

PyMiceTracking: An Open-Source Toolbox For Real-Time Behavioral Neuroscience Experiments

Richardson Menezes
Department of Computer
Engineering and Automation
UFRN, Natal, Brazil

Aron de Miranda
Brain Institute
Natal, Brazil

Helton Maia
School of Science and Technology
UFRN, Natal, Brazil
helton.maia@ect.ufrn.br

Abstract

The development of computational tools allows the advancement of research in behavioral neuroscience and elevates the limits of experiment design. Many behavioral experiments need to determine the animal's position from its tracking, which is crucial for real-time decision-making and further analysis of experimental data. Modern experimental designs usually generate the recording of a large amount of data, requiring the development of automatic computational tools and intelligent algorithms for timely data acquisition and processing. The proposed tool in this study initially operates with the acquisition of images. Then the animal tracking step begins with background subtraction, followed by the animal contour detection and morphological operations to remove noise in the detected shapes. Finally, in the final stage of the algorithm, the principal components analysis (PCA) is applied in the obtained shape, resulting in the animal's gaze direction.

1. Introduction

The main goal of systems neuroscience is to describe causal relationships between neural circuit activity and animal behavior. Over the last two decades, new technologies for monitoring [1, 2] and manipulating [3] neural activity with striking spatiotemporal resolution in freely behaving animals have become available. To take full advantage of these technologies, we critically depend on the ability to interfere with real-time neural circuit activity in response to ongoing animal behavior. However, current technology for accurate online and real-time animal behavior tracking is a major computational challenge in neuroscience experiments.

There are many approaches to tracking animal head direction. Classically, head-orientation detection is performed by tracking at least two LEDs that are attached over the head of the animal for this specific purpose [4–6]. However, LEDs

represent additional weight for the animal headstage with significant consequences in the case of small animals, like the mouse.

Recent advances in tracking technology using machine learning [7] can satisfactorily solve the problem of head-direction tracking without the necessity of using LEDs. However, the computational cost of its implementation turns difficult its utilization for real-time online detection of behavioral parameters in neuroscience experiments.

This study presents the development of a computational tool called PyMiceTracking. It is written in the Python programming language. It allows simultaneous and real-time tracking of rodent body parts without markers (e.g., center of mass, snout tip, and tail base position), allowing for closed-loop experiments. PyMiceTracking also provides an algorithm for estimation of the yaw of the head (i.e., the head's orientation in the 2D coordinates over the plane where the animal navigates). Combining the simultaneous real-time tracking of mouse body parts and head orientation allows several possibilities for social and objects experimental investigation designs that could be associated with sensory and optogenetic stimulations.

It is intended to demonstrate the functioning and implementation details of the PyMiceTracking toolbox and the algorithms used for detecting body parts and estimating head orientation. Is presented a benchmark test using online available mouse behavioral datasets. Finally, we provide a step-by-step guide for using PyMiceTracking covering all installation steps and the creation of customized processing pipelines for experiments with closed-loop feedback stimulation.

2. Methods

2.1. Computational Development

Here, computational development techniques are described, including the environment for software development,

problem modeling, programming languages, and usability for researchers, the target audience for this type of research.

The Python programming language was adopted to develop the software proposed in this research. Also, the Open Source Computer Vision (OpenCV) library was essential for the acquisition and processing of the frames necessary for the analysis contained in this work. All steps involved for Mice Tracking and Head Orientation Detection can be viewed in summary form in the UML-based activity diagram shown in Figure 1.

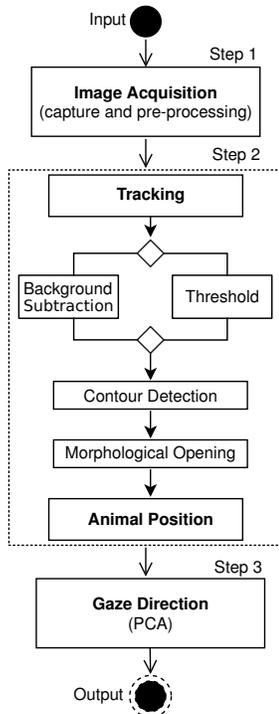


Figure 1. UML activity diagram of the developed system for the animal tracking and detection of head orientation.

In step 1, images are acquired from a video source, which can be a video file or even a live stream from a camera. Thus, the pre-processing of the images is performed using techniques such as noise filtering and histogram equalization, enabling improvements in image quality for the subsequent steps.

In step 2, the goal is to perform animal tracking. Initially, the user has to choose between two methods of tracking: background subtraction and threshold.

In the case of background subtraction, an absolute difference operation is performed between a reference background frame, possibly obtained before the animal entered the arena and the current frame. This subtraction will allow the removal of objects that may appear in the scene and reflections of animals on the acrylic or glass walls common in this type of experiment.

For threshold tracking, the Otsu method is applied. This method returns a single intensity threshold that separates pixels into two classes, foreground, and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance [8].

Next, a contour detection based on edge detection is executed, looking for the boundaries of a shape with the same intensity to determine the pixels that belong to the animal being tracked. The final part of this step is the execution of a computational process called morphological aperture that is performed on the shape obtained from the previous contour detection, improving the detection and considering the animal pixels that may have been excluded due to noise in previous steps.

Finally, in step 3, the Principal Component Analysis (PCA) algorithm is executed in a shape obtained from the previous morphological operation, extracting the essential features of the shape and giving the elliptical characteristic of mice. Thus, two axes can be traced, indicating the directions in which the most shape changes are. One of those axes shows the constant changes in the position of the animal's head, therefore, aligning the axis with the gaze direction.

2.2. Experimental setup

The software under development can be used in different animal behavioral experiments scenarios. To validate the computational development proposed in this work, we analyzed videos of behavioral experiments on social interaction with mice, kindly provided by [9]. Figure 2-a shows a 3D view of a typical setup for this mouse social interaction test. Figure 2-b shows an example of a frame captured from a camera with a superior view during a behavioral experiment.

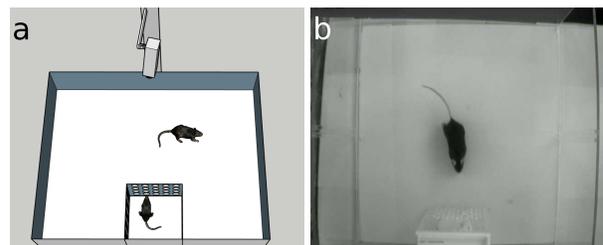


Figure 2. Setup designed for behavioral experiment of social interaction. a) 3D representation of the experimental configuration for the social interaction test. It is possible to visualize the arena for free movement, a retention cage for another animal and a camera for image acquisition. b) Example of captured frame in a top view of the arena during the social interaction test.

The images analyzed in this work for software validation come mainly from the behavioral testing environment presented in [9]. The tests initially placed the mouse in the

center of a square arena (white plexiglass open field, 37 cm on each side and 30 cm high), and the behavior was monitored using a video camera (Cineplex Studio, 50 fps) placed above the arena.

The experimental mouse was allowed to explore an arena under two different experimental sessions. In the first (“object” session), there was an empty perforated plexiglass cage placed in the middle of one wall of the arena. In the second session (“social” session), an unfamiliar male mouse was introduced into the cage as a social stimulus. Classically, patterns of arena occupation such as time spent in the vicinity of the cage and arena corners are compared between sessions and used to identify anxiety and depressive-like behaviors in mice.

This experimental setup where an arena is recorded from an upper camera and from there some behavioral analysis of the animal is performed is quite common [10–12]. Thus, the software presented in this paper has been validated in this scenario.

2.3. Image Processing Techniques

2.3.1 Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something more meaningful and easier to analyze. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share specific characteristics [13].

Generally, segmentation algorithms are either based on discontinuity, which analyses the boundaries of the regions in search of pronounced differences, or similarities that search the pixels of a region that can be clustered by similarity. Segmentation is very useful for line detection [14], edge detection [15, 16], region clustering [17] among other applications.

The segmentation process divides a region R into n connected sub-regions R_1, R_2, \dots, R_n , so that Equations 1 and 2 are satisfied.

$$R = \cup_{i=1}^n R_i \quad (1)$$

$$R_i \cap R_j = \emptyset \quad \forall i \neq j \quad (2)$$

2.3.2 Mathematical morphology

In nature, the term morphology refers to studying the form and structure of plants and animals. The mathematical morphology is the toolbox based on set theory through which the forms and structures in digital images are studied [18–21].

In binary images, the sets are members of the \mathbb{Z}^2 space where the coordinates are the pixel coordinates components.

For images in grayscale, the sets are members of the \mathbb{Z}^3 space where besides the pixel coordinates, an extra dimension is added to account for the pixel grayscale color.

The operations in mathematical morphology are based on small sets called structuring elements (SE) these are sub-images used to search an image for properties of interest. Some set operations can be performed over an input image with these elements, the main ones being erosion and dilation.

The erosion is the morphological operation through which the boundary of a set is worn out. Equation 3 formally defines the set operation performed.

$$A \ominus B = \{ z \mid (B)_z \subseteq A \} \quad (3)$$

On the other hand, dilation is the morphological operation through which the boundary of a set is expanded, formally defined by Equation 4.

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \} \quad (4)$$

Trough the combination of the aforementioned operations, two very useful operations can be defined, namely opening and closing.

The opening operation is used to soften outlines, breaks channels and eliminate small protrusions. The main application of the opening operation is to remove small error regions from noise in the segmentation. This operations is defined by Equation 5.

$$A \circ B = (A \ominus B) \oplus B \quad (5)$$

The closing operation is also used to soften outlines but tends to merge narrow discontinuities, eliminate small holes and fill gaps in a contour. The operation is defined by Equation 6.

$$A \cdot B = (A \oplus B) \ominus B \quad (6)$$

2.3.3 Principal Component Analysis

The Principal Component Analysis (PCA), also known as Karhunen-Loève Transform or Hotelling Transform [22, 23], perform an orthogonal transformation on a dataset of interdependent variables to another dataset of linearly de-correlated variables [24].

Therefore the PCA acts as a linear transformation that transfers the data to a new coordinate system, so that the largest variation by any projection of the data will be in the first coordinate (first principal component), the second-largest variation is in the second coordinate (second principal component), and so on.

One of the significant advantages of this type of analysis lies in the fact that the variables relevant to many problems are often correlated, which makes some of them redundant

at some level, so when using this type of transformation, the data can be analyzed in a much smaller dimension without losing information.

The PCA is formulated as follows, let $\bar{x} = [x_1, x_2, \dots, x_n]^T$ be a n-dimensional vector from the dataset so the average vector (7) and the matrix of covariance (8) of this dataset are define as:

$$\bar{m}_x = \frac{1}{M} \sum_{i=1}^M x_i \quad (7)$$

$$C_x = \frac{1}{M} \sum_{i=1}^M (\bar{x}_i \cdot \bar{x}_i^T) - \bar{m}_i \cdot \bar{m}_i^T \quad (8)$$

The matrix C_x is real and symmetric, therefore its eigenvalues λ_i are real and distinct, with their respective eigenvectors \bar{e}_i associated.

For the transformations the eigenvalues are sorted in descending order, thus $\lambda_j \geq \lambda_{j+1}$ where $j = 1, 2, \dots, n - 1$ and a matrix A is constructed with each of its lines being a unitary eigenvectors associated with the eigenvalues from the C_x matrix. The first row of A contains the eigenvector associated with the largest eigenvalue, therefore the last line will contain the eigenvector associated with the smallest eigenvalue.

The Hotelling Transform can be written as shown in Equation 9, where the transformed vectors have a zero average.

$$\bar{y} = A(\bar{x} - \bar{m}) \quad (9)$$

3. Results

The first step of the approach here proposed is to subtract a user-provided image of the experimental setup background from every frame captured by the camera, therefore obtaining a image where the pixels belonging to static elements in the scene are going to have a value close to 0 and pixels of moving objects are going to have values close to 255. Figure 3 depict this subtraction process.

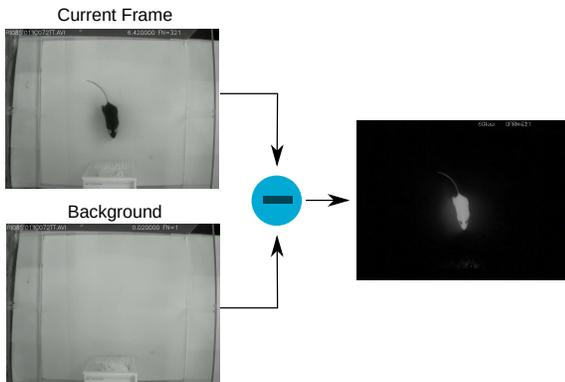


Figure 3. Subtraction process of user provided background image and a frame obtained from the experimental video.

The subtraction process is effective given that in the experimental configurations studied here, the lighting conditions rarely change, and the tracking subject stands out from the background. In Figure 4 a histogram plot of the previously subtracted frame can be seen, with the strategy here proposed the pixels belonging to the animal can be placed in the closed interval $[120, 160]$, as shown by the highlighted part of the figure, with this a band-pass like the filter is executed over the image where the pixels inside the interval are assigned a value of 1, and a value of 0 is given to that outside.

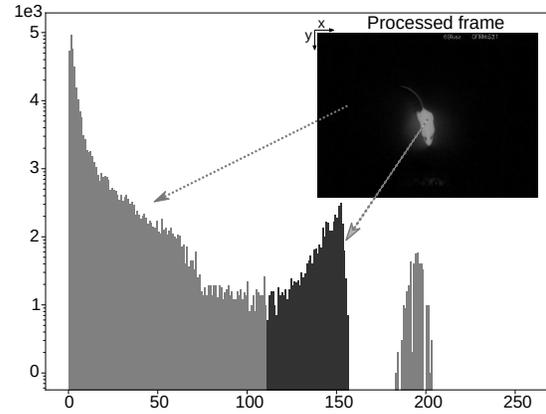


Figure 4. Histogram of the image subtraction result.

To further increase the system's robustness on top of the background subtraction, a series of morphological operations are performed over the binarized image, therefore, accounting for possible errors in the binarization process.

To harvest the benefits of morphological operations, the output of the binarization process depicted in Figure 5(a) passes through an erosion operation with the kernel depicted in Figure 5(K_1) this operation can be written as $A \ominus K_1$ with this all the possible noise is removed from the image however some parts of interest may also be removed, to overcome this problem a dilation operation with the kernel shown in Figure 5(K_2) is performed over the previous result thus increasing the area of interest and counterposing possibly removed parts. The final result is shown in Figure 5(b) and the complete operation can be written as $B = (A \ominus K_1) \oplus K_2$.

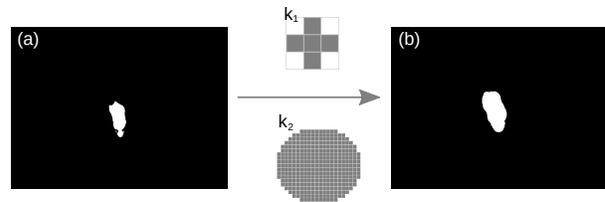


Figure 5. Process of morphological opening

Figure 6(a) shows the input to the final step in the compu-

tational tool proposed here. In this image, the PCA algorithm is executed, and the two axes can be found representing the principal components of the image, one of which will always indicate the direction of the mice’s gaze. These axes and the masks’ center of mass are depicted in Figure 6(b).

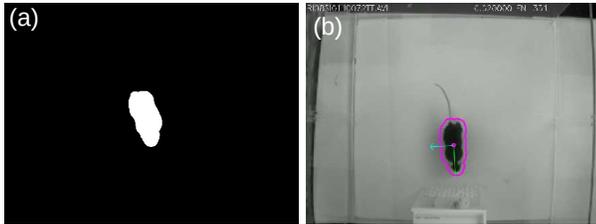


Figure 6. Input and output for the PCA algorithm.

With the tool, researchers can analyze the animal’s trajectory throughout the arena during the experiment as shown in Figure 7(a). In addition, from the trajectory points, a heatmap plot can be generated to analyze the prolonged presence of the animal in specific parts of the arena as depicted in Figure 7(b). Furthermore, it’s possible to select regions of interest (ROI) where the amount of time the animal spent inside will be automatically accounted as shown in Figure 7(c).

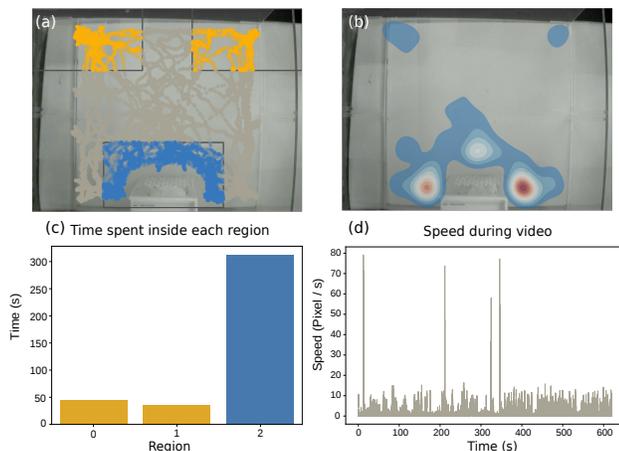


Figure 7. Examples of analyses performed with the tool: (a) Trajectory plot of the position of a mice during a social session showing the interaction zone (blue) and corners (yellow); (b) Heatmap plot of the trajectory showing the prolonged presence of the animal around the interaction zone; (c) Time spent inside each region depicted where Region 0 and 1 are the top left and top right respectively and Region 2 is the bottom one; (d) Speed of the animal during the experiment.

Moreover, the previously described analyses capabilities the tool also provides an interface for tracking and plotting the animal’s speed during the experiment, as shown in Figure 7(d).

To facilitate the access to the tools presented, a graphical user interface (GUI) showed in Figure 8 was developed. There, the user can load the video and visualize the frames being processed while making corrections on the fly to the hyperparameters of the tracking algorithm.

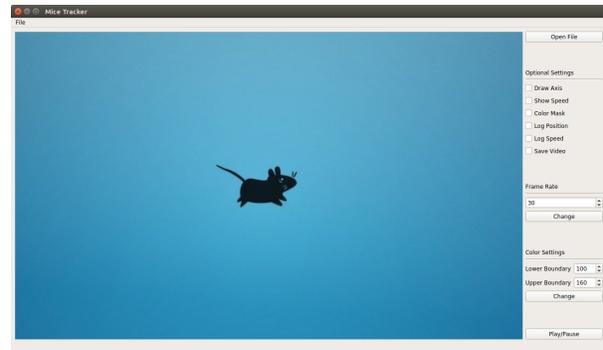


Figure 8. User interface developed.

With the computational tool presented by this work, researchers in neuroscience can automatically analyze data collected from the position of an animal during behavioral experiments, therefore, being able to determine if the animal spent more time exploring the ambient or isolated in the corners and with this information inferences in the animal’s level of anxiety can be made [25].

4. Conclusion

This study described the computational development that uses digital image processing to detect and track mice during behavioral neuroscience experiments. Researchers worldwide can have a reliable, real-time, and fully automated tracking system with the proposed tool. The entire system can be modified and adapted for different needs with open-source development.

All computational tools developed during this work can be found in a repository under the General Public License (GPL) available at <https://github.com/xarmison/proj-pca>, which includes examples and tutorials for the usage of the proposed tools.

5. Acknowledgment

This work was supported by the School of Sciences and Technology at the Federal University of Rio Grande do Norte (ECT-UFRN).

References

- [1] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın *et al.*, “Fully integrated silicon

- probes for high-density recording of neural activity,” *Nature*, vol. 551, no. 7679, pp. 232–236, 2017. 1
- [2] W. Zong, R. Wu, M. Li, Y. Hu, Y. Li, J. Li, H. Rong, H. Wu, Y. Xu, Y. Lu *et al.*, “Fast high-resolution miniature two-photon microscopy for brain imaging in freely behaving mice,” *Nature methods*, vol. 14, no. 7, pp. 713–719, 2017. 1
- [3] E. S. Boyden, F. Zhang, E. Bamberg, G. Nagel, and K. Deisseroth, “Millisecond-timescale, genetically targeted optical control of neural activity,” *Nature neuroscience*, vol. 8, no. 9, pp. 1263–1268, 2005. 1
- [4] J. O’Keefe and J. Dostrovsky, “The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat.” *Brain research*, 1971. 1
- [5] L. M. Giocomo, T. Stensola, T. Bonnevie, T. Van Cauter, M.-B. Moser, and E. I. Moser, “Topography of head direction cells in medial entorhinal cortex,” *Current Biology*, vol. 24, no. 3, pp. 252–262, 2014. 1
- [6] J. I. Sanguinetti-Scheck and M. Brecht, “Home, head direction stability, and grid cell distortion,” *Journal of neurophysiology*, vol. 123, no. 4, pp. 1392–1406, 2020. 1
- [7] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, “DeepLabcut: markerless pose estimation of user-defined body parts with deep learning,” *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018. 1
- [8] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. 2
- [9] A. M. Henriques-Alves and C. M. Queiroz, “Ethological evaluation of the effects of social defeat stress in mice: beyond the social interaction ratio,” *Frontiers in behavioral neuroscience*, vol. 9, p. 364, 2016. 2
- [10] M. J. Kas, A. J. de Mooij-van Malsen, B. Olivier, B. M. Spruijt, and J. M. van Ree, “Differential genetic regulation of motor activity and anxiety-related behaviors in mice using an automated home cage task.” *Behavioral neuroscience*, vol. 122, no. 4, p. 769, 2008. 3
- [11] M. Yang and J. N. Crawley, “Simple behavioral assessment of mouse olfaction,” *Current protocols in neuroscience*, vol. 48, no. 1, pp. 8–24, 2009. 3
- [12] M. Yang, J. L. Silverman, and J. N. Crawley, “Automated three-chambered social approach task for mice,” *Current protocols in neuroscience*, vol. 56, no. 1, pp. 8–26, 2011. 3
- [13] G. Stockman and L. G. Shapiro, *Computer Vision*, 1st ed. USA: Prentice Hall PTR, 2001. 3
- [14] L. A. Fernandes and M. M. Oliveira, “Real-time line detection through an improved hough transform voting scheme,” *Pattern recognition*, vol. 41, no. 1, pp. 299–314, 2008. 3
- [15] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, “Image selective smoothing and edge detection by nonlinear diffusion,” *SIAM Journal on Numerical analysis*, vol. 29, no. 1, pp. 182–193, 1992. 3
- [16] R. S. T. Menezes, L. d. A. Lima, O. Santana, A. M. Henriques-Alves, R. M. S. Cruz, and H. Maia, “Classification of mice head orientation using support vector machine and histogram of oriented gradients features,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–6. 3
- [17] B. C. Ko and J.-Y. Nam, “Object-of-interest image segmentation based on human attention and semantic region clustering,” *JOSA A*, vol. 23, no. 10, pp. 2462–2470, 2006. 3
- [18] R. M. Haralick, S. R. Sternberg, and X. Zhuang, “Image analysis using mathematical morphology,” *IEEE transactions on pattern analysis and machine intelligence*, no. 4, pp. 532–550, 1987. 3
- [19] F. Zana and J.-C. Klein, “Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation,” *IEEE transactions on image processing*, vol. 10, no. 7, pp. 1010–1019, 2001. 3
- [20] P. Trahanias, “An approach to qrs complex detection using mathematical morphology,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 2, pp. 201–205, 1993. 3
- [21] Z. Yu-Qian, G. Wei-Hua, C. Zhen-Cheng, T. Jing-Tian, and L. Ling-Yun, “Medical images edge detection based on mathematical morphology,” in *2005 IEEE engineering in medicine and biology 27th annual conference*. IEEE, 2006, pp. 6492–6495. 3
- [22] K. Pearson, “Liii. on lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. 3
- [23] H. Hotelling, “Analysis of a complex of statistical variables into principal components.” *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933. 3
- [24] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987. 3

- [25] J. Winne, R. Franzon, A. de Miranda, T. Malfatti, J. Patriota, S. Mikulovic, K. E. Leão, and R. N. Leão, “Salicylate induces anxiety-like behavior and slow theta oscillation and abolishes the relationship between running speed and fast theta oscillation frequency,” *Hippocampus*, vol. 29, no. 1, pp. 15–25, 2019. 5