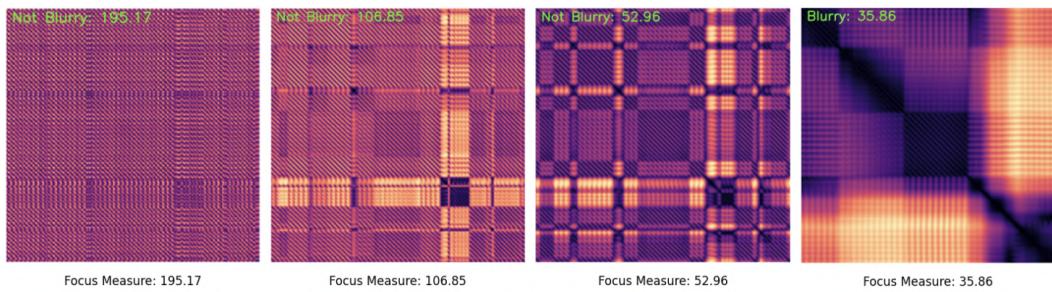


Figure 6.14. Blur detection applied to different recurrence plots with increasing number of delay embeddings. Image credit: FB

It is crucial to highlight that the focus measure differs from the number of delay embeddings, as the latter may yield a different unfolding depending on the input time series. In fact, different delays can be associated with the same focus measure. Blur detection is also more robust than the aforementioned points correlation method, which lacks a clear threshold definition to determine whether a point is spatially distant.

Chapter 7

Web Application for Data Visualization

A further contribution of mine was the design and implementation of a full-stack web application for the 2D and 3D visualization of data captured through motion capture and the modes of deformation obtained from PCA. The data collected and processed are derived from an abstraction of the mouse's body (marker configuration), and due to the complexity of their structure, they are not easily interpretable. This web application, called Mouse Movement, is developed to meet the need to visualize the collected data for a qualitative analysis of the mouse's movement. This application was primarily built as a qualitative tool complementary to the project's quantitative analysis, thus forming an integral part of it.

7.1 Aim of Mouse Movement

The goal of Mouse Movement is to serve as a tool for accurately visualizing the mean body configuration in order to study the discrete parts during overall motion. Additionally, it serves as a comparative tool between the animations and time series derived from the egocentric coordinate transform and the first six principal components. Mouse Movement exists also to meet the need for a customized tool designed for the specific type of data collected using Qualisys Track Manager, in support of the research conducted by the Neuronal Rhythms in Movement unit.

7.2 Software Design and Use Cases

A key element of the application's functionality is the time series contained in the .mat file, which is the output from exporting the animation from Qualisys Track Manager. The application is divided into two sections: Dashboard and Tables, both accessible via a sidebar menu. The application's main page is located in the "Dashboard" section, where 3D and 2D visualizations of animations are displayed. In the "Tables" section, files uploaded to the storage can be loaded and viewed along with their details (Figure 7.1). Therefore, users must upload at least one file (Figure 7.2), after which they can view the acquisition details or remove it (Figure 7.3) from the Tables section. From the Dashboard section, users can select

one of the uploaded files, which will display the original time series with the applied egocentric transform and one of the six modes of deformation (Figure 7.4).

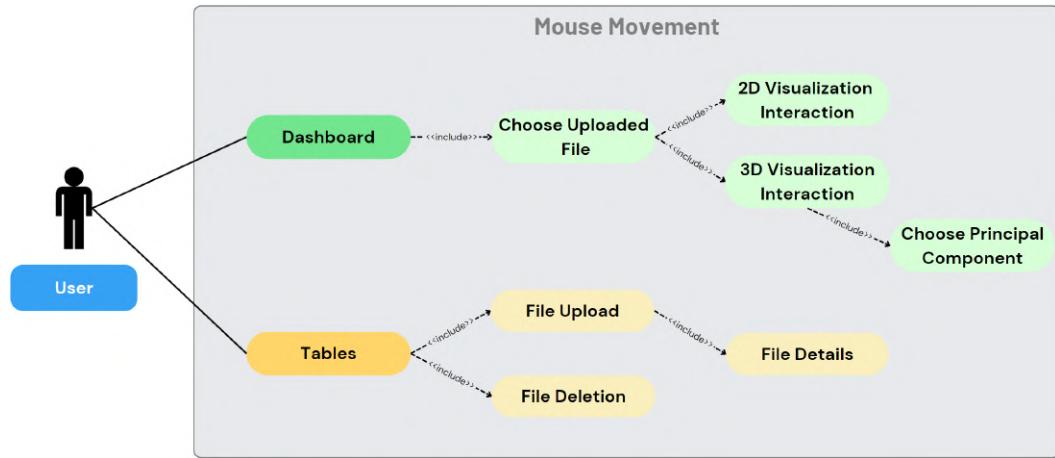


Figure 7.1. UML Use Case Diagram showing the User interaction with the two sections "Dashboard" and "Tables". Image credit: FB

3D Visualization The first type of visualization is in 3D, showing an interactive scene where the marker configuration is plotted, allowing users to navigate the environment by rotating, zooming, and panning the camera. The camera's position and orientation can be reset using specific buttons that align it with one of the three axes. Users can choose to start or stop the animation, which is calculated in real-time, and adjust playback settings, such as changing the current frame and the playback speed in frames per second. Among the display options, there are several toggles, including a dark grid mode for better contrast and visibility, the X, Y, and Z axes for reference to the mean body configuration's position, the complete trajectory path of the points, and bones connecting specific markers. Though the two animations are displayed side by side, each has its own settings and play buttons. However, users can activate synchronization mode to play both animations simultaneously at the same speed (Figure 7.9). In this case, playback settings are controlled solely from the egocentric transform animation side, while the navigation menu for the PCs is disabled. Users can still change the PCs at any time.

2D Visualization As for the 2D visualization, users can zoom and pan through a graph that displays the time series of the markers for each coordinate. These are also shown side by side, comparing the egocentric transform with the MDs. Additionally, users can click on a point in the time series to set the corresponding frame in the 3D view (Figure 7.7).

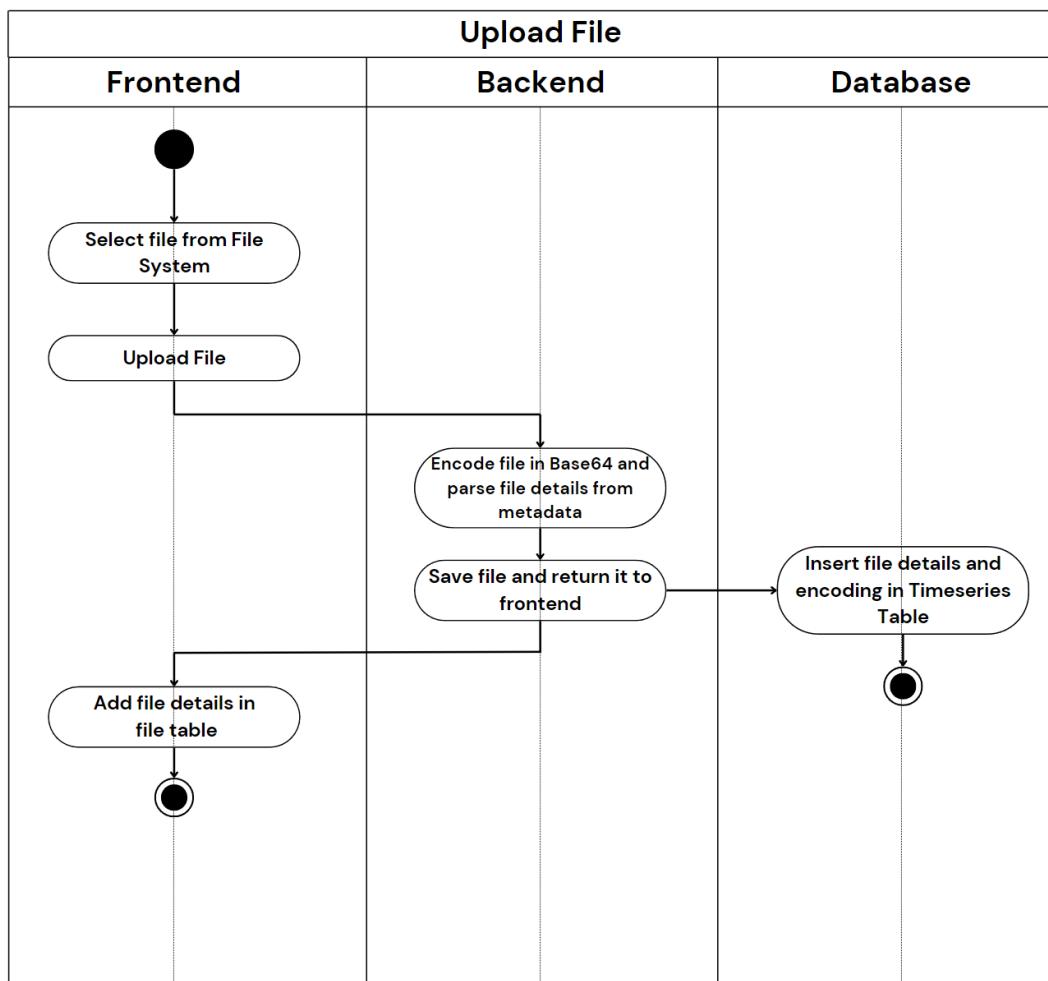


Figure 7.2. Activity Diagram showing the file upload process. Image credit: FB

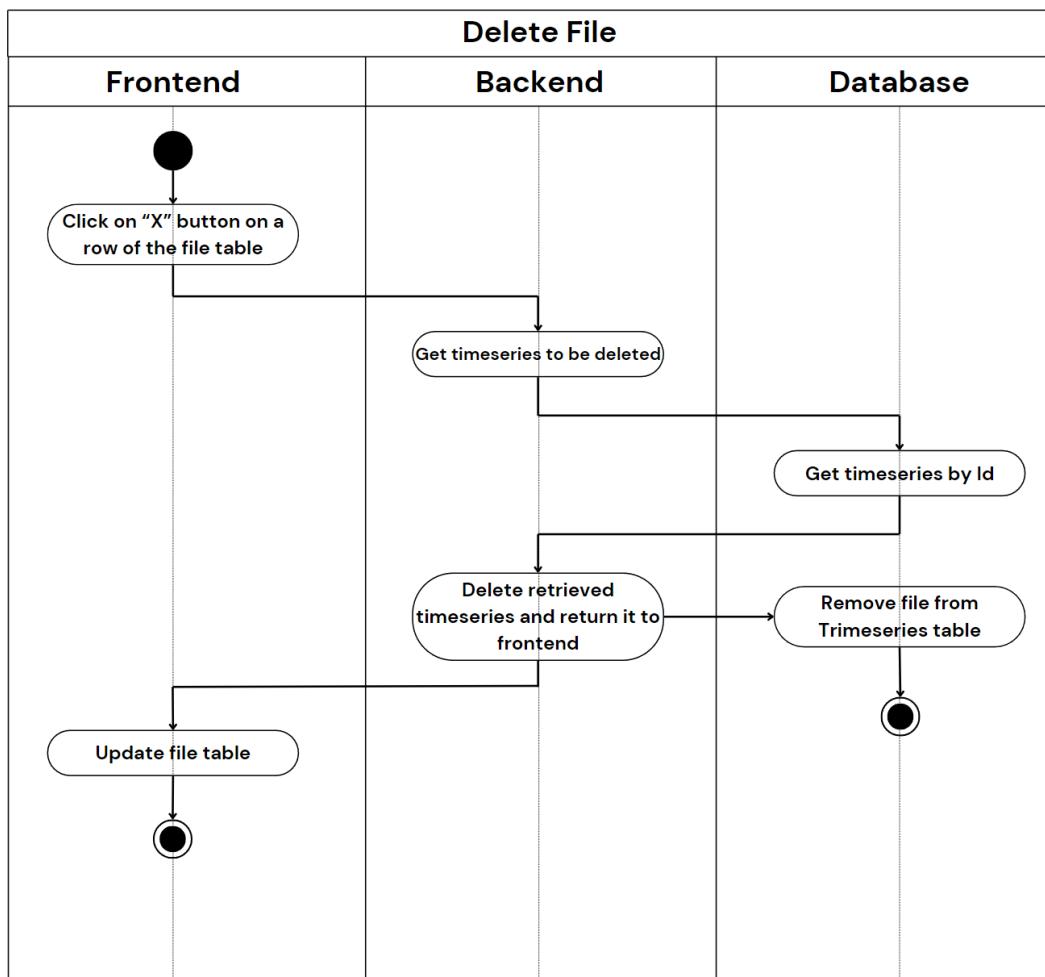


Figure 7.3. Activity Diagram showing the file deletion process. Image credit: FB

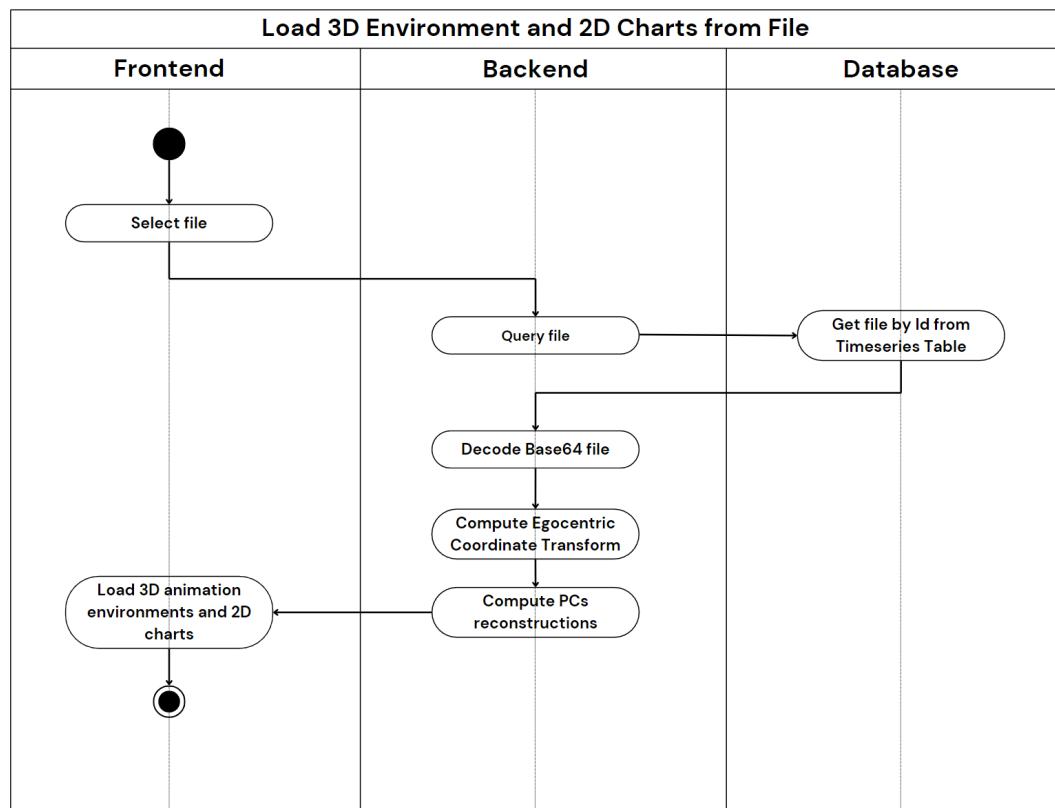


Figure 7.4. Activity Diagram showing the loading process of 3D Environment and 2D Charts from a file. Image credit: FB

7.3 Software Architecture

For Mouse Movement, I chose a client-server architecture with relational database support (Figure 7.5).

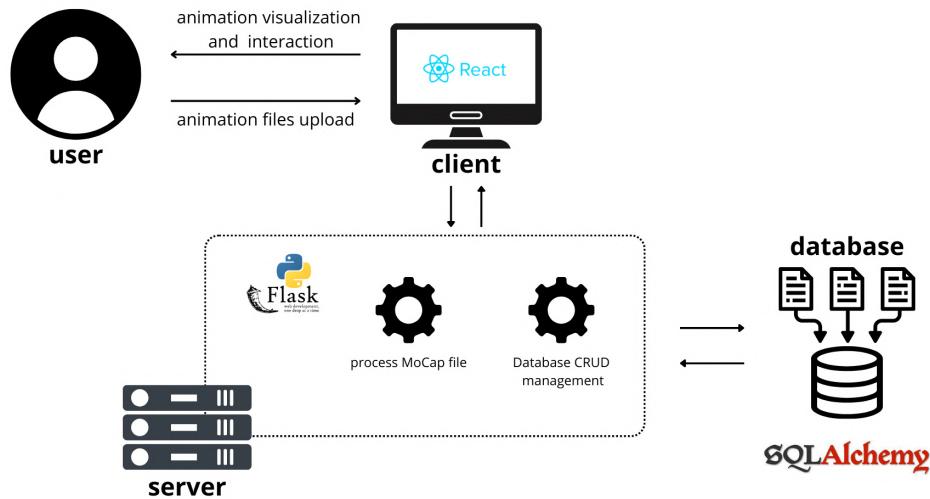


Figure 7.5. Mouse Movement Software Architecture. Image credit: FB

Client The client is developed using the ReactJs framework, a popular JavaScript library for building user interfaces, particularly suited for creating dynamic and responsive web applications [37]. Through ReactJs, users interact with the file management interface and visualization tools.

Server The front-end communicates with the back-end, which is developed using Flask, a lightweight Python web framework designed for building scalable web applications [38]. Flask allows the definition of APIs for the application and retrieves information from the database.

Database The database is managed using SQLAlchemy, a Python-based Object-Relational Mapping (ORM) library that simplifies database interactions [39]. It provides a simple, lightweight, and portable solution for storing uploaded files, saving time during each use. This is used to store information about the uploaded file in a table named "Timeseries" (Figure 7.6).

Timeseries	
Id:	Integer
Experiment:	String
Sample:	String
Cage:	String
Mouse:	String
Task:	String
Task_Detail:	String
Acquisition:	String
Processed:	String
Data:	String

Figure 7.6. Database "Timeseries" Table. Image credit: FB

7.4 Implementation

The graphical design of the ReactJs front-end is implemented using MaterialUI, a popular React component library that provides pre-built, customizable components following Google's Material Design guidelines [40]. It helps create a consistent, responsive, and visually appealing interface with minimal effort. Both the 3D (Figure 7.8) and 2D (Figure 7.10) visualizations are developed with ThreeJs, a powerful JavaScript library that simplifies the creation of complex 3D graphics and animations in the browser [41]. ThreeJs allows for the rendering of interactive 3D scenes directly in the web application, providing users with an immersive visualization experience. The Flask back-end is responsible for applying the egocentric transform, performing Principal Component Analysis using scikit-learn [42], and extracting the Modes of Deformation from the files. To run correctly Mouse Movement, the Python 3.12.4 and Node.js 22.6.0 versions are required.

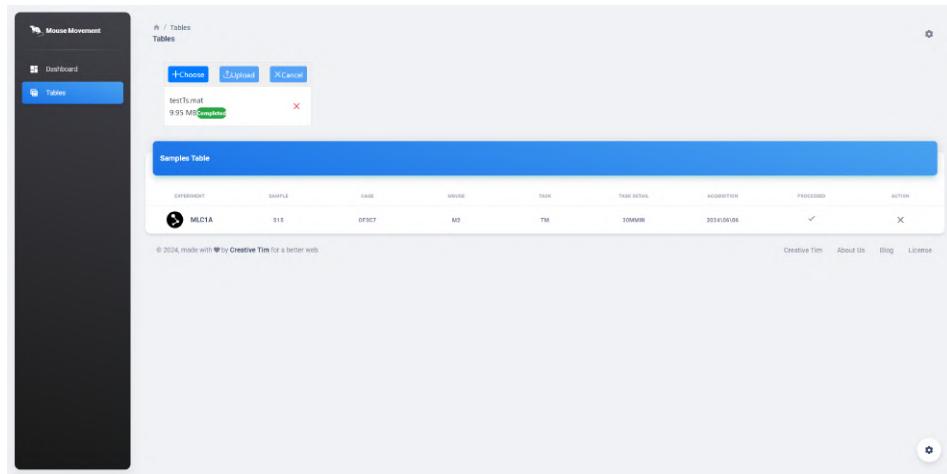


Figure 7.7. Time series file upload page "Tables". The experiment name and other details are extracted from the metadata of the .mat file, which is derived from the original QTM filename according to the naming convention discussed in Chapter 3.4 and are independent of the .mat name. Image credit: FB

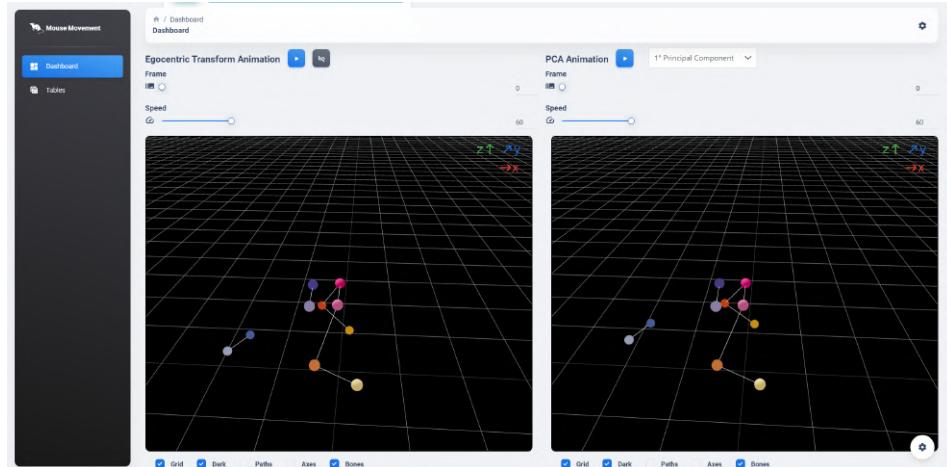


Figure 7.8. 3D environment displaying the original processed animation on the left side and the deformation mode resulting from the first principal component on the right side **not synchronized**. Image credit: FB

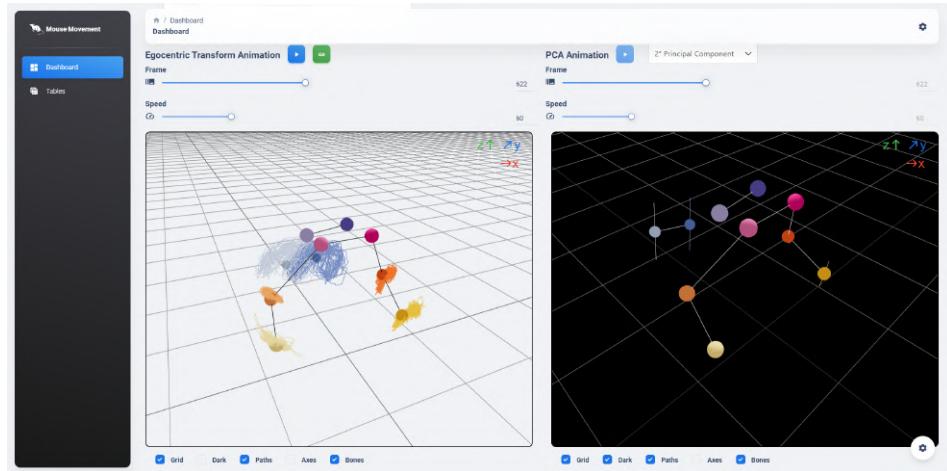


Figure 7.9. 3D environment displaying the original processed animation on the left side and the deformation mode resulting from the second principal component on the right side **synchronized**, showing the paths of the trajectories. Image credit: FB

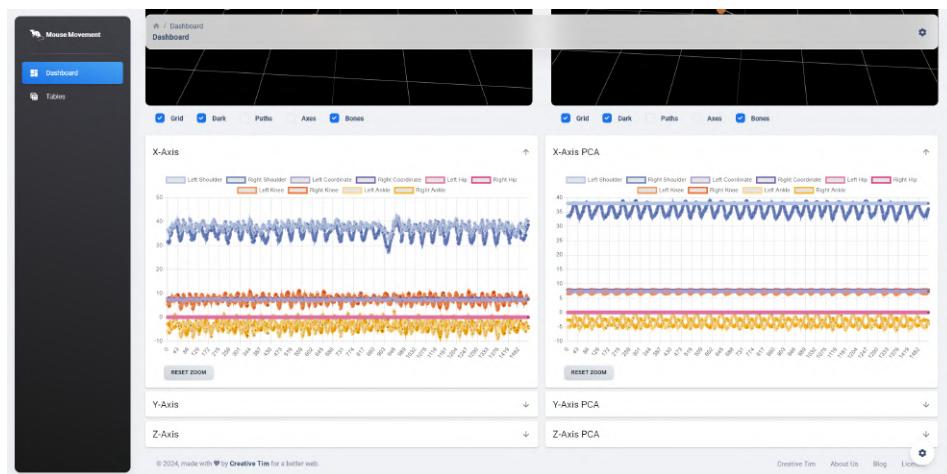


Figure 7.10. 2D graphs comparing the X-axis of the original processed time series with the X-axis of the principal component. Image credit: FB

7.5 Known Issues and Limitations

The front-end relies on local hardware, which can become burdened when visualizing time series exceeding 10,000 frames.

7.6 Possible Applications

This application is designed to accept any .mat file exported from QTM, making it compatible with other animals subjected to motion capture as well. Full portability to MacOs, Linux, and Windows operative systems is ensured through Node.js and Python. The application is intended for use by the entire research staff of the Neuronal Rhythms in Movement unit. Finally, the ability to compare time series with their MDs could also prove useful in other fields, such as medicine.

Chapter 8

Ethics and Animal Care

The entire animal handling process follows the highest international standards for animal care and use, as it is conducted with accreditation by Association for Assessment and Accreditation of Laboratory Animal Care International (AAALAC International) [43]. The animal study protocol was conducted in accordance with procedures approved by the Okinawa institute of Science and Technology (OIST) Institutional Animal Care and Use Committee (IACUC) (Protocol IDs: 2023-040) in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals (National Research Council, 2011) Every effort was made to minimize suffering [22].

Chapter 9

Conclusions

In this research, I played a pivotal role in several key aspects of the study. My contributions were essential in the data processing phase, where I handled critical tasks such as labeling and manually gap-filling motion capture data, ensuring that the dataset was robust and ready for subsequent analyses. Additionally, I contributed significantly to the Recurrence Quantification Analysis by proposing and implementing a vision-based method to enhance the accuracy of delay embedding parameter selection. This step was crucial for the segmentation and detection of Unitary Movements, which are central to the findings of this study. Furthermore, I developed a custom visualization tool to compare gaits, providing a more comprehensive understanding of the data. This research offers significant advancements in the study of animal locomotion, specifically in mice, by introducing a novel method for isolating and analyzing Unitary Movements. The approach provides a deeper understanding of movement variability and how different gaits are coordinated through the body.

9.1 Future remarks

Future work should focus on improving the recognition and classification of Unitary Movements. Better training techniques, possibly incorporating machine learning and advanced algorithms, could enhance the automatic detection of these movement patterns. Additionally, feature engineering can be explored to extract more meaningful and interpretable features from the data, potentially improving the accuracy and efficiency of gait analysis. Another area of progress involves extending the current methods to other animal models. Adapting the approach for use with different species would increase the utility and scope of the research, enabling broader applications in comparative locomotion studies and expanding its relevance across various fields of biological and medical research. Finally, this research holds promise for medical applications. The ability to analyze and compare movement patterns in a precise, data-driven manner could inform clinical studies related to motor control disorders, rehabilitation strategies, and gait abnormalities in both animals and humans. Future developments in this direction could offer valuable insights for medical interventions, contributing to advancements in therapeutic techniques for neurological and musculoskeletal conditions.

Abbreviations

Abbreviation	Full Form
2D	2-dimensional
3D	3-dimensional
AI	Artificial Intelligence
BIJ	Bogna Ignatowska-Jankowska
CNS	Central Nervous System
csv	Comma Separated Values
ECG	Electrocardiogram
FB	Federico Barreca
FPS	Frames per Second
ID	Identifier
IR	Infrared
ISOMAP	Isometric Mapping
LDA	Linear Discriminant Analysis
LS	Lakshmipriya Swaminathan
mat	Microsoft Access Table
MD	Mode of Deformation
MP	Mega Pixels
NaN	Not a Number
nRIM	Neuronal Rhythms in Movement
OIST	Okinawa Institute of Science and Technology
PC	Principal Component
PCA	Principal Component Analysis
QTM	Qualisys Track Manager
RGB	Red Green Blue
RQA	Recurrence Quantification Analysis
SLEAP	Social LEAP Estimates Animal Poses
txt	Text file
UMAP	Uniform Manifold Approximation and Projection
UV	Ultraviolet

Appendix A

9.1.1 Evaluation of Performance



Evaluation of performance

Federico Barreca - OIST internship 2024

Mr. Federico Barreca joined our laboratory (Neuronal Rhythms in Movement Unit) in OIST, Okinawa, Japan in July 2024, and he stayed until September of that year. He was tasked with working with one of our senior PhD students on a challenging project involving visualization of high-resolution, high-dimensional mouse motion capture data.

Federico completed his task admirably. Showing considerable independence, proactivity and ingenuity, he familiarized himself with the motion capture data framework and preprocessed recordings from a large experiment, giving him a solid understanding of the features in our data and informing him of the best ways to proceed with the visualization.

Specifically, the pre-processing involved manual labeling motion capture marker data, as well as filling any gaps occurring due to incomplete tracking.

During his internship, he created a fully-functional webapp tool that allows researchers to load and explore kinematic data in real-world and principal component dimensions. The tool will be used in the upcoming work of the PhD student's publication. Furthermore, he developed a method for machine vision-based definition of temporal delay for embedding.

In all, Federico has been one of the most productive and capable interns we have ever hosted. His attention to detail, reproducibility, robustness and maintainability of his code are exemplary. We have no doubt he will have a very successful future career.

A handwritten signature in black ink that reads "M. Uusisaari".

Marylka Yoe Uusisaari
October 7th, 2024
Associate Professor nRIM, OIST, Okinawa, Japan

Appendix B

9.1.2 OIST Research Internship Certificate of Completion



Certificate of Completion

修了証明書

This is to certify that the student below has completed the Research Internship Program at the Okinawa Institute of Science and Technology Graduate University.

下記の者は、沖縄科学技術大学院大学におけるリサーチ・インターンシッププログラムを修了したことを証明する。

Student ID 学籍番号:	2303115
Name 氏名:	Barreca, Federico バレカ フェデリコ
Research Unit 研究室:	Neuronal Rhythms in Movement Unit 神經活動リズムと運動遂行ユニット
Supervisor 指導教員:	Uusisaari, Marylka Yoc ウーシサーリ マリルカ ヨエ
Period 期間:	2024-07-18 to 2024-09-30

Thomas Busch
Dean of Graduate School of Science and Technology
Okinawa Institute of Science and Technology Graduate University

Seal of OIST
Graduate School



Bibliography

- [1] Patrick T. Sadtler, Kristin M. Quick, Matthew D. Golub, Steven M. Chase, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Byron M. Yu, and Aaron P. Batista. Neural constraints on learning. *Nature*, 2014.
- [2] Marcel Bélanger, Marc Bélanger, Trevor Drew, Trevor Drew, Janyne Provencher, Janyne Provencher, Serge Rossignol, and Serge Rossignol. A comparison of treadmill locomotion in adult cats before and after spinal transection. *Journal of Neurophysiology*, 1996.
- [3] Patrick J. Whelan and Patrick J. Whelan. Control of locomotion in the decerebrate cat. *Progress in Neurobiology*, 1996.
- [4] SF Giszter, FA Mussa-Ivaldi, and E Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.
- [5] Juan Álvaro Gallego, Juan Álvaro Gallego, Matthew G. Perich, Matthew G. Perich, Lee E. Miller, Lee E. Miller, Sara A. Solla, and Sara A. Solla. Neural manifolds for the control of movement. *Neuron*, 2017.
- [6] J. D. B. Stillman and Eadweard Muybridge. *The Horse in Motion as Shown by Instantaneous Photography, with a Study on Animal Mechanics Founded on Anatomy and the Revelations of the Camera, in Which Is Demonstrated the Theory of Quadrupedal Locomotion*. J.R. Osgood and Company, Boston, 1882. Executed and published under the auspices of Leland Stanford.
- [7] Carmelo Bellardita, Carmelo Bellardita, Ole Kiehn, and Ole Kiehn. Phenotypic characterization of speed-associated gait changes in mice reveals modular organization of locomotor networks. *Current Biology*, 2015.
- [8] Alexander Mathis, Pranav Mamidanna, Taiga Abe, Kevin M. Cury, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Markerless tracking of user-defined features with deep learning. *arXiv: Computer Vision and Pattern Recognition*, 2018.
- [9] Talmo D. Pereira, Nathaniel Tabris, Arie Matsliah, David M Turner, Junyu Li, Shruthi Ravindranath, Eleni S. Papadoyannis, Edna Normand, David Deutsch, Z. Yan Wang, Grace C. McKenzie-Smith, Catalin Mitelut, Marielisa Diez Castro, John D'Uva, Mikhail Kislin, Dan H. Sanes, Sarah D. Kocher, Samuel S.-H. Wang, Annegret L. Falkner, Joshua W. Shaevitz, Joshua W. Shaevitz, and Mala

- Murthy. Sleap: A deep learning system for multi-animal pose tracking. *Nature Methods*, 2022.
- [10] Tony Fong, Hao Hu, Pankaj Gupta, Braeden Jury, and Timothy H. Murphy. Pymousetracks: Flexible computer vision and rfid-based system for multiple mouse tracking and behavioral assessment. *eNeuro*, 10(5), 2023.
- [11] Pierre Karashchuk, Katie L. Rupp, Ewyn 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. *bioRxiv*, 2021.
- [12] Qualisys. Qualisys track manager. Software, 2023. Available from <https://www.qualisys.com/>.
- [13] Harold Hotelling and Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 1933.
- [14] Greg J. Stephens, Bethany L. Johnson-Kerner, William Bialek, and William S. Ryu. Dimensionality and dynamics in the behavior of *c. elegans*. *PLOS Computational Biology*, 2008.
- [15] Tosif Ahamed, Á. S. Costa, Antonio Carlos Costa, Antonio de Lisboa Lopes Costa, and Greg J. Stephens. Capturing the continuous complexity of behaviour in *caenorhabditis elegans*. *Nature Physics*, 2020.
- [16] Charles Webber and Joseph Zbilut. Recurrence quantification analysis of nonlinear dynamical systems. *Tutorials in Contemporary Nonlinear Methods for the Behavioral Sciences*, 01 2005.
- [17] Rakesh Veerabhadrappa, Imali T. Hettiarachchi, and Asim Bhatti. Using recurrence quantification analysis to quantify the physiological synchrony in dyadic ecg data. In *2021 IEEE International Systems Conference (SysCon)*, pages 1–8, 2021.
- [18] João A. Bastos and Jorge Caiado. Recurrence quantification analysis of global stock markets. *Physica A: Statistical Mechanics and its Applications*, 390(7):1315–1325, 2011.
- [19] Z. Q. ZHAO, S. C. LI, J. B. GAO, and Y. L. WANG. Identifying spatial patterns and dynamics of climate change using recurrence quantification analysis: A case study of qinghai-tibet plateau. *International Journal of Bifurcation and Chaos*, 21(04):1127–1139, 2011.
- [20] In-Ho Song, Doo-Soo Lee, and Sun I Kim. Recurrence quantification analysis of sleep electroencephalogram in sleep apnea syndrome in humans. *Neuroscience Letters*, 366(2):148–153, 2004.
- [21] Robert W. Taft, Robert A. Taft, Robert A. Taft, Muriel T. Davisson, Muriel T. Davisson, Muriel T. Davisson, Michael V. Wiles, and Michael V. Wiles. Know thy mouse. *Trends in Genetics*, 2006.

- [22] Bogna M. Ignatowska-Jankowska, Aysen Gurkan Ozer, Alexander Kuck, Micah J. Niphakis, Daisuke Ogasawara, Benjamin F. Cravatt, and Marylka Y. Uusisaari. Stimulatory effect of monoacylglycerol lipase inhibitor mjn110 on locomotion and step kinematics demonstrated by high-precision 3d motion capture in mice. *bioRxiv*, 2023.
- [23] Qualisys. Markers for motion capture. <https://www.qualisys.com/accessories/markers/>.
- [24] Bogna M. Ignatowska-Jankowska, Tara H. Turkki, Lakshmi Swaminathan, and Marylka Y. Uusisaari. Disruption of treadmill running by harmaline and cannabinoid agonist in marker-based 3d motion capture of mice. In *FENS Forum*, Vienna, Austria, June 24-29 2024.
- [25] Tara H. Turkki, Lakshmi Swaminathan, Bogna M. Ignatowska-Jankowska, and Marylka Y. Uusisaari. Disruption of climbing behavior by harmaline in marker-based 3d motion capture of freely moving mice. In *FENS Forum*, Vienna, Austria, June 24-29 2024.
- [26] Qualisys. How to use the trajectory editor - gap-filling: Fill types. https://docs.qualisys.com/getting-started/content/37_trajectory_editor_series/37b_how_to_use_the_trajectory_editor_-_gap-filling/fill_types.htm.
- [27] John L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 1965.
- [28] Yuta Sato, Takahiro Kondo, Reo Shibata, Masaya Nakamura, Hideyuki Okano, and Junichi Ushiba. Functional reorganization of locomotor kinematic synergies reflects the neuropathology in a mouse model of spinal cord injury. *Neuroscience Research*, 177:78–84, 2022.
- [29] Angelo Bartsch-Jimenez, Michalina Błażkiewicz, Hesam Azadrou, Ryan Novotny, and Francisco J. Valero-Cuevas. “fine synergies” describe motor adaptation in people with drop foot in a way that supplements traditional “coarse synergies”. *Frontiers in Sports and Active Living*, 2023.
- [30] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.
- [31] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [32] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [33] Carmelo Bellardita and Ole Kiehn. Phenotypic characterization of speed-associated gait changes in mice reveals modular organization of locomotor networks. *Current Biology*, 25(11):1426–1436, 2015. Epub 2015 May 7.
- [34] N. A. Bernshtejn. The co-ordination and regulation of movements. 1967.

- [35] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pages 366–381. Springer, 2006.
- [36] Rainer Hegger, Holger Kantz, Lorenzo Matassini, and Thomas Schreiber. Coping with nonstationarity by overembedding. *Phys. Rev. Lett.*, 84:4092–4095, May 2000.
- [37] Meta Platforms, Inc. React - a javascript library for building user interfaces. <https://react.dev/>.
- [38] Pallets Projects. Flask documentation. <https://flask.palletsprojects.com/en/3.0.x/>.
- [39] SQLAlchemy Project. Sqlalchemy documentation. <https://www.sqlalchemy.org/>.
- [40] MUI Core Team. Material ui documentation. <https://mui.com/material-ui/>.
- [41] Three.js Developers. Three.js documentation. <https://threejs.org/>.
- [42] scikit-learn developers. scikit-learn: Machine learning in python. <https://scikit-learn.org/stable/>.
- [43] Okinawa Institute of Science and Technology. Safe and responsible research. <https://www.oist.jp/research/safe-responsible-research>.