Toward Constrained Animal Pose Estimation

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Arne Monsees

aus Wuppertal

Tübingen

2022

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	04.07.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Jakob Macke
2. Berichterstatter/-in:	Prof. Dr. Gerard Pons-Moll

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Acronyms

avg.	average
CNN	convolutional neural network
E-step	expectation step
e.g.	*exempli gratia* (for example)
ELBO	evidence lower bound
EM	expectation-maximization
FTIR	frustrated total internal reflection
i.e.	*id est* (that is)
KL	Kullback–Leibler
M-step	maximization step
max.	maximum
min.	minimum
MRI	magnetic resonance imaging
RTS	Rauch-Tung-Striebel
s.d.	standard deviation
SMPL	skinned multi-person linear
vs.	versus

Acknowledgments

First of all, I would like to thank Jason Kerr for giving me the opportunity to work on my PhD project within the sheltering realm of the Max Planck Society. Pursuing my PhD in the scientific environment offered by the Max Planck Institute (MPI) for Neurobiology of Behavior as well as the International Max Planck Research School (IMPRS) for Brain and Behavior was certainly a great learning experience.

Furthermore, I would like thank Jakob Macke for his perpetual supervision, support and advice during my doctorate. Without him a successful completion of my PhD would have been infeasible.

Besides my two primary supervisors Jason Kerr and Jakob Macke, I would also like to thank Gerard Pons-Moll and Andreas Geiger for their willingness to evaluate my thesis and participate in my PhD defense.

Also, I would like to thank all current members of the Department of Behavior and Brain Organization (BBO), who helped me throughout my PhD, particularly my co-authors Kay-Michael Voit, Damian Wallace and Jürgen Sawinski, but also Uwe Czubayko, Carl Holmgren, Jeanine Klesing, Ruth Pohle and Paul Stahr. Besides, I would also like to thank former BBO members, particularly Giacomo Bassetto, Pranjal Dhole, Florian Franzen, Patrick Rose and David Greenberg, who has not only been a mentor to me but also a friend.

Additionally, my gratitude extends to Edyta Charyasz (former Edyta Leks), and Klaus Scheffler, who provided substantial help with generating MRI data, as well as Jan-Matthis Lückmann, Pedro Gonçalves, Artur Speiser, Jan Bölts, Marcel Nonnenmacher and Poornima Ramesh, who provided seldom yet highly-appreciated help, input and discussions revolving around technical questions and challenges of my PhD project, while not being limited to such topics.

Finally, I would also like to thank all supporting colleagues from the MPI for Neurobiology of Behavior, all fellow PhD students from the IMPRS for Brain and Behavior as well as my funding sources, i.e. the Max Planck Society, particularly the MPI for Neurobiology of Behavior, as well as the Center of Advanced European Studies and Research.

Besides expressing my gratitude to scientific companions, I would also like to thank my family, particularly Carolin Monsees for being an outstanding and supportive person, role model and overall great sister. Lastly, I would like to thank my close friends, who are like my additional family, as well as Zaina Batool for being the supportive, intelligent, easy-to-talk-to, open-minded, discussion-friendly and indispensable partner on my side.

Scientific Contributions

The work presented in this thesis resulted in the following preprint:

- **Monsees, A.**, Voit, K.-M., Wallace, D. J., Sawinski, J., Leks, E., Scheffler, K., Macke, J. H., and Kerr, J. N. D.; Anatomically-based skeleton kinetics and pose estimation in freely-moving rodents; bioRxiv; 2021

Consequently, the contents of this thesis and the aforementioned preprint are overlapping. All authors of the preprint were involved in the project and contributed to it.

Jason Kerr provided the initial idea for developing a method for tracking freely-moving animals with the intention to relate their behavior to an accurate anatomical model. He furthermore contributed to the development of the proposed pose estimation framework and the use of anatomical constraints. Additionally, he provided the idea for using MRI scans and FTIR imaging to facilitate comparisons between reconstructed and ground truth skeletal poses. Lastly, he also provided the idea for using gap-crossing tasks in order to analyze complex animal behavior. Jakob Macke contributed by conceptualizing a structured plan for implementing and completing the project. Additionally, he contributed to the computational strategy and provided suggestions and guidance with respect to technical details of the proposed pose estimation framework. Klaus Scheffler and Edyta Leks were involved in performing MRI scans of all animal subjects and developing the respective scan sequences. Juergen Sawinski built the setup used for generating video data via FTIR imaging, including the FTIR plate. Damian Wallace was involved in generating video data of freely-behaving animal subjects. Particularly, he was in charge of anesthetizing animal subjects for applying surface markers and handled animal subjects during experiments. Additionally, he trained animal subjects for gap-crossing tasks. Kay-Michael Voit was involved in building the setups used for generating video data. Particularly, he substantially contributed by developing and building hardware for synchronizing video cameras, recording video data and illuminating setups. Additionally, he also recorded video data during experiments and helped to maintain and administer the used operating systems.

As the first author of the preprint, I conceptualized and developed the proposed pose estimation framework, including all methods and respective code, with contributions of the other authors as described above. Particularly, I wrote software for synchronizing video cameras and calibrating multi-camera setups. Additionally, I also wrote software to implement the skeletal model including all constraints and numerical optimization schemes, the used Bayesian filter and smoother as well as the EM algorithm. Furthermore, I conceptualized and developed the code for the graphical user interfaces required for exploring and labeling the recorded video and MRI data. With the exception of the gap-crossing track, I built the setups for generating video data of freely-behaving animals in strong collaboration with Kay-Michael Voit. Particularly, this includes building arenas in which animals could move freely as well as mounting and adjusting video cameras and setup illumination hardware. Furthermore, I developed and produced the used MRI markers, applied surface markers and MRI markers to animal subjects and calibrated the used multi-camera setups. Lastly, I carried out, conducted and supervised the generation of video data in collaboration with all authors. Particularly, Jason Kerr, Juergen Sawinski, Damian Wallace, Kay-Michael Voit and me were

involved in generating the video data, whereas Jason Kerr, Klaus Scheffler, Edyta Leks and me were involved in generating the MRI data. Edyta Leks performed MRI scans and developed scan sequences.

The initial draft of the preprint was written by me, whereas the figures of the draft were conceptualized by Jason Kerr and me. The figures and videos as well as the main text of the final version of the preprint were conceptualized and written by Jason Kerr and me, whereas all other authors provided comments and revisions. The supplementary text of the preprint was written by me, whereas Kay-Michael Voit and Juergen Sawinski provided comments and revisions. The code for generating all figures and videos as well as for performing all analyses and statistical tests in the preprint was developed and written by me.

Ultimately, the aforementioned preprint resulted in the following peer-reviewed publication:

- **Monsees, A.**, Voit, K.-M., Wallace, D. J., Sawinski, J., Charyasz, E., Scheffler, K., Macke, J. H., and Kerr, J. N. D.; Estimation of skeletal kinematics in freely moving rodents; Nature Methods; 2022

After the initial submission of the preprint, further experiments, i.e. recording additional behavioral data in mice and additional kinematic data in rats using inertial measurement units (IMU) as well as corresponding MRI scans of the animal subjects, were conceived and performed by Kay-Michael Voit, Damian Wallace, Juergen Sawinski, Edyta Charyasz (former Edyta Leks) and Jason Kerr.

I provided advice on how to use and adapt the proposed pose estimation framework to analyze the resulting data. Kay-Michael Voit improved and extended the framework's interface and workflow to make it usable by other users and suitable for publication in Nature Methods. The additional behavioral data was analyzed using the original code, which was conceptualized and developed by me and further improved by Kay-Michael Voit, whereas the additional kinematic data was analyzed using code, which was conceptualized and developed by Kay-Michael Voit. For the additional kinematic data, Juergen Sawinski built a customized IMU system and wrote firmware to use it, whereas Kay-Michael Voit developed code for analyzing it. Kay-Michael Voit, Damian Wallace, Juergen Sawinski and Jason Kerr wrote additional text describing the added results and analyses and revised the manuscript, with comments on revisions from me.

Besides the main authors, the preparation of the preprint and the peer-reviewed publication was supported by additional people, as listed in the acknowledgment sections of both manuscripts: David Greenberg developed video acquisition software. Florian Franzen conceptualized and designed an external camera trigger together with Kay-Michael Voit. Jan-Matthis Lückmann developed initial code for reading single images from recorded video data. Michael Bräuer, Rolf Honnef, Bernd Scheiding and Michael Straussfeld fabricated setup components. Oliver Holder supported the deployment of IMUs. Kristina Barragan, Caleb Berdahl, Abhilash Cheekoti, Uwe Czubayko, Nada Eiadeh, Yvonne Grömping, Gizem Görünmez, Carl Holmgren, Katrin Junker, Jeanine Klesing, Alexandr Klioutchnikov, Po-Yu Liao, Yolanda Mabuto, Daniela Martin Machado, Anastasiia Nychyporchuk, Aarya Pawar, Verena Pawlak, Paul Stahr, Adam Sugi and Nurit Zorn manually labeled images to enable the training of artificial neural networks. Julia Kuhl provided illustrations.

During the course of my doctoral research, I also worked on other scientific subjects, which resulted in the following preprint:

- Greenberg, D. S., Wallace, D. J., Voit, K.-M., Wuertenberger, S., Czubayko, U., **Monsees, A.**, Handa, T., Vogelstein, J. T., Seifert, R., Groemping, Y., and Kerr, J. N. D.; Accurate action potential inference from a calcium sensor protein through bio-physical modeling; bioRxiv; 2018

Due to the different nature of the research field, the contents of this second preprint are not part of this thesis.

Abstract

Quantifying animal behavior is a crucial aspect of the ongoing neuroscientific endeavor to understand the brain, since it is a prerequisite for studying how neural computations relate to behavioral outputs. One method for obtaining an objective yet detailed description of an animal's unconstrained and therefore natural behavior is given by estimating its pose, i.e. the collective positions and orientations of all individual body parts in space at a given moment in time. While various approaches have been proposed for estimating the pose of a freely-moving animal, so far, studies relying on video cameras for recording the required behavioral data have neglected reconstructing the actual skeleton of an animal and only considered inferring the positions of anatomical landmarks located on its body surface. Additionally, many approaches lack incorporating mechanistic knowledge of an animal's anatomy, which leaves room for improving the resulting pose reconstruction accuracy. Consequently, methods for quantifying skeletal animal poses during free motion sequences are desirable tools for future neuroscientific studies.

The work presented in this thesis tackles the problem of inferring skeletal poses from recorded video data of freely-moving animal subjects via a constrained animal pose estimation framework, which enables reconstructing underlying three-dimensional joint positions from observable surface markers while enforcing anatomical and temporal constraints. Anatomical constraints are implemented via a realistic skeleton model, which accounts for physiological joint angle limits, bone lengths and body symmetry. Besides, the realistic skeleton model allows for learning individual skeleton anatomies directly from recorded video data of behaving animals, taking into account subject-specific differences with respect to bone lengths and body-symmetry. Furthermore, to ensure that reconstructed joint positions follow smooth motion trajectories, the proposed animal pose estimation framework also enforces temporal constraints. Particularly, temporal constraints are implemented via an underlying state space model, which allows for deploying a Bayesian smoother for inferring bone rotations as well as an expectation-maximization algorithm for learning the unknown probabilistic hyper-parameters of the state space model.

The proposed animal pose estimation framework is evaluated and tested with respect to its reconstruction accuracy and usability for quantifying a range of different behaviors. By comparing learned skeleton anatomies with ground truth data obtained via magnetic resonance imaging, it is shown that the framework offers the opportunity to learn three-dimensional joint positions and bone lengths solely from two-dimensional video data. Besides, to test whether poses of freely-moving animals are accurately inferred, independently measured paw positions are obtained using a frustrated total internal reflection imaging system and compared to their reconstructed counterparts, while the effects of the enforced anatomical and temporal constraints are analyzed. This analysis shows the advantages of constrained over unconstrained animal pose estimation, since enforcing constraints reduces errors with respect to reconstructed paw positions and orientations. Furthermore, to assess if the proposed pose estimation framework is capable of accurately quantifying common behaviors, periodic gait cycles are analyzed based on reconstructed skeletal poses, which shows that enforcing constraints is essential for successfully extracting characteristic movement patterns from recorded video data. Finally, the proposed pose estimation framework is also

used to quantify complex gap-crossing behaviors, where animals jump over gaps of various distances. This analysis shows that reconstructing skeletal poses enables computing characteristic movement patterns during jumping and correlating skeletal kinematic quantities with each other as well as the jumped distances.

In summary, this thesis proposes an animal pose estimation framework, which allows for reconstructing anatomically-plausible as well as time-consistent three-dimensional skeletal poses of freely-moving animals from two-dimensional video data. To achieve this, anatomical and temporal constraints are implemented into the proposed pose reconstruction framework, which transpired to be essential for obtaining accurate pose reconstruction results. Consequently, this thesis contains analyses, which demonstrate the importance of the implemented constraints in the context of animal pose estimation.

Kurzfassung

Die Quantifizierung von tierischen Bewegungsmustern ist ein Grundpfeiler fortwährender neurowissenschaftlicher Bestrebungen das Gehirn zu verstehen, da es eine Grundvoraussetzung für die Erforschung der wechselseitigen Beziehung zwischen neuronaler Aktivität und Tierverhalten darstellt. Eine verbreitete Methode zur objektiven und detaillierten Beschreibung von unbeschränkten und daher natürlichen Bewegungsmustern eines Tieres ist die Schätzung der Körperhaltung, welche die kollektiven Positionen und Ausrichtungen aller einzelnen Körperteile im Raum zu einem bestimmten Zeitpunkt beinhaltet. Zwar existieren verschiedene Ansätze für die Schätzung der Körperhaltung eines sich frei bewegenden Tieres, doch haben Studien, bei denen Videokameras für die Auzeichnung der erforderlichen Verhaltensdaten verwenden werden, bisher die Rekonstruktion des eigentlichen Skeletts vernachlässigt und sich lediglich mit der Berechnung von Positionen anatomischer Merkmale auf der Körperoberfläche des Tieres befasst. Darüber hinaus mangelt es vielen Ansätzen an der expliziten Miteinbeziehung von mechanistischem Wissen über die Anatomie eines Tieres, was die Genauigkeit von Körperhaltungsrekonstruktionen negativ beeinträchtigen kann. Folglich sind Methoden zur objektiven sowie detaillierten Quantifizierung tierischer Skelett-Konfigurationen wünschenswerte Werkzeuge für zukünftige neurowissenschaftliche Studien.

Die in dieser Dissertation vorgestellten Inhalte liefern einen Lösungsvorschlag für das Problem die Skelett-Konfigurationen von sich frei bewegenden Tieren auf Basis von aufgezeichneten Videodaten zu berechnen. Dafür wird ein entsprechender Algorithmus vorgeschlagen, welcher es ermöglicht, dreidimensionale Gelenkpositionen aus beobachtbaren Oberflächen-Markierungen zu rekonstruieren und dabei anatomische und zeitliche Beschränkungen berücksichtigt. Anatomische Beschränkungen werden durch die Einbeziehung eines realistischen Skelett-Modells erfasst, welches physiologische Gelenkwinkelgrenzen, Knochenlängen und Körpersymmetrien beinhaltet. Darüber hinaus ermöglicht die Verwendung eines realistischen Skelett-Modells das Erlernen individueller Skelette auf Basis von aufgezeichneten Videodaten von sich frei bewegender Tiere, wobei subjektspezifische Unterschiede in Bezug auf Knochenlängen und Körpersymmetrie berücksichtigt werden. Um sicherzustellen, dass die rekonstruierten Gelenkpositionen kontinuierlichen Bewegungstrajektorien folgen, erzwingt der Algorithmus auch zeitliche Beschränkungen. Diese werden über ein Zustandsraummodell implementiert, welches die Rekonstruktion von Skelett-Konfigurationen durch die Bestimmung von Knochenrotationen mittels eines Bayesschen Filter- beziehungsweise Glättungs-Algorithmus ermöglicht. Darüber hinaus erlaubt dieses Vorgehen ebenfalls das Erlernen der unbekannten probabilistischen Hyper-Parameter des Zustandsraummodells mittels eines Erwartungs-Maximierungs-Algorithmus.

Die daraus resultierenden rekonstruierten Skelett-Konfigurationen werden im Zuge dieser Dissertation auf ihre Genauigkeit hin bewertet sowie auf ihre Verwendbarkeit hinsichtlich der Quantifizierung von verschiedenen Verhaltensweisen hin getestet. Durch den Vergleich von erlernten Skeletten mit aus der Magnetresonanztomographie gewonnenen Daten wird gezeigt, dass der vorgeschlagene Algorithmus die Möglichkeit bietet, dreidimensionale Gelenkpositionen sowie Knochenlängen allein aus zweidimensionalen Videodaten zu berechnen. Um die Genauigkeit

der berechneten Skelett-Konfigurationen von sich frei bewegenden Tieren zu testen, werden Pfotenpositionen mittels eines Bildgebungsverfahrens, welches auf dem Konzept der verhinderten totalen Reflektion basiert, gemessen und mit ihren rekonstruierten Gegenstücken verglichen, wobei die Auswirkungen der erzwungenen anatomischen und zeitlichen Beschränkungen analysiert werden. Diese Analyse zeigt die Vorteile der beschränkten gegenüber der unbeschränkten Berechnung von Skelett-Konfigurationen, da die Einbeziehung von Beschränkungen die Fehler in Bezug auf die rekonstruierten Pfotenpositionen und -orientierungen verringert. Um zu beurteilen, ob der vorgeschlagene Algorithmus in der Lage ist, gängige Verhaltensweisen genau zu quantifizieren, werden außerdem periodische Gangzyklen auf der Grundlage rekonstruierter Skelett-Konfigurationen analysiert. Diese Analyse zeigt, dass das Erzwingen der anatomischen und zeitlichen Beschränkungen eine wesentliche Voraussetzung für die erfolgreiche Extraktion von charakteristischen Bewegungsmustern aus aufgezeichneten Videodaten ist. Schließlich werden auch komplexere Verhaltensweisen quantifiziert, bei denen Tiere über Lücken unterschiedlicher Länge springen müssen. Diese Analyse zeigt, dass die Rekonstruktion von Skelett-Konfigurationen die Berechnung von charakteristischen Bewegungsmustern während des Springens und von Korrelationen zwischen kinematischer Metriken untereinander sowie mit den gesprungenen Distanzen ermöglicht.

Zusammenfassend wird in dieser Dissertation ein Algorithmus zur Schätzung von Skelett-Konfigurationen vorgestellt, welcher anatomisch plausible sowie zeitlich konsistente Gelenkpositionen und Knochenorientierungen von sich frei bewegenden Tieren liefert. Um dies zu erreichen, werden anatomische und zeitliche Beschränkungen in den Algorithms implementiert, die sich als wesentlicher Bestandteil für die Erzielung genauer Posenrekonstruktionsergebnisse erwiesen haben. Folglich enthält diese Arbeit Analysen, welche die Bedeutung der implementierten Beschränkungen im Rahmen der Schätzung von tierischen Skelett-Konfigurationen hervorhebt.

Chapter 1

Introduction

1.1 Motivation and background

A major goal of neuroscientific research is to gain mechanistic knowledge about the brain by studying how neural activity gives rise to the behavior of animals and vice versa [1–3]. While this requires accurate measurements of neural data to obtain insights into the computations performed by the brain, it is equally important to correctly quantify animal behavior to determine how neural computations relate to behavioral outputs [1–8]. In the past, simultaneously analyzing brain activity and animal behavior has already led to insights into neural circuits and mechanisms, e.g. when it was discovered that the isolated activity of specific neurons is highly correlated with the spatial position of an animal [9, 10]. Since it is hypothesized that the identification of fine-grained behavioral motifs will lead to an even richer understanding about the functionality and implementation of neural circuits, deploying robust and accurate approaches for extracting both, behavioral as well as neural data, is crucial for future neuroscientific research [2, 3, 11]. Thus, to perpetually advance the limits of neuroscientific research it is key to continuously develop new and permanently improve existing measuring techniques in order to overcome the shortcomings of currently predominant methods [1, 3, 6]. With the intention of contributing to this endeavor the contents presented in this thesis are aimed at pushing the limits of what is currently possible in the realm of quantifying neuroscientific data.

1.1.1 Quantifying neural activity

Electrophysiology is regarded as the gold standard for studying the brain, since it allows for measuring neural activity via microelectrodes, which yields ground truth voltage traces of individual neurons [12, 13]. Penetrating the membrane of a neuron with an electrode allows for recording the local voltages inside the cell. This process is a direct measure of the cell's neuronal activity and can even be achived in fully awake animals [14]. Since each recorded neuron requires a single electrode and establishing a sealed connection between the electrode and the cell is usually a manual and labor-intensive task with relatively low success rates [12], using this technique to simultaneously measure the activity of a large amount of neurons is challenging. However, simultaneously recording the activity of an entire ensemble of neurons can be achieved, for instance, by using entire arrays of microelectrodes [15]. The costs of obtaining voltage traces via those arrays are given by comparably imprecise recordings, since the placement of the respective arrays is usually less targeted. As a consequence the individual electrodes are not necessarily penetrating neurons directly, which causes extracellular background noise to be recorded as well when micoelectrode arrays are used [15].

By utilizing calcium-sensitive fluorescent proteins combined with high-resolution microscopy, it

is possible to overcome these limitations [16,17]. When such proteins are located within individual neurons they bind calcium ions and, as a result, emit fluorescent light, which is detectable via microscopes. Since neural activity is positively correlated to the local calcium concentration within a cell, a high light intensity corresponds to a high neural activity [18]. Consequently, measuring the neural activity of many neurons at the same time becomes feasible using this technique [19, 20]. For specific microscopic animals, like nematodes, whose organisms only encompass a few hundred neurons in total [21], the technique of measuring neural activity via calcium-sensitive fluorescent proteins can even be scaled up to such an extend that the ensemble of recorded neurons represents the entire nervous system of the animal subject [22].

When larger and more complex animals, like rodents, are studied, recording brain activity becomes more challenging. Due to their larger brain sizes, neurons can be located more than a thousand microns below the brain's surface [23], causing microscopic imaging techniques to be less efficient, e.g. due to increased light scattering [24,25]. Furthermore, compared to nematodes the overall number of neurons in rodent brains is several magnitudes larger, which, so far, deems measuring the entire neural activity of a rodent model organism an infeasible endeavor. Nevertheless, for rodents it is still possible to simultaneously record activity traces of dozens of neurons using miniaturized microscopes, which are light-weight and only cause marginal restrictions to an animal's movement capabilities, such that neural activity can be measured while the animal is showing unconstrained behaviors [26–28].

1.1.2 Quantifying animal behavior

While techniques for measuring the brain's neural functions are well-established (Section 1.1.1), methodological advances in the realm of quantifying unconstrained behavior have been comparably sparse, such that, so far, accurately describing how a freely-moving animal is interacting with its environment remains a less standardized procedure [1]. In fact, many different approaches for estimating animal behavior have been proposed, ranging from inertial measurement units [29,30] to radio-frequency identified tagging [31,32] and videography with either normal [33–35] or time-of-flight cameras [36–38]. Among these, the deployment of normal video cameras represents an accurate, non-invasive, easy to use and cost-efficient option for monitoring a behaving animal. Consequently, videography is regarded as a promising measuring technique for future developments in the area of behavior quantification [7].

One option for tracking and describing animal behavior via cameras is given by quantifying an animal's pose at any given time point with high spatial and temporal resolution. This yields knowledge about the animal's entire body posture, which, for instance, includes the positioning of individual limbs as well as the direction in which the head is pointing [8].

However, a common challenge faced by all camera-based pose estimation approaches is the elimination of ambiguities, which arise when three-dimensional poses are reconstructed from two-dimensional images [39,40]. Since depth information is lost, when a scene with a behaving animal is projected onto the sensor of a camera [41], many different three-dimensional poses can be consistent with the two-dimensional pose, which is observed on the recorded image [40, 42]. While using multiple cameras can limit the effect of such ambiguities and therefore allows for reconstructing three-dimensional poses, e.g. via triangulation, it is not necessarily guaranteed that the deployment of a multi-camera setup leads to accurate pose reconstructions, e.g. when individual body parts are temporally occluded in all but one camera view [40]. In this scenario only a single camera provides information about the entire body pose, which negates the benefits of using a multi-camera setup since recovering the three-dimensional pose via mere triangulation becomes infeasible. Furthermore, deploying an exceptionally high amount of cameras is cost-intensive and introduces challenges on its own with respect to the logistics of capturing and storing the result-

ing video data [43]. Additionally, low hardware requirements in the form of a limited amount of used cameras are generally preferable, since measuring naturalistic animal behaviors is regarded a long-term neuroscientific goal [3, 7], which implies that respective studies have to be conducted in the wild – outside of a perfectly controllable lab environment.

This causes a demand for appropriate pose estimation algorithms, yielding accurate results based on a limited amount of video data, which is recorded from only a few video cameras. To obtain algorithms with the desired robustness, deploying probabilistic instead of deterministic models within the context of animal pose estimation is a viable option. Additionally, anatomical and temporal constraints can be introduced to a pose estimation framework in order to boost its accuracy and performance [6–8, 42].

1.1.3 Constrained pose estimation

Taking into account mechanistic knowledge about the movement capabilities of an animal is one option for reducing the number of ambiguities in a pose estimation framework, since many animals are physically limited in their range of motions by anatomical constraints [44]. Anatomical constraints naturally exist in many animal species in the form of skeletons, which limit the amount of possible bone configurations [42]. Anatomical skeletons consist of joints and bones, where each bone is connected to a specific type of joint, which usually has between one and three rotational degrees of freedom. This constrains rotational bone motions to stay within physiological joint angle limits. A particular subset of anatomical constraints is given by spatial constraints, which describe the rigid nature of bones, i.e. each skeletal bone is a single entity of constant length, which is attached to its respective joints and is only allowed to rotate and move as a whole [6, 8, 42]. Thus, the solution space for possible poses can be narrowed within a pose reconstruction framework, since a fraction of otherwise valid poses can be rejected, when they fail to comply with the enforced anatomical constraints. Consequentially, introducing anatomical constraints into a pose estimation framework can improve the quality of reconstructed poses [6–8, 42].

Another option for narrowing the solution space for possible poses is to enforce smooth temporal transitions of individual body parts, such that they follow continous movement trajectories in three-dimensional space [6, 8, 42]. At a given time point a single pose is always correlated with the pose shortly before and after this particular time point. In fact, poses of two adjacent time points become identical for an infinitesimally small time difference. Thus, given a reasonably high sampling rate with respect to the recorded video data, the position of each reconstructed body part should not change with an arbitrary high degree, such that poses at consecutive time points remain similar. When reconstructing the pose of a given time point this principle allows for creating temporal dependencies of body part positions, e.g. by processing pose information from past and future time points. As a result, such an approach can increase pose reconstruction accuracy, e.g. when body parts are occasionally occluded [6, 8, 42].

While implementing anatomical and temporal constraints into a pose estimation framework is beneficial, it does not allow for assessing the probabilistic certainty of a reconstructed pose, i.e. its overall likelihood given the recorded video data. In fact, for longer behavioral sequences the occurrence of incorrectly estimated poses is expected at some point, e.g. due to measurement noise or shortcomings of the reconstruction algorithm itself. When the pose reconstruction algorithm is deterministic each reconstructed pose is a mere point estimate and the actual spectrum of theoretically possible poses remains unknown. An alternative to this is to model poses via a stochastic process, such that the underlying pose-encoding variables are drawn from a probability density function, e.g. a multivariate normal distribution [8]. In this case inferring the free parameters of the probability density function, e.g. the mean and covariance matrix of a multivariate normal distribution, is equivalent to reconstructing a whole range of poses, which are consistent

with the measured video data. Deploying such a probabilistic model for pose estimation offers the ability to assess the likelihood of a reconstructed pose [42], which causes probabilistic approaches to be more informative and therefore more versatile compared to their deterministic counterparts. For instance, having access to the likelihoods of reconstructed poses allows for excluding poses with low probabilistic certainty, i.e. low likelihoods, from further neuroscientific analyses, which is a useful asset whenever error tolerances are required to be minimal [42].

1.2 Related work

Endeavors to understand how animals change their body posture to move and interact with their environment date back over a century [45–47]. However, with the recent dawning of machine learning techniques, animal pose estimation capabilities have improved massively, whereas many of the achieved advances were fueled by innovations in the field of human pose estimation [7, 42].

1.2.1 Human pose estimation

A common approach in human pose estimation is to use a pseudo-skeleton model, which allows for describing a complex human pose in a rather low-dimensional space [40, 48]. For instance, the two-dimensional pose of a human has been reconstructed by modeling the individual body parts, i.e. the head, torso, arms and legs, via pictorial structures [49] even prior to the beginning of the so-called deep learning revolution in 2012 [50]. In this model formulation, the reconstruction of human poses relies on minimizing an energy function, which penalizes mismatches between reconstructed and observed poses as well as deformations of spring-like connections between individual body parts [51]. While combining the concept of pictorial structures with strong body part detectors led to further improvements regarding the reconstruction of human poses [52, 53], the focus of the research field shifted more to deep neural networks after their first utilization in this context [54]. Particularly, the availability of steadily increasing computational power in combination with large amounts of training data made convolutional neural networks (CNN) [55, 56] a popular tool used for estimating human poses. As a consequence of this development estimating body postures of multiple humans in images has been achieved by performing the detection of individual humans and the reconstruction of their poses in a joint operation instead of executing both computations independently from each other [57, 58].

Further improvements were made by not only focusing on the locations of individual body parts but also on their orientations via two-dimensional vector fields, so-called part affinity fields, which additionally indicate the direction in which a detected body part is pointing [59]. This allowed for not only reconstructing the body postures of multiple humans but also the pose of their feet, hands and faces in real time from raw image data [60, 61].

While the previously mentioned work on human pose estimation is primarily aimed at reconstructing two-dimensional poses based on images from a single camera, three-dimensional poses can be obtained via triangulation, when two-dimensional poses are accurately estimated from a set of images recorded with several synchronized and calibrated cameras. Alternatively, three-dimensional human poses have also been obtained by reconstructing the entire human shape, i.e. the visible surface area of a human, rather than only considering individual body parts or distinct anatomical features thereof [40]. Based on full three-dimensional body scans as well as motion capture data of three-dimensional surface marker trajectories, three-dimensional human poses and shapes have been reconstructed, while the resulting surface meshes were modeled as a function of an underlying three-dimensional skeleton model [62]. Subsequent improvements with respect to the estimation accuracy of three-dimensional human poses and shapes have been achieved via a skinned multi-person linear (SMPL) model, such that especially the reconstruction

quality of the visible surface area could be enhanced [63]. Here, the improved reconstruction capabilities were primarily fueled by increasing the number of full body scans to several thousands and additionally introducing linear dependencies between the reconstructed surface mesh and the rotation matrices, which determine the bone orientations of the underlying skeleton model. Combining this approach with a CNN-based two-dimensional human pose estimation framework even allowed for reconstructing three-dimensional human poses from a single image [64].

1.2.2 Animal pose estimation

Recent improvements in the field of human pose estimation were primarily driven by the existence of large data sets [7] containing up to several million annotated images of humans in different poses, contexts and environments, sometimes even accompanied by ground truth data on three-dimensional surface marker trajectories acquired via motion capture [65–67]. While efforts have been made to also generate large data sets for a range of different animal species [68–71], they are still comparably rare [42]. Furthermore, the wide variety in form and shape, which can be found within the animal realm, renders generating a universal data set for animal pose estimation a rather challenging task [7]. Additionally, this circumstance also complicates the direct deployment of tools developed for pose estimation in humans for the same task in animals without adjusting them accordingly [42].

A proposed solution for circumventing the issue of comparably small training data sets with respect to animal poses, is the usage of synthetically-generated data [72–74]. Nevertheless, even without relying on synthetic data, using human pose estimation schemes as a first starting point has proven to be effective to also enhance animal pose reconstructing capabilities [7, 42]. Particularly, the deployment of CNN architectures, which were originally designed for human pose estimation, in combination with so-called transfer learning improved the level of detail and accuracy to which animal poses can be reconstructed. In this context transfer learning describes the process of initially training a neural network on a task different from its final purpose and subsequently training it on the task it is actually intended for [7, 75]. Using this approach, reconstructing two-dimensional animal poses has been achieved based on comparably little training data [76]. While the resulting time spans needed for inferring respective poses are comparably long, it has been shown that they can be shortened at the cost of accuracy [77]. The resulting challenge of balancing inference speed against accuracy has been addressed as well [78], allowing for robust yet efficient two-dimensional animal pose estimation.

However, research on reconstructing three-dimensional animal poses has also been fruitful. For instance, equivalently to human pose estimation, surface markers have been tracked via motion capture to obtain three-dimensional animal poses [79, 80]. Besides, three-dimensional animal poses have been estimated by triangulating previously reconstructed two-dimensional poses from multiple images, which were recorded via different cameras at the same time point [81–83]. By introducing a CNN architecture in which the respective convolutions are also performed on a discrete three-dimensional grid rather than only being computed on the two-dimensional image domain, three-dimensional animal poses have also been inferred from multiple images directly, without the need for triangulation [69, 84]. Alternatively, three-dimensional animal poses have been reconstructed from only a single image by utilizing the SMPL model, while the necessity for obtaining full body scans of the animal subjects was replaced by scanning respective animal toy figures instead [85–89]. When ground truth data on the three-dimensional positions of body parts is available for initial training, reconstructing three-dimensional animal poses from single images has also been achieved without any additional requirements for full body or toy figure scans [90]. Furthermore, this approach also showed the practicability of so-called domain adaption, which allows for estimating three-dimensional poses of freely-moving animals, even when the corre-

sponding training data shows subjects of the respective animal species in a different context, e.g. a head-constrained setting.

While the techniques for reconstructing three-dimensional animal poses are versatile, many of them rely on standard video cameras to record data of behaving animals [47]. However, since camera-based approaches are prone to errors, e.g. due to occlusions of body parts, temporal as well as spatial constraints have been introduced to pose estimation frameworks to improve their accuracy [6–8, 42].

For instance, temporal constraints have been exploited in the context of three-dimensional animal pose estimation by penalizing large Euclidean distances between landmark locations of consecutive time points in the optimization procedures of respective pose reconstruction schemes [68, 82]. Besides, temporal constraints have been implemented into animal pose reconstruction frameworks indirectly via Bayesian filters [70], an approach which has also enabled classifying animal poses in real-time with zero latency [91]. Another technique for using temporal constraints within an animal pose estimation scheme is given by computing and utilizing the optical flow of heatmaps, which are generated as the output of a trained CNN and contain probabilistic certainty values for where different body parts are located in a two-dimensional image [83]. Furthermore, temporal constraints have also been implemented in the context of three-dimensional animal pose estimation based on sequences of two-dimensional images reorded at consecutive time points using temporal convolutions, which consider an additional time dimension instead of only operating on the two-dimensional image domain alone [92].

Spatial constraints have been deployed in animal pose estimation as well, either independently or alongside temporal constraints. For instance, pictorial structures were used to reconstruct the three-dimensional movement patterns and poses of flies [93]. Similarly, by enforcing the lengths of individual limbs or the distances between respective surface markers to be constant or nearly constant over time, plausible three-dimensional poses have been generated for larger animal species, i.e. cheetahs [70] and macaques [68], as well as smaller ones, i.e. mice and flies [82]. Furthermore, spatial constraints have also been implemented in the form of a kinematic chain, such that anatomical landmarks or surface markers on the fur of an animal subject are connected via a directed graph in three-dimensional space [70, 94]. This approach allowed for further boosting the quality of reconstructed three-dimensional animal poses, when compared to existing techniques based on triangulation or spatial three-dimensional convolutions [94].

1.3 Thesis goal and outline

So far, none of the existing techniques for reconstructing three-dimensional animal poses (Section 1.2.2) have considered estimating the positions of individual joints and bones underneath the visible body surface of a freely-moving animal. However, the anatomical skeleton of an animal imposes rigorous constraints to its movement capacities by introducing physiological motion limits. Thus, the skeleton determines which bone orientations are actually possible to reach and which are not. Consequently, modeling a realistic skeleton in the context of animal pose estimation is a promising approach for ensuring that reconstructed poses are anatomically-feasible and accurate (Section 1.1.3).

Therefore, the goal of this thesis is to develop a camera-based pose reconstruction algorithm for freely-moving animals, particular rats, to allow for quantitative analyses of animal behaviors in neuroscientific contexts, in which measurements of skeletal kinematics are a particular point of interest. A central aim in this regard is to constrain the resulting animal pose reconstruction framework based on realistic assumptions. Consequently, the developed pose estimation scheme contains an accurate skeleton model, which introduces anatomical constraints, and guarantees for reconstructed skeletal poses to be time consistent, which is realized by enforcing temporal

constraints. Furthermore, an additional aim with respect to the developed pose reconstruction framework is to tackle the challenges of animal pose estimation in a probabilistic manner to grasp the extent to which reconstructed poses are actually consistent with the recorded video data, which is used as an input to the framework.

After Chapter 1 gives the scientific motivation and background for this thesis (Section 1.1) as well as an overview of related work in the field (Section 1.2), Chapter 2 describes the methods, which were used to develop the proposed pose reconstruction framework. Thus, Chapter 2 starts with introducing the mathematical concepts for modeling three-dimensional rotations and video cameras (Section 2.1). Furthermore, a skeleton model is described, which allows for learning and reconstructing the skeletal anatomy of a freely-moving animal from recorded video data via gradient descent optimization, while anatomical constraints are taken into account (Section 2.2). In order to model the dynamics of skeletal pose changes over time a state space model is introduced, which adds temporal constraints and enables reconstructing skeletal poses in a probabilistic manner via a Bayesian smoother (Section 2.3). Finally, Chapter 2 concludes with a description of how the unknown probabilistic hyper-parameters of the previously introduced state space model are learned from video data via an expectation-maximization algorithm (Section 2.4).

The results of the work presented in this thesis are outlined in Chapter 3. To evaluate whether the proposed pose reconstruction framework is able to accurately infer skeleton anatomies, ground truth data on three-dimensional joint positions was obtained via magnetic resonance imaging and compared to learned skeleton anatomies (Section 3.1). Subsequently, skeletal poses were reconstructed based on recorded behavioral sequences of freely-moving animals and analyzed with respect to the reconstruction accuracy of the proposed pose estimation framework (Section 3.2). Here, a frustrated total internal reflection imaging system was used to obtain ground truth data on paw positions, which were then compared to their reconstructed counterparts. To assess if the proposed pose reconstruction framework allows for quantifying periodic gait motions, skeletal poses were also reconstructed based on recorded video data of animals, which were allowed to move freely in a spacious arena (Section 3.3). Subsequently, resulting reconstructed skeletal poses were used to extract and analyze cyclic gait patters. Finally, the proposed pose reconstruction framework was also deployed to quantify gap-crossing behaviors, where animals crossed gaps of varying lengths via jumping (Section 3.4). Here, reconstructed skeletal poses were used for characterizing distinct behavioral decision points, like the start-points of the jumps, and extracting correlations between kinematic quantities, e.g. individual joint velocities, and behavioral outcomes, i.e. jumped distances.

This thesis ends with Chapter 4, which contains conclusive remarks on the proposed pose reconstruction framework. Particularly, a brief summary of the previous contents as well as an assessment of the scientific value of this thesis is included (Section 4.1). Furthermore, this last chapter discusses the limitations (Section 4.2) and future potential (Section 4.3) of the proposed pose reconstruction framework within the scope of animal pose estimation and beyond.

Chapter 2

Methods

2.1 Rotations and cameras

The pose reconstruction framework proposed in this thesis relies on the ability to model three-dimensional rotations and cameras. Rotating an object in three-dimensional space is an essential mathematical operation in the context of animal pose estimation, since it allows for describing bone rotations. In fact, to store the entire three-dimensional configuration of an animal's bones and joints, i.e. its skeletal pose, only considering bone rotations suffices. Furthermore, three-dimensional rotations are also useful for modeling the orientations of cameras, which are crucial measuring devices for recording data from freely-behaving animals.

This section covers the theoretical background needed for parameterizing three-dimensional rotations (Section 2.1.1). Additionally, this section introduces the pinhole camera model (Section 2.1.2), which allows for approximating the process of how a camera captures an image. Finally, a numerical optimization scheme for calibrating a multi-camera setup is described in this section (Section 2.1.3), which allows for estimating the positions and orientations of different cameras with respect to each other.

2.1.1 Parameterizing rotations

In mathematical terms, a three-dimensional rotation is a linear transformation expressed via a matrix $R \in \mathbb{R}^{3\times3}$, where all columns of R are orthogonal to each other and its determinant equals one [95]. The orthogonality of R ensures that the overall shape of the rotated object is preserved while a determinant equal to one guarantees that the volume of the rotated object does not change [95]. However, while a three-dimensional rotation has only three degrees of freedom [95, 96], R has nine elements in total. Nevertheless, it is generally advantageous to only consider the actually existing three degrees of freedom of a three-dimensional rotation. For instance, when learning bone rotations via gradient descent optimization by minimizing a respective objective function. Here, optimizing all nine elements of R requires enforcing additional constraints to account for the shape- and volume-preserving properties of R, which is inferior to only optimizing the actually existing three degrees of freedom. Thus, an efficient way for parameterizing R is needed.

One option for parameterizing an arbitrary three-dimensional rotation R is given by Euler angles, which were first described by Leonhard Euler in the 18th century [97, 98]. Euler angles decompose R into a set of three independent rotations R_z, R_y and R_x, which describe rotations of an object around the z-, y- and x-axis respectively [95]. Applying these rotations consecutively yields a final rotation $R = R_x R_y R_z$ (Figure 2.1). Unfortunately, this parameterization has two major drawbacks. Different sets of Euler angles can lead to the same three-dimensional rotation, i.e. there does not necessarily exist a unique set for R_z, R_y and R_x to obtain R [95]. Additionally,

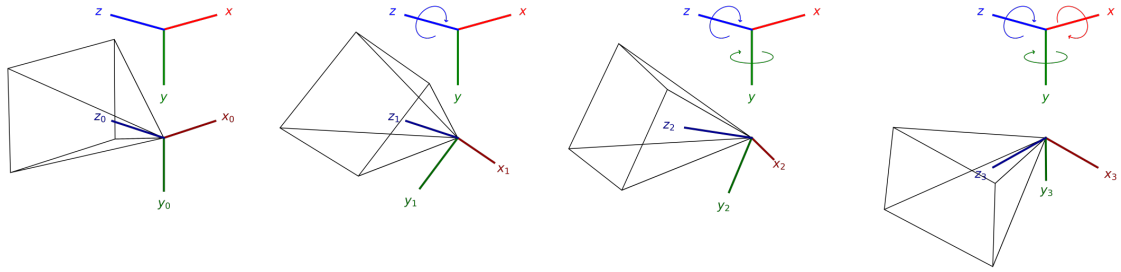

Figure 2.1: Illustrative example for how Euler angles are used to rotate an object, i.e. a camera, whose internal coordinate system (dim colors) is initially aligned with the world coordinate system (bright colors). Starting from the initial orientation (left), the camera is successively rotated 45 deg around the z-axis (center left), -20 deg around the y-axis (center right) and -45 deg around the x-axis (right). Each rotation is highlighted by an arrow associated with the world coordinate system, such that each arrow indicates the direction of a positive rotation.

specific sets of R_z, R_y and R_x can lead to a loss of rotational degrees of freedom. This situation is called gimbal lock and occurs when two of the rotational axes align [96, 99] (Figure 2.2).

Another possibility for parameterizing rotations is given by Rodrigues vectors, which are named after 18th-century-born Olinde Rodrigues [98, 100]. Using Rodrigues vectors to parameterize rotations eliminates the possibility for gimbal lock configurations, which is a key factor for why they are a suitable option for modeling bone rotations [96]. A Rodrigues vector r is given by a rotation axis $\omega \in \mathbb{R}^3$ and an associated rotation angle $\theta \in \mathbb{R}$, such that

$$r = \theta\omega = \theta(\omega_1, \omega_2, \omega_3)^T, \tag{2.1}$$

where $||\omega|| = 1$ (Figure 2.3). Given a Rodrigues vector r the corresponding rotation matrix R is calculated using function

$$\mathrm{f}_{\mathrm{r}\to\mathrm{R}}(r) = I + \hat{\omega}\sin(\theta) + \hat{\omega}^2(1 - \cos(\theta)) = R, \tag{2.2}$$

where $I \in \mathbb{R}^{3\times3}$ is the identity matrix and $\hat{\omega} \in \mathbb{R}^{3\times3}$ is a skew-symmetric matrix given by

$$\hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{2.3}$$

The Rodrigues vector parameterization is still ambiguous, since increasing the rotation angle θ by a multiple of 2π yields the same rotation matrix due to the trigonometric functions involved in

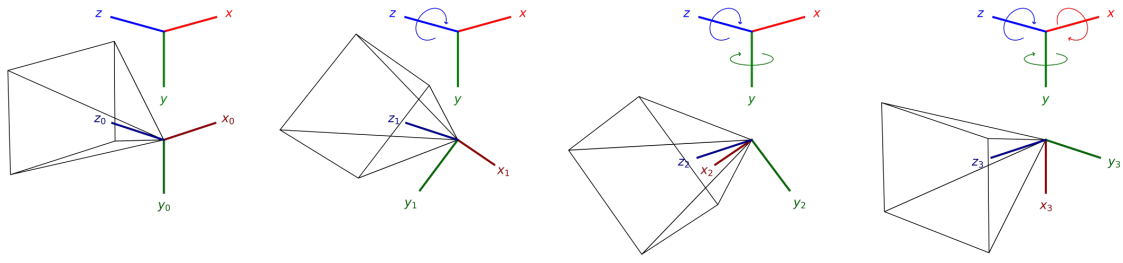

Figure 2.2: Illustrative example for how rotating an object with Euler angles leads to a gimbal lock configuration. The figure conventions and shown rotations are the same as in Figure 2.1, except that here the second rotation of the camera around the y-axis has a magnitude of -90 deg (center right). This causes a gimbal lock configuration, such that the first 45 deg rotation around the z-axis (center left) is negated by the third -45 deg rotation around the x-axis (right). Consequently, the final orientation could also be obtained by only applying the rotation around the y-axis.

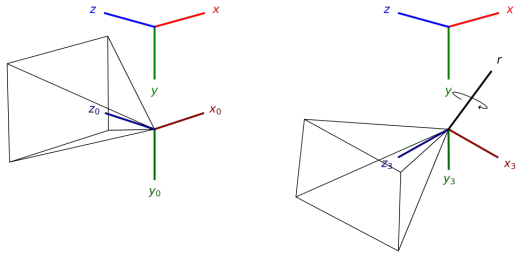

Figure 2.3: Illustrative example for how a Rodrigues vector r is used to rotate an object. Instead of applying three consecutive rotations around the world coordinate system, only a single rotation is performed. The axis of this single rotation is indicated by the Rodrigues vector itself, while its magnitude is encoded by the length of the vector. Note how this rotation results in the same final orientation as in Figure 2.1.

Equation 2.2, e.g. $\mathrm{f}_{\mathrm{r}\to\mathrm{R}}(\theta\omega) = \mathrm{f}_{\mathrm{r}\to\mathrm{R}}((\theta + 2\pi)\,\omega)$. Furthermore, a singularity exists for $\theta = 0$, since ω is not defined in this case [96].

A third option for expressing rotations, which solves the remaining issues of the Rodrigues vector parameterization, is given by parameterizing them via quaternions, which were first introduced by William Rowan Hamilton in the 19th century [101]. Quaternions are an extension of the complex numbers $\mathbb{C}$ to three-dimensional space, i.e. a single quanternion represents an individual point on a four-dimensional unit sphere [95]. As such, this parameterization is more challenging to interpret geometrically. Additionally, when it comes to parameterizing bone rotations for estimating animal poses, using quaternions complicates the implementation of joint angle limits.

The pose reconstruction framework proposed in this thesis contains physiological joint angle limits, which were measured as Euler angles in the physical world (Section 2.2.7). When bone rotations are parameterized via Euler angles or Rodrigues vectors it is possible to implement these measured joint angle limits as simple box constraints into the optimization scheme, which is used for reconstructing poses. Furthermore, the existing ambiguities of the Rodrigues vector parameterization did not cause any issues when poses were reconstructed with the proposed pose reconstruction framework (Chapter 3). Thus, for the scope of this thesis Rodrigues vectors are the chosen parameterization for three-dimensional rotations.

2.1.2 The pinhole camera model

To reconstruct poses from a freely-moving animal, its body posture has to be recorded using video cameras. Recording a single image with a camera is equivalent to projecting a three-dimensional scene onto the camera's two-dimensional image plane and storing the therefore generated image. This process is approximated using the perspective projection, which is part of the pinhole camera model [102]. The perspective projection relates a three-dimensional point $m_{3\mathrm{D}} \in \mathbb{R}^3$ in space to the corresponding two-dimensional point $m_{2\mathrm{D}} \in \mathbb{R}^2$ on the camera's image plane:

$$m_{2\mathrm{D}} = \tilde{A}\left(\mathrm{f}_{\mathrm{norm}}\left(\tilde{x}\right)\right) = \tilde{A}\left(\mathrm{f}_{\mathrm{norm}}\left(\mathrm{f}_{\mathrm{r}\to\mathrm{R}}\left(\tilde{r}\right) m_{3\mathrm{D}} + \tilde{t}\right)\right). \tag{2.4}$$

Here, $\tilde{r} \in \mathbb{R}^3$ is a Rodrigues vector and $\tilde{t} \in \mathbb{R}^3$ is a translation vector, which together determine the orientation and location of the camera in three-dimensional space. The expression $\tilde{x} = \mathrm{f}_{\mathrm{r}\to\mathrm{R}}\left(\tilde{r}\right) m_{3\mathrm{D}} + \tilde{t}$ maps $m_{3\mathrm{D}}$ from the world coordinate system to the coordinate system of the camera. The projection on the camera's image plane at distance 1 in the camera's z-direction is then performed via function $\mathrm{f}_{\mathrm{norm}}\left(\tilde{x}\right) = \left(\frac{\tilde{x}_1}{\tilde{x}_3}, \frac{\tilde{x}_2}{\tilde{x}_3}, 1\right)^T$. However, since the image plane of a camera is actually located at a distances equal to the camera's focal lengths $\tilde{A}_{11}$ and $\tilde{A}_{22}$, it is necessary to perform an additional scaling operation. Furthermore, a correction term is needed to translate

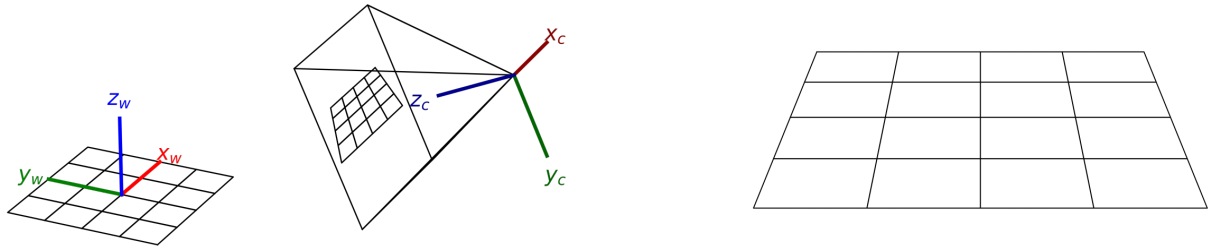

Figure 2.4: Illustrative example for how a regular grid is mapped to the image plane of a camera via the perspective projection (left) and for how the resulting projected grid visually appears in an image (right). Note how grid lines remain straight and corner points close to the camera appear further apart compared to points at greater distances, when examining the grid in the image plane.

the origin of the coordinate system of the image from the camera's optical center $\left(\tilde{A}_{11}, \tilde{A}_{22}\right)$ to the upper left corner of the image. Both operations, scaling and translating, are performed at once using the camera matrix $\tilde{A} \in \mathbb{R}^{2\times 3}$, which is given by

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{11} & 0 & \tilde{A}_{13} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \end{pmatrix}. \tag{2.5}$$

A distinctive characteristic of the resulting projection is given by the fact that it causes straight lines in a three-dimensional scene to also appear straight in a recorded image of that scene (Figure 2.4).

However, in practice cameras do not necessarily perform this ideal projection and apply distortions to a recorded image. One possible form of distortions are radial distortions, which cause straight lines to appear skewed in the recorded image (Figure 2.5). In mathematical terms, radial distortions are described via a distortion function

$$\mathrm{f}_{\mathrm{distort}}\left(\bar{x}, \tilde{k}\right) = \begin{pmatrix} \bar{x}_1 \left(1 + \tilde{k}_1 s + \tilde{k}_2 s^2\right) \\ \bar{x}_2 \left(1 + \tilde{k}_1 s + \tilde{k}_2 s^2\right) \\ 1 \end{pmatrix}, \tag{2.6}$$

where $\bar{x} = \mathrm{f}_{\mathrm{norm}}(\tilde{x})$, $\tilde{k} = (k_1, k_2)^T$ and $s = \bar{x}_1^2 + \bar{x}_2^2$ [102]. Here, $\tilde{k} \in \mathbb{R}^2$ is a distortion vector, which stores the radial distortion coefficients k_1 and k_2 of the camera. In principle, there exist

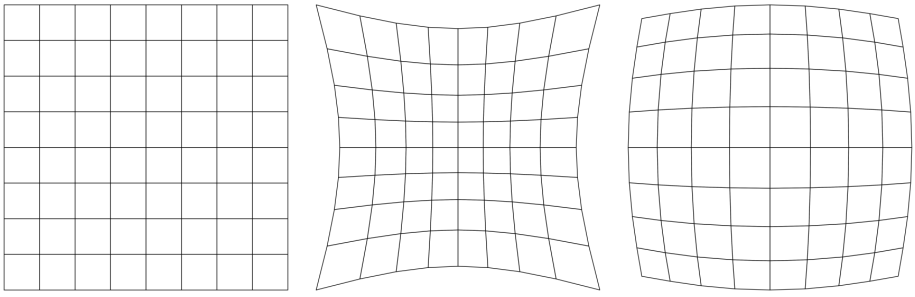

Figure 2.5: Illustrative example for how radial distortions affect the appearance of a projected regular grid in an image. Without distortions the grid appears regular, such that all lines are straight and parallel to each other (left). When radial distortions are applied, the previously straight lines appear curved and are not longer parallel to each other. Here, the distortion coefficients are $k_1 = 10^{-2}$ and $k_2 = 10^{-4}$ (center) as well as $k_1 = -10^{-2}$ and $k_2 = 10^{-4}$ (right).

further kinds of distortions, e.g. tangential or higher-order radial distortions, but considering only two radial distortion coefficients already yields reasonable results when reconstructing poses with the proposed pose reconstruction framework (Chapter 3).

Combining the perspective projection with radial distortions allows for mapping an arbitrary three-dimensional point m_{3D} in space to the corresponding two-dimensional point in a recorded image using function

$$\mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}\left(m_{\mathrm{3D}}, \tilde{r}, \tilde{t}, \tilde{k}, \tilde{A}\right) = \tilde{A}\,\mathrm{f}_{\mathrm{distort}}\left(\mathrm{f}_{\mathrm{norm}}\left(\mathrm{f}_{\mathrm{r}\to\mathrm{R}}\left(\tilde{r}\right) m_{\mathrm{3D}} + \tilde{t}\right), \tilde{k}\right). \tag{2.7}$$

Therefore, each camera has four model parameters: $\tilde{r}$ and $\tilde{t}$, which are called the intrinsic parameters, and $\tilde{A}$ and $\tilde{k}$, which are called the extrinsic parameters [102]. Consequently, the intrinsic and extrinsic parameters need to be estimated before using function $\mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}$, which can be achieved by calibrating the camera.

2.1.3 Multi-camera calibration

Function $\mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}$ (Equation 2.7) is primarily used to predict where a point in three-dimensional space will be mapped on the two-dimensional image plane of a camera. When additional numerical optimization steps are performed, function $\mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}$ can also be used to triangulate the unknown three-dimensional position of such a point from a given set of corresponding two-dimensional points located on the image planes of multiple different cameras. In principle, this process already allows for recovering an animal's three-dimensional pose (Section 2.2.3). To obtain matching two-dimensional points, it is necessary to record a scene with several cameras at the same time point using a multi-camera setup. To subsequently deploy function $\mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}$ full information about all intrinsic and extrinsic parameters of the cameras in the setup is required. Globally calibrating all cameras in the setup via gradient descent optimization gives this information.

To perform a calibration of a multi-camera setup it is possible to leverage the known structures and dimensions of a physical object. A common example for such an object is a checkerboard, which is a frequently used item in this context. The main advantage of a checkerboard is its regular pattern, which allows for the automated detection of the n_{corner} corners of the checkerboard's quadratic tiles [102]. Additionally, in the context of camera calibration, the only essential parameter of a checkerboard is the length of a quadratic tile l_{tile}. To increase the detection accuracy checkerboards are combined with ArUco markers [103], which gives so-called ChArUco boards [102]. Using ChArUco boards allows for generating data sets for calibrating multiple cameras by recording an image sequence in which the board has to be visible in several cameras, whose three-dimensional positions and orientations differ from each other (Figure 2.6).

Given a data set containing images from the n_{cam} cameras of the setup reordered at n_{time} individual time points, a respective objective function is minimized via gradient descent optimization to learn the intrinsic and extrinsic parameters of all cameras. In the context of the proposed pose reconstruction framework this optimization problem is given by

$$\underset{\substack{\tilde{r}_i,\tilde{t}_i,\tilde{k}_i,\tilde{A}_i,\hat{r}_\tau,\hat{t}_\tau \\ \forall i\in\{1,\ldots,n_{\mathrm{cam}}\} \\ \forall \tau\in\{1,\ldots,n_{\mathrm{time}}\}}}{\arg\min} \sum_{\tau=1}^{n_{\mathrm{time}}} \sum_{i=1}^{n_{\mathrm{cam}}} \sum_{j=1}^{n_{\mathrm{corner}}} \delta_{\tau ij} \left|\left| \bar{c}_{\tau ij} - \mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}\left(\mathrm{f}_{\mathrm{r}\to\mathrm{R}}\left(\hat{r}_\tau\right)\hat{m}_j + \hat{t}_\tau, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i\right)\right|\right|^2, \tag{2.8}$$

where gradients are computed automatically via an auto-differentiation library [104] and the minimization is performed using an implementation of the Trust Region Reflective algorithm [105, 106]. In Equation 2.8 $\tilde{A}_i$ and $\tilde{k}_i$ are the intrinsic and $\tilde{r}_i$ and $\tilde{t}_i$ the extrinsic parameters of camera i. The orientation and position of the ChArUco board at time point τ is represented by the Rodrigues vector $\hat{r}_\tau$ and the translation vector $\hat{t}_\tau$ respectively. An automatically detected checkerboard corner j

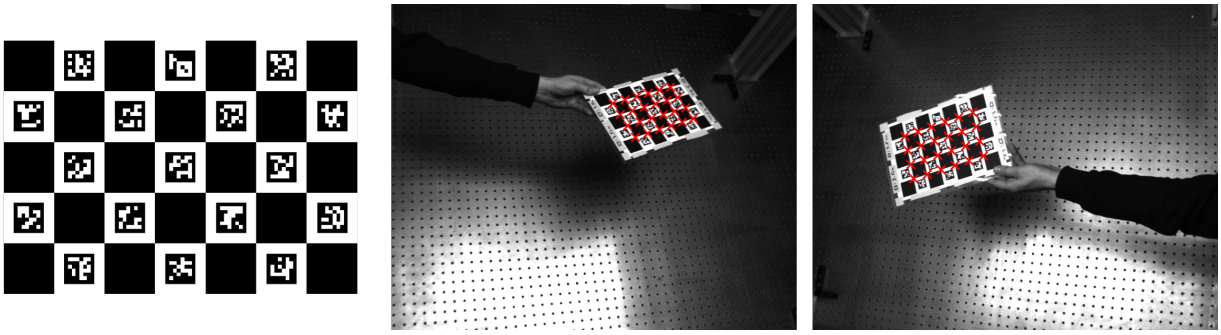

Figure 2.6: Example of a ChArUco board pattern (left) and two images recorded at the same time point to calibrate a multi-camera setup (center, right). Automatically detected corners are highlighted (red crosses). Note how using multiple cameras allows for recording the ChArUco board from different viewing angles.

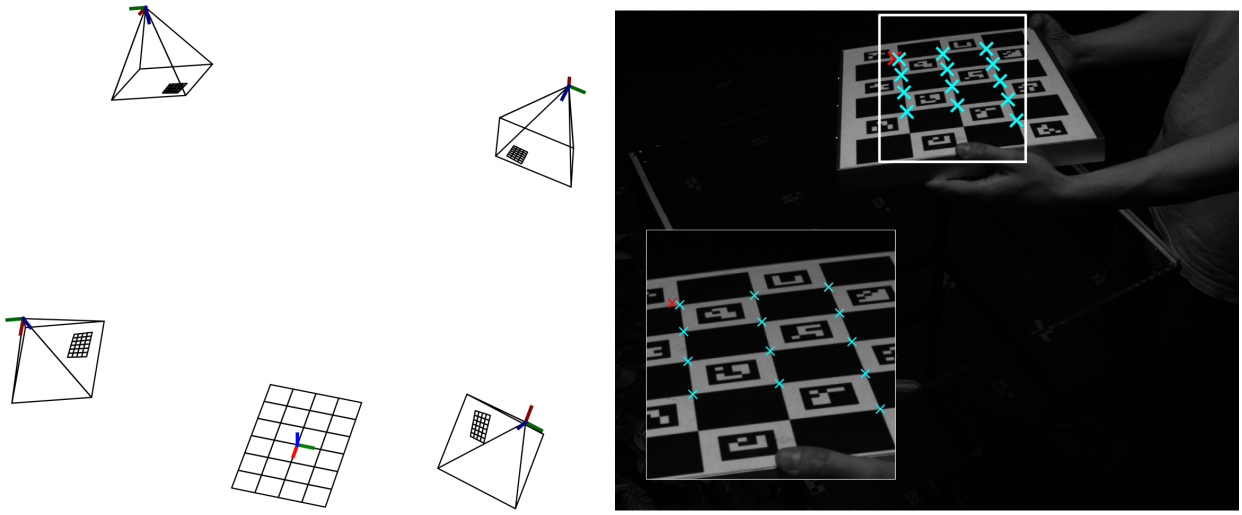

Figure 2.7: Example for how calibrating a multi-camera setup gives full information about all camera positions and orientations, which allows for reconstructing the setup in three-dimensional space (left). The respective calibration is obtained by minimizing the error between automatically detected (red crosses) and projected (cyan crosses) corner positions (right). Note how the calibration compensates for an incorrectly detected corner position (right inset).

in camera i at time point τ is denoted as $\bar{c}_{\tau ij} \in \mathbb{R}^2$ and $\hat{m}_j = l_{\text{tile}} \, (a_j, b_j, 0)^T \in \mathbb{R}^3$ represents the known planar structure of the ChArUco board, such that $a_j \in \mathbb{N}$ and $b_j \in \mathbb{N}$. To account for occluded checkerboard corners the objective function contains a delta function $\delta_{\tau ij}$, which indicates whether in camera i a corner j at time point τ is successfully detected, i.e. $\delta_{\tau ij} = 1$, or not, i.e. $\delta_{\tau ij} = 0$. After successfully calibrating a multi-camera setup discrepancies between automatically detected and estimated two-dimensional corner positions are minimized and all intrinsic and extrinsic camera parameters are known (Figure 2.7), which is a prerequisite for estimating animal poses.

2.2 Skeleton-based pose estimation

The previous section introduced mathematical concepts for parameterizing three-dimensional rotations and discussed how they are used in different contexts, e.g. to model the orientations of cameras or bone rotations (Section 2.1.1). Additionally, the previous section introduced the pinhole camera model, which allows for modeling how an image is captured by a video camera (Section 2.1.2). Finally, the previous section also introduced a method for calibrating a multi-camera setup

in order to determine the internal and external parameters of each camera in the setup (Section 2.1.3).

However, reconstructing poses of a freely-moving animal requires an additional model, which describes the skeleton of the animal, such that the configuration of modeled joints and bones can serve as a proxy for the animal's body posture. Furthermore, such a skeleton model should allow for simulating individual limb movements by dynamically changing bone rotations to describe poses of animals in motion. Therefore, this section introduces a respective skeleton model (Section 2.2.1) and states how the modeled bones and joints are linked to observable surface markers, which are located on an animal's body surface (Section 2.2.2). Additionally, this section describes how unknown skeleton parameters, like bone lengths, are learned from two-dimensional image data via numerical optimization (Section 2.2.3). To improve the respective optimization scheme, this section also introduces a variety of enforceable constraints, e.g. body symmetry or physiological joint angle limits (Section 2.2.4 to 2.2.8), and finally discusses how they are implemented in the proposed pose estimation framework (Section 2.2.9).

2.2.1 The skeleton model

The pose of an animal at a single time point is defined as the overall appearance of its body. This includes, for instance, the placement of individual limbs or the orientation of the animal's head. The limiting and therefore governing factor for the possible motion sequences, which determine how an animal can deform its own body and how it can change its body posture, is given by the rigidity of the animal's underlying skeleton (Section 1.1.3). Thus, the proposed pose reconstruction framework deploys the underlying skeletal configuration of bones and joints as a proxy for the overall body posture of an animal. Consequently, an animal's body pose is represented via a skeleton model, which approximates the animal's physiological joints and bones and allows for reconstructing their positions in three-dimensional space.

The mathematical concept of a graph [107] allows for approximating a physiological skeleton. In this case non-leaf vertices represent joints, leaf vertices represent anatomical features on the animal's body surface and edges represent bones (Figure 2.8). This skeleton graph is directed, such that it starts at the root vertex, i.e. the animal's snout, and ends at the leaf vertices, e.g. the animal's fingers. The connectivity of the skeleton graph, determining which vertices are connected to each other, is constant, since it is given by the known anatomy of the animal species of interest, e.g. shoulder, elbow and wrist joints are sequentially connected. However, the three-dimensional configuration of the vertices and edges is variable: either by globally translating the entire skeleton graph or by locally rotating individual edges. Since the rotation of a single edge does not only affect the edge itself but also all child-vertices and their associated edges, the skeleton graph forms a kinematic chain [108]. Since the vertices and edges approximate the animal's physiological joints and bones, using such a kinematic chain allows for representing different poses of a moving animal.

A respective kinematic chain is constructed using n_{bone} bones, such that each bone j has an associated local coordinate system $R_j \in \mathbb{R}^{3\times3}$, an associated bone length $l_j \in \mathbb{R}^3$ and two associated joint positions $p_{j_0} \in \mathbb{R}^3$ and $p_{j_1} \in \mathbb{R}^3$. The local coordinate system R_j determines the bone orientation $e_j = \left(R_{j13}, R_{j23}, R_{j33}\right)^T$, specifying in which three-dimensional direction the bone is facing. The associated bone length l_j together with the associated joint positions p_{j_0} and p_{j_1} determine the positions of the start- and end-point of the bone, such that $p_{j_1} = p_{j_0} + e_j\ l_j$. Consequently, $p_{j_0} \in \mathbb{R}^3$ is denoting the start-point of the bone, whereas $p_{j_1} \in \mathbb{R}^3$ denotes its end-point. To encode the skeletal pose of an animal it is therefore sufficient to consider the three-dimensional location of the root joint, given by a translation vector $t \in \mathbb{R}^3$, as well as the bone rotations and lengths, given by a set of Rodrigues vectors $r \in \mathbb{R}^{n_{\mathrm{bone}}\times3}$ and the vector

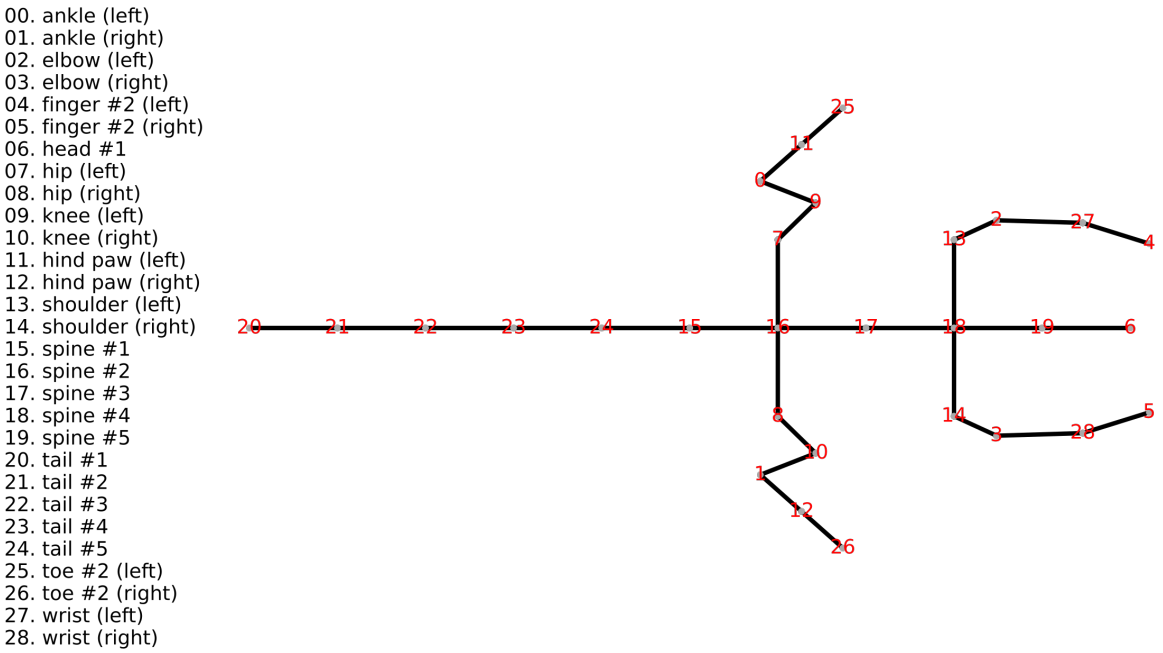

Figure 2.8: Illustration of the skeleton graph used in the proposed pose reconstruction framework to approximate the three-dimensional joint and bone positions of an animal. The skeleton graph is directed, i.e. it starts at the root vertex (head #1) and ends at the leaf vertices (fingers, toes and tail #1). All vertices approximate anatomical joint locations, except the root and leaf vertices, which represent anatomical landmarks on the animal's body surface.

$l \in \mathbb{R}^{n_{\text{bone}}}$ respectively. The resulting local coordinate systems $R \in \mathbb{R}^{n_{\text{bone}} \times 3 \times 3}$ as well as the three-dimensional start- and end-points $p \in \mathbb{R}^{n_{\text{bone}} \times 2 \times 3}$ of all bones are then calculated according to Algorithm 1.

Algorithm 1 requires a predefined and constant resting pose $R_0 \in \mathbb{R}^{n_{\text{bone}} \times 3 \times 3}$, which stores the local coordinate systems of all bones, when no bone rotations are present, i.e. $r_j = (0, 0, 0)\ \forall\ j \in \{1, ..., n_{\text{bone}}\}$. Furthermore, the definition of Algorithm 1 is based on the assumption that the set $\{1, ..., n_{\text{bone}}\}$ stores the sorted bone indices, such that an iteration through the set starts at the root joint and ends at the leaf joints. Consequently, index 1 is associated with the bone, whose start-point is the root joint, and index n_{bone} is associated with a bone, whose end-point is a leaf joint. The transpose operation used in Algorithm 1 for updating downstream bone rotations allows for iterating through the kinematic chain from the root joint to the leaf joints, while the bone rotations are actually carried out in reversed order [109]. For instance, given individual bone rotations around the shoulder, elbow and wrist joints, i.e. R_{shoulder}, R_{elbow} and R_{wrist} respectively, the new local coordinate system R^*_{wrist} of the bone, whose start-point is the wrist joint and whose resting pose is given by $R_{0_{\text{wrist}}}$, is calculated as follows:

$$R^*_{\text{wrist}} = \left(R_{\text{wrist}}{}^T R_{\text{elbow}}{}^T R_{\text{shoulder}}{}^T\right)^T R_{0_{\text{wrist}}} = R_{\text{shoulder}} R_{\text{elbow}} R_{\text{wrist}} R_{0_{\text{wrist}}}. \tag{2.9}$$

2.2.2 Modeling surface markers

Given a kinematic chain representing the skeleton of an animal, function f_{pose} (Algorithm 1) allows for obtaining the skeletal pose of an animal by computing the local coordinate systems R of the bones as well as the three-dimensional joint locations p. However, observing the actual joint positions of a freely-moving animal using video cameras is not feasible, since only the body surface of the animal is visible. Nonetheless, to relate the underlying skeleton to the actually observable body surface, it is possible to use surface markers painted onto the animal's body (Figure 2.9).

Algorithm 1: Computing joint locations via the skeleton model.

function $\mathrm{f}_{\mathrm{pose}}(t, r, l)$
  for $i \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
    $R_i \leftarrow I$ ▷ Initialize bone rotations
  for $i \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
    $R_s \leftarrow \mathrm{f}_{\mathrm{r}\to\mathrm{R}}(r_i)$ ▷ Calculate bone rotation R_s
    for $j \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
      if p_{j_1} is child of p_{i_0} **then** ▷ Find bones affected by rotation R_s
        $R_j \leftarrow {R_s}^T R_j$ ▷ Update all downstream bone rotations
  for $j \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
    $R_j \leftarrow {R_j}^T R_{0j}$ ▷ Apply bone rotations to resting pose R_0
  $p_{1_0} \leftarrow t$ ▷ Initialize root joint location p_{1_0}
  for $i \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
    $e_i \leftarrow (R_{i13}, R_{i23}, R_{i33})^T$ ▷ Obtain bone orientation e_i
    $p_{i_1} \leftarrow p_{i_0} + e_i\, l_i$ ▷ Calculate bone end-point p_{i_1}
    for $j \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
      if p_{i_1} is start-point of bone j **then** ▷ Find upcoming downstream bone
        $p_{j_0} \leftarrow p_{i_1}$ ▷ Initialize bone start-point p_{j_0}
  return p, R

In the kinematic chain this relationship is modeled by adding n_{marker} surface markers and rigidly attaching them to the joints. A connection between a single marker k and its associated joint is given by a respective joint-to-marker-translation vector v_k, whose orientation is altered by rotating bones. When the joint-to-marker-translation vectors $v \in \mathbb{R}^{n_{\mathrm{marker}}\times 3}$ for all surface markers are provided, the three-dimensional surface marker positions $m \in \mathbb{R}^{n_{\mathrm{marker}}\times 3}$ are calculated according to Algorithm 2.

Given the three-dimensional marker position $m_{\tau j} = \mathrm{f}_{\mathrm{surface}}(t_\tau, r_\tau, l, v)_j$ of a single marker j for a specific time point τ, the projected two-dimensional marker location $\tilde{m}_{\tau ij} \in \mathbb{R}^2$ in camera i is

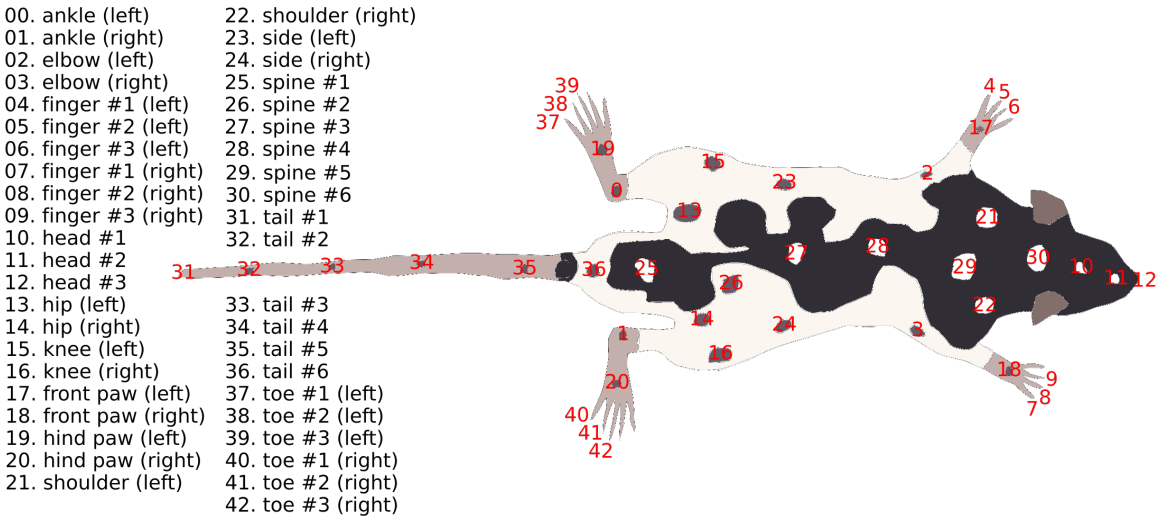

Figure 2.9: Illustration of the symmetric surface marker pattern painted onto the bodies of different animal subjects. In the used skeleton model each surface marker is rigidly attached to a joint, which links observable surface marker locations to underlying joint positions.

Algorithm 2: Computing surface marker locations from joint positions.

function $\mathrm{f}_{\mathrm{surface}}(t, r, l, v)$
  $p, R \leftarrow \mathrm{f}_{\mathrm{pose}}(t, r, l)$ ▷ Modify skeletal pose
  for $k \in \{1, ..., n_{\mathrm{marker}}\}$ **do**
    for $j \in \{1, ..., n_{\mathrm{bone}}\}$ **do**
      if m_k is attached to p_{j_1} **then** ▷ Find joint attached to marker
        $m_k \leftarrow p_{j_1} + R_j v_k$ ▷ Calculate surface marker positions
  return m

computed by propagating $m_{\tau j}$ through the projection function $\mathrm{f}_{\mathrm{3D}\rightarrow\mathrm{2D}}$ (Equation 2.7):

$$\tilde{m}_{\tau i j} = \mathrm{f}_{\mathrm{3D}\rightarrow\mathrm{2D}}\left(m_{\tau j}, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i\right). \tag{2.10}$$

Thus, Equation 2.10 relates underlying joints to observable surface markers, which, in principle, already allows for reconstructing poses (Section 2.2.3).

2.2.3 Anatomy learning

For a single time point τ Equation 2.10 establishes a direct mapping from the pose-encoding parameters, i.e. the translation vector t_τ and the bone rotations r_τ, and the anatomy-encoding parameters, i.e. the bone lengths l and the joint-to-marker-translation vectors v, to the two-dimensional location $\tilde{m}_{\tau i j}$ of surface marker j in camera i. However, initially the pose- and anatomy-encoding parameters are unknown. Nevertheless, it is possible to learn them from ground truth surface marker locations $\bar{m}_{\tau i j}$ via gradient descent optimization, since all mathematical operations involved in computing the projected two-dimensional surface marker location $\tilde{m}_{\tau i j}$ (Equation 2.10) are fully differentiable with respect to the pose- and anatomy-encoding parameters. The ground truth surface marker locations $\bar{m}_{\tau i j}$ are generated by manually labeling a behavioral sequence consisting of images recorded at n_{time} different time points. Similarly to how a multi-camera setup is calibrated (Section 2.1.3), the pose- and anatomy-encoding parameters are learned by minimizing a respective objective function given by

$$\underset{\substack{t_\tau, r_\tau, l, v \\ \forall \tau \in \{1,\ldots,n_{\mathrm{time}}\}}}{\arg\min} \sum_{\tau=1}^{n_{\mathrm{time}}} \sum_{i=1}^{n_{\mathrm{cam}}} \sum_{j=1}^{n_{\mathrm{marker}}} \delta_{\tau i j} ||\bar{m}_{\tau i j} - \tilde{m}_{\tau i j}||^2. \tag{2.11}$$

However, to obtain reasonable results for the anatomy- and pose-encoding variables it is not necessarily sufficient to minimize the objective function given by Equation 2.11, since the respective solution space is rather large. Additionally, labeling the two-dimensional marker locations in all cameras for each individual time point of a respective training sequence is a labor-intensive process, such that the resulting training data set is typically small. Consequently, it is necessary to constrain the solution space with respect to t_τ, r_τ, l and v within the minimization scheme given by Equation 2.11. Such constraints are implemented by incorporating prior knowledge about an animal's anatomy (Section 2.2.4 and 2.2.5), the applied surface marker pattern on the animal's body (Section 2.2.6) as well as physiological joint angle limits (Section 2.2.7).

2.2.4 Enforcing body symmetry

Constraining the unknown bone lengths l and the joint-to-marker-translation vectors v narrows the space for possible solutions, when these variables are learned for a specific animal subject by

Algorithm 3: Computing body-symmetric bone lengths.

function $\mathrm{f}_{l^* \to l}(l^*)$
  $i_l \leftarrow 1$ ▷ Initialize counter for right-sided bones
  for $i \in \{1, ..., n^*_{\mathrm{bone}}\}$ **do**
    $l_i \leftarrow l^*_i$ ▷ Set value for left-sided or central bone
    if i is left-sided bone **then** ▷ Check if bone is left-sided
      $l_{n^*_{\mathrm{bone}}+i_l} \leftarrow l^*_i$ ▷ Copy value for right-sided bone
      $i_l \leftarrow i_l + 1$ ▷ Increase counter for right-sided bones
  return l

Algorithm 4: Computing body-symmetric joint-to-marker-translation vectors.

function $\mathrm{f}_{v^* \to v}(v^*)$
  $i_v \leftarrow 1$ ▷ Initialize counter for right-sided markers
  for $j \in \{1, ..., n^*_{\mathrm{marker}}\}$ **do**
    $v_j \leftarrow v^*_j$ ▷ Set value for left-sided or central marker
    if j is left-sided marker **then** ▷ Check if marker is left-sided
      $v_{n^*_{\mathrm{marker}}+i_v} \leftarrow \left(-v^*_{j\,1}, v^*_{j\,2}, v^*_{j\,3}\right)^T$ ▷ Copy value for right-sided marker
      $i_v \leftarrow i_v + 1$ ▷ Increase counter for right-sided markers
  return v

minimizing Equation 2.11. Particularly, using constraints, which exploit the natural symmetry of an animal's body, reduces the total number of free parameters, which need to be learned. For instance, when assuming that a left-sided bone, e.g. the left humerus, has the same length as the corresponding right-sided bone, i.e. the right humerus, only a single parameter needs to be learned to estimate the lengths of both bones.

Following this approach, the number of free parameters in Equation 2.11 is decreased by introducing the reduced bone lengths $l^* \in \mathbb{R}^{n^*_{\mathrm{bone}}}$. Here, n^*_{bone} denotes the number of left-sided bones plus the number of central bones, which are neither left- nor right-sided, e.g. the bones used to model an animal's spine. Given the normal bone lengths l, the reduced bone lengths l^* are obtained according to Algorithm 3. The definition of Algorithm 3 is based on the assumption that the set $\{1, ..., n^*_{\mathrm{bone}}\}$ stores the bone indices, such that the indices of the central and left-sided bones are always smaller than those of the right-sided bones.

The number of free parameters with respect to the joint-to-marker-translation vectors v are reduced as well, since the surface markers are painted onto an animal's body in a systematic pattern, which reflects the symmetry of the animal's body (Section 2.2.2). For instance, a surface marker i placed closely to the left knee joint has a corresponding surface marker j placed closely to the right knee joint. This symmetric placement allows for obtaining the x-component of the corresponding right-sided joint-to-marker-translation vector v_j by mirroring the x-component of the left-sided joint-to-marker-translation vector v_i, i.e. $v_{j_1} = -v_{i_1}$. Thus, the reduced joint-to-marker-translation vectors $v^* \in \mathbb{R}^{n^*_{\mathrm{marker}} \times 3}$ are introduced, where n^*_{marker} denotes the number of the left-sided surface markers plus the number of central surface markers, which are neither left- nor right-sided. Given the normal joint-to-marker-translation vectors v, the reduced joint-to-marker-translation vectors v^* are obtained according to Algorithm 4. The definition of Algorithm 4 is based on the assumption the set $\{1, ..., n^*_{\mathrm{marker}}\}$ stores the surface marker indices, such that the indices of the central and left-sided surface markers are always smaller than those of the right-sided surface markers.

To learn a symmetric skeletal anatomy the reduced variables l^* and v^* are used. The computations for obtaining the projected surface marker locations $\tilde{m}_{\tau ij}$, which are needed in Equation 2.10,

are given by:

$$\tilde{m}_{\tau ij} = \mathrm{f}_{\mathrm{3D}\to\mathrm{2D}}\left(\mathrm{f}_{\mathrm{surface}}\left(t, r, \mathrm{f}_{\mathrm{l}^*\to\mathrm{l}}\left(l^*\right), \mathrm{f}_{\mathrm{v}^*\to\mathrm{v}}\left(v^*\right)\right)_{\tau j}, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i\right). \tag{2.12}$$

2.2.5 Constraining bone lengths

When learning the reduced bone lengths l^* the respective solution space is narrowed by enforcing box constraints on the individual elements of l^*. These box constraints are based on an allometric study, which investigated correlations between body weights and bone lengths of different animal species [110]. The study found roughly linear relationships between both quantities, such that the weight of an individual animal subject gives insights into the lengths of its bones. These linear relationships are incorporated into the optimization scheme given by Equation 2.11, such that the resulting learned bone lengths are enforced to stay within predefined physiological ranges.

Given the body weight m_{subject} of an animal subject of a specific species – *rattus norvegicus* in the context of the proposed pose reconstruction framework – the upper and lower physiological bound for the length of a limb bone is calculated using the expectation value and the standard deviation of the slope parameter, which defines the linear relationship between the body weight and the limb bone (Table 2.1). Particularity, for each limb bone the respective box constraint is defined as

$$\left[m_{\mathrm{subject}}\left(\mu_{\mathrm{slope}} - \sigma_{\mathrm{slope}} s_\sigma\right), m_{\mathrm{subject}}\left(\mu_{\mathrm{slope}} + \sigma_{\mathrm{slope}} s_\sigma\right)\right], \tag{2.13}$$

with the expectation value μ_{slope}, the standard deviation σ_{slope} and a scalar factor s_σ.

In the proposed pose reconstruction framework s_σ is set to a relatively large value, i.e. $s_\sigma = 10$, which ensures that the solution space for l^* remains broad enough to take into account the possibility of individual outlier subjects, whose bone lengths do not precisely follow the linear relationship. Furthermore, box constraints are set to $[0, \mathrm{inf})$ when the respective bones are not part of the limbs, since linear relationships between body weight and non-limb bones are not provided by the study.

2.2.6 Constraining surface marker positions

In Section 2.2.4 the reduced joint-to-marker-translation vectors v^* are introduced to account for a symmetric surface marker pattern, which reduces the number of free parameters in the optimization scheme given by Equation 2.11. Equivalently, it is possible to enforce respective spatial constraints for v^*, which further narrows the solution space, when learning an animal's anatomy.

bone	start-joint	end-joint	avg. slope (cm/kg)	s.d. (cm/kg)
humerus	shoulder	elbow	7.5	0.5
radius	elbow	wrist	6.9	0.4
metacarpal	wrist	finger	2.3	0.1
femur	hip	knee	10.2	0.6
tibia	knee	ankle	14.4	0.6
metatarsal	ankle	hind paw	5.3	0.3

Table 2.1: Expectation values and standard deviations of slope parameters for the limb bones of *rattus norvegicus* taken from a respective study [110]. The slope parameters describe linear relationships between an animal's body weight and its bone lengths. This allows for calculating maximum and minimum bone lengths based on the weight of an animal subject. When the anatomy of an animal subject is learned via gradient descent optimization (Section 2.2.3), respective constraints on the maximum and minimum bone lengths are enforced.

Particularly, incorporating prior knowledge about the arrangement of surface markers on an animal's body forces the individual elements of v^* to stay within predefined intervals during the optimization procedure (Table 2.2). For instance, when surface markers are painted on an animal's body along its main axis, i.e. the axis pointing from the head to the tail, the learned marker positions are forced to stay in the plane spanned by this main axis and an additional orthogonal axis, which points in the upward direction. In this concrete example the number of free parameters per surface marker is decreased from three to two. Thus, using this approach further reduces the number of free parameters in the optimization scheme given by Equation 2.11.

surface marker	connected joint	x	y	z
head #1	spine #5	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
head #2	spine #5	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
head #3	head (root)	$[0, 0]$	$[0, 0]$	$[0, 0]$
spine #1	spine #2	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
spine #2	spine #2	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
spine #3	spine #3	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
spine #4	spine #3	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
spine #5	spine #4	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
spine #6	spine #5	$[0, 0]$	$[0, \inf)$	$[0, 0]$
tail #1	tail #1 (leaf)	$[0, 0]$	$[0, 0]$	$[0, 0]$
tail #2	tail #2	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
tail #3	tail #3	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
tail #4	tail #4	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
tail #5	tail #5	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
tail #6	spine #1	$[0, 0]$	$[0, \inf)$	$(-\inf, \inf)$
left shoulder	left shoulder	$(-\inf, 0]$	$[0, \inf)$	$[0, 0]$
left elbow	left elbow	$(-\inf, 0]$	$[0, 0]$	$[0, 0]$
left wrist	left wrist	$[0, 0]$	$(-\inf, 0]$	$[0, 0]$
left finger #1	left finger (leaf)	$(-\inf, \inf)$	$[0, 0]$	$(-\inf, \inf)$
left finger #2	left finger (leaf)	$[0, 0]$	$[0, 0]$	$[0, 0]$
left finger #3	left finger (leaf)	$(-\inf, \inf)$	$[0, 0]$	$(-\inf, \inf)$
left side	spine #3	$(-\inf, 0]$	$(-\inf, \inf)$	$(-\inf, \inf)$
left hip	left hip	$(-\inf, 0]$	$[0, \inf)$	$[0, 0]$
left knee	left knee	$(-\inf, 0]$	$[0, 0]$	$[0, 0]$
left ankle	left ankle	$(-\inf, 0]$	$[0, 0]$	$[0, 0]$
left front paw	left front paw	$[0, 0]$	$(-\inf, 0]$	$[0, 0]$
left toe #1	left toe (leaf)	$(-\inf, \inf)$	$[0, 0]$	$(-\inf, \inf)$
left toe #2	left toe (leaf)	$[0, 0]$	$[0, 0]$	$[0, 0]$
left toe #3	left toe (leaf)	$(-\inf, \inf)$	$[0, 0]$	$(-\inf, \inf)$

Table 2.2: Enforced intervals for the elements of the reduced joint-to-marker-translation vector v^*. The intervals are in agreement with the surface marker pattern, which is applied to different animal subjects (Figure 2.9). The x-direction points from the left to the right, the y-direction from the bottom to the top and the z-direction from the front to the back, when an animal's internal coordinate system is used as a frame of reference. Note that each enforced interval $[0, 0]$ reduces the number of free parameters by one, when the anatomy of an animal subject is learned via gradient descent optimization (Section 2.2.3). A single exception from the stated intervals is given for the upper bound of the left-sided surface marker on the shoulder in the z-direction. This upper bound was set to 0 for two large animal subjects to prevent the bone lengths of the collarbones to become zero during learning (animals #5 and #6 in Section 3.1).

2.2.7 Constraining bone rotations

Unlike the anatomy-encoding variables l and v, the pose-encoding variables t_τ and r_τ are dismissed after minimizing Equation 2.11, since deploying plain gradient descent optimization for reconstructing poses, i.e. learning t_τ and r_τ, does not give sufficient results (Section 3.2 and 3.3). Nevertheless, it is still beneficial to constrain t_τ and r_τ, since learning pose- and anatomy-encoding variables is a joint process. As a consequence, learning severely erroneous poses has the potential to also introduce errors to the learned anatomy. Thus, a subset of the bone rotations r_τ is constrained by enforcing limb bone rotations to stay within physiological limits (Figure 2.10). These physiological limits correspond to joint angle limits taken from a study, in which minimal and maximal limb bone rotations with respect to flexion / extension, abduction / adduction and internal / external rotation were measured in domestic house cats [111]. These measured joint angle limits are implemented into the kinematic chain used for modeling an animal's skeletal pose via constant box constraints, which are enforced for individual elements of Rodrigues vectors representing limb bone rotations (Table 2.3).

Since the measured joint angle limits are only available for the limb bones, the respective limits for all other bones are set to $[-90, 90]$ for the x- and y-rotation and $[0, 0]$ for the z-rotation. This ensures that the end-joint of a non-limb bone is allowed to reach any point on a hemisphere with radius equal to the bone length of the respective bone, which is assumed to be sufficient for modeling all physiologically-feasible bone rotations. The only exception to this is the bone, whose start-point is the root joint, since rotations around the root joint define the overall orientation of an animal's entire body and do not correspond to any anatomical rotation. Consequently, rotations around the root joint remain unconstrained.

However, in the context of the proposed pose reconstruction framework rats are the animal species of interest. Therefore, it is assumed that the measured joint angle limits for domestic house cats generalize to those for rats, since both animal species are four-legged mammals, which share a common anatomical structure. Besides, there is no equivalent study available, which states the corresponding physiological joint angle limits for rats.

While the measured joint angle limits in the study refer to the three anatomical rotations, i.e.

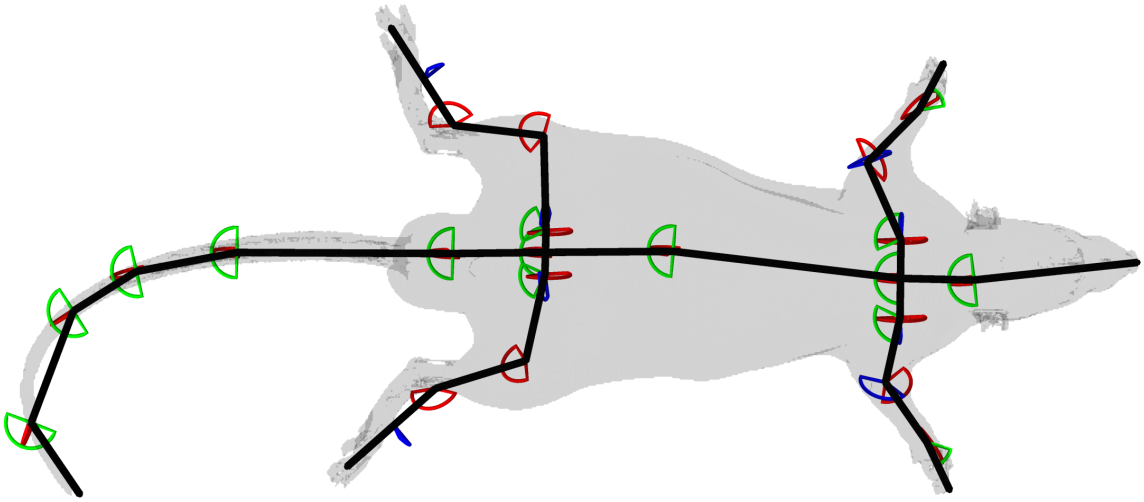

Figure 2.10: Illustration of enforced joint angle limits. The shown skeletal pose is reconstructed based on three-dimensional surface marker positions obtained via an MRI scan (Section 3.1). Physiological bones are approximated by a kinematic chain (black lines), whereas the body surface of the animal subject is directly obtained from the MRI scan (gray area). In anatomical terms, the shown rotations indicated by the colored angles are equivalent to flexion / extension (red), abduction / adduction (green) and internal / external rotation (blue).

flexion / extension, abduction / adduction and internal / external rotation, which are effectively equivalent to rotations via Euler angles, the proposed pose reconstruction framework uses Rodrigues vectors to parameterize rotations (Section 2.1.1). In the study all bone rotations were measured independently from each other. However, anatomical joint angle limits are in fact co-dependent. For instance, in humans the extent to which the forearm can be flexed depends on the rotation of the shoulder. Thus, the maximal extent of flexion is higher when the forearm is flexed in front of the torso, compared to when it is flexed behind the torso. While it is in principle possible to learn such co-dependent joint angle limits from recorded images [112], there is no such data available for rats. Additionally, not every bone movement can be described by only applying a single anatomical rotation to a respective bone. In fact, there are bone movements, which are the result of a superposition of different anatomical rotations, e.g. a combination of flexion and abduction. Physically measuring joint angle limits for such bone movements is complex as it would require not only considering rotations around one of the three axes given by Euler angles but any rotation possible. As a consequence, it is necessary to find reasonable approximations, when modeling physiological bone rotations whose constraints are furthermore co-dependent.

In the proposed pose reconstruction framework it is assumed that joint angle limits for different bone rotations are independent from each other, which allows for implementing them as constant quantities in the form of simple box constraints. Concerning the discrepancies between Euler angles and Rodrigues vectors, two different cases have to be differentiated. For situations in which only a single anatomical rotation is sufficient to describe bone movements, the Euler angle and Rodrigues vector parameterization coincide and are effectively equivalent. This is, for instance, the case for rotations around the knee or ankle joint, since those rotations have only a single rotational degree of freedom (Table 2.3). Here, the enforced box constraints in the proposed pose estimation framework are therefore in line with those stated by the study, irrespective of the fact that Rodrigues vectors are used to parameterize rotations. For all situations, in which bone movements are described by a superposition of anatomical rotations, the Euler angle and Rodrigues vector parameterization differ. However, for those situations accurately measured joint angle limits are not available. Here, it is therefore assumed that any respective bone rotation is in a transition state between one of the three anatomical rotations, such that enforcing box constraints on the individual elements of an associated Rodrigues vector is justified. Additionally, using Rodrigues vectors for parameterizing rotations generally prevents the potential occurrence of issues related to the shortcomings of the Euler angle paramterization, i.e. gimbal lock configurations (Section 2.1.1). As a consequence, it is assumed that parameterizing bone rotations via Rodrigues vectors

joint	x (deg)	y (deg)	z (deg)
left shoulder	$[25, 205]$	$[-85, 25]$	$[-35, 35]$
left elbow	$[2.5, 145]$	$[0, 0]$	$[-100, 45]$
left wrist	$[-135, 35]$	$[-12.5, 37.5]$	$[0, 0]$
left hip	$[35, 195]$	$[-65, 25]$	$[-85, 40]$
left knee	$[-145, 15]$	$[0, 0]$	$[0, 0]$
left ankle	$[-10, 145]$	$[0, 0]$	$[0, 0]$
left hind paw	$[0, 0]$	$[0, 0]$	$[-15, 35]$

Table 2.3: Physiological joint angle limits for the limb bones of the domestic house cat taken from a respective study [111]. Limb bone rotations are enforced to stay within the stated intervals, when the anatomy of an animal subject is learned via gradient descent optimization (Section 2.2.3). In anatomical terms, the x-rotation is equivalent to flexion / extension, the y-rotation to abduction / adduction and the z-rotation to internal / external rotation. Due to the symmetry of an animal's body, the joint angle limits for left- and right-sided limb bones are identical with the exception that absolute values of the lower and upper limits for the y-rotations are flipped, e.g. the y-rotation limit for the right shoulder joint is $[-25, 85]$.

and implementing respective box constraints on their individual elements represents a reasonable solution for approximating physiological bone rotations.

2.2.8 Re-scaling input and output parameters

It is generally advantageous to re-scale the variables involved in a numerical optimization scheme, such that their magnitudes are roughly identical [113]. Otherwise different magnitudes of, for instance, the translation vector t_τ and the bone rotations r_τ have the potential to cause the magnitudes of their respective gradients to be different as well. Since these gradients are required and used in the optimization scheme given by Equation 2.11, this can in turn have negative consequences for learning skeletal anatomies and poses. For instance, not re-scaling the variables might lead to an exaggeration of how much influence the optimization of t_τ has on the final cost value, placing insufficient emphasis on also learning a good solution for r_τ, as long as t_τ is estimated reasonably well. Thus, the translation vector t_τ and the bone rotations r_τ are re-scaled before any form of optimization is performed, which gives the re-scaled input variables t_τ^* and r_τ^*:

$$t_\tau^* = \frac{t_\tau}{s_t} \tag{2.14}$$

$$r_\tau^* = \frac{r_\tau}{s_r}, \tag{2.15}$$

with the scaling constants $s_t = 50$ and $s_r = 90$.

The value of the scaling constant s_t for the translation vector t_τ is based on the sizes of the different arenas used during the experiments, which were conducted within the scope of this thesis (Chapter 3). The maximum distance an animal could cross inside the arenas was close to one meter before it reached an obstacle, i.e. a wall or an abyss. In each conducted experiment the given frame of reference for the translation vector t_τ was a world coordinate system, which was placed in the center the respective arena, aligned with the arena's three main axes and whose measuring unit was given in centimeters. Consequently, dividing t_τ by s_t ensured that entries of t_τ^* stayed within $[-1, 1]$ during the optimization process.

The choice for the value of the scaling constant s_r for the bone rotations r_τ is based on the interval $[-90, 90]$ for joint angle limits associated to non-limb bones (Section 2.2.7). As a result of the scaling, the entries of r_τ^* were ensured to be roughly of the same magnitude as those of t_τ^* during the optimization process. To obtain the required re-scaled box constraints for r_τ^*, the intervals for all enforced joint angle limits (Table 2.3) are re-scaled accordingly, i.e. they are divided by s_r.

Furthermore, the optimization procedure given by Equation 2.11 contains the x- and y-distances between ground truth and projected surface marker positions $d_{\tau ij} = \bar{m}_{\tau ij} - \tilde{m}_{\tau ij}$, which are re-scaled as follows:

$$d_{\tau ij}^* = \left(\frac{d_{\tau ij_1}}{s_x}, \frac{d_{\tau ij_2}}{s_y} \right)^T, \tag{2.16}$$

with $s_x = 640$ and $s_y = 512$. Here, the choice for the values of the scaling constants s_x and s_y are based on the sensor sizes of the used cameras. The respective sensor dimensions were equal to 1280x1024 px^2 in all conducted experiments. Consequently, re-scaling ensured that the value of each calculated distance $d_{\tau ij}^*$ stayed within the interval $[-2, 2]$ during the optimization process.

To learn skeletal anatomies it is also necessary to learn the reduced bone lengths l^* as well as the reduced joint-to-marker-translation vectors v^* (Section 2.2.9). However, additional scaling of these two variables was not performed, since sufficient results with respect to the learned anatomies could already be obtained without any re-scaling (Section 3.1).

2.2.9 Constrained anatomy learning

Taking into account all of the previously discussed constraints (Section 2.2.4 to 2.2.7) as well as the parameter re-scaling (Section 2.2.8), the pose- and anatomy-encoding parameters are learned by minimizing a reformulated version of the objective function given in Equation 2.11 via gradient descent optimization. The respective optimization scheme is given by

$$\underset{\substack{t^*_\tau, r^*_\tau, l^*, v^* \\ \forall \tau \in \{1,\dots,n_{\text{time}}\}}}{\arg\min} \sum_{\tau=1}^{n_{\text{time}}} \sum_{i=1}^{n_{\text{cam}}} \sum_{j=1}^{n_{\text{marker}}} \delta_{\tau ij} \left|\left| d^*_{\tau ij} \right|\right|^2, \tag{2.17}$$

where box constraints for the reduced bone lengths l^* (Section 2.2.5), the reduced joint-to-marker-translation vectors v^* (Section 2.2.6) and the re-scaled bone rotations r^* (Section 2.2.7) are enforced. Here, the gradients are computed automatically via an auto-differentiation library [114] and the minimization itself is performed using an implementation of the L-BFGS-B algorithm [106, 115]. By minimizing discrepancies between ground truth and projected two-dimensional marker positions, given by the re-scaled distances $d^*_{\tau ij}$ (Section 2.2.8), the skeletal anatomy of an animal is learned and the pose for each time point τ is reconstructed (Figure 2.11).

However, while learning the anatomy-encoding variables l^* and v^* via gradient descent optimization gives reasonable results (Section 3.1), the pose-encoding variables t^* and r^* need to be discarded, since respective pose reconstruction results are not satisfactory. For instance, when learning t^* and r^* via gradient descent optimization by minimizing the constrained objective function given by Equation 2.17, resulting poses are not consistent in time, which leads to unreasonable fast limb movements in three-dimensional space (Section 3.2). This is due to the fact that the pose-encoding variables vary through time and are learned independently, such that for each individual time point τ the learned values for t_τ and r_τ are uncorrelated and therefore vastly different from each other. For the anatomy-encoding parameters l and v this is not the case, since they do not change with time and are therefore shared across all time points. As a consequence, learning l and v by minimizing the constrained objective function given in Equation 2.17 gives reasonable solutions, such that the kinematic chain used for modeling poses becomes a plausible representation of the skeletal anatomy of an individual animal subject. Particularly, measurable quantities, like bone lengths and three-dimensional joint positions, are consistent with the therefore learned skeletal anatomies (Section 3.1). However, to obtain reasonable estimates for an animal's skeletal pose, given by t_τ and r_τ, it is necessary to further improve the pose reconstruction framework (Section 2.3).

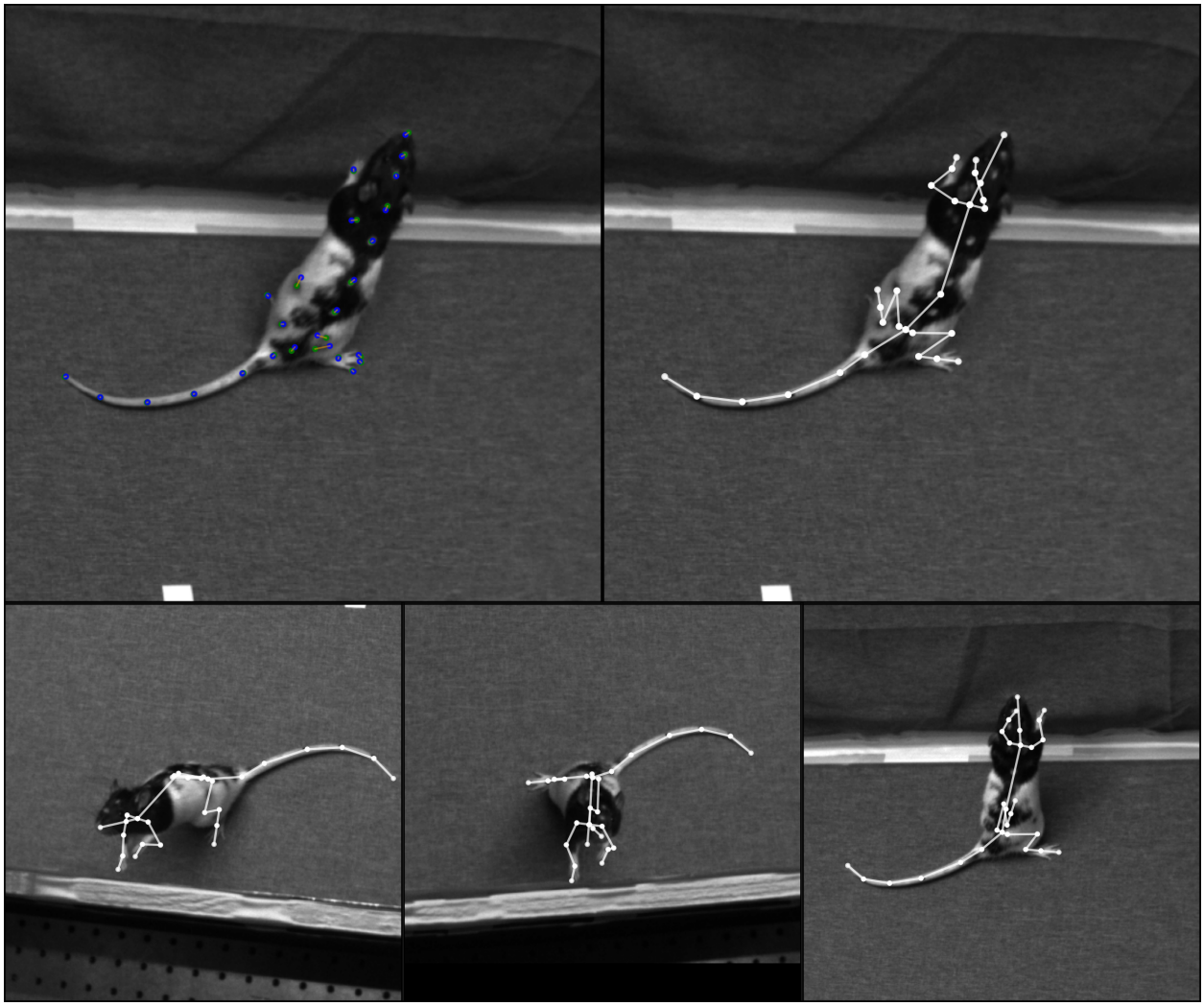

Figure 2.11: The anatomy-encoding variables of an animal subject, i.e. its bone lengths and joint-to-marker-translation vectors, are learned using multiple images recorded via four different cameras at the same time point. To achieve this the two-dimensional Euclidean distances (orange lines) between manually labeled and projected surface marker locations (green and blue dots respectively) are minimized (top left). Learning the anatomy-encoding variables also yields the animal's pose, which allows for projecting the resulting three-dimensional configuration of the skeleton onto the image plane of all four cameras (top right, bottom).

2.3 Probabilistic skeleton-based pose estimation

The previous section introduced a skeleton model containing a kinematic chain, which allows for representing skeletal poses of freely-moving animals by approximating their three-dimensional joint and bone positions in space (Section 2.2.1). Furthermore, the previous section described how approximated joint positions are related to markers painted on an animal's body surface (Section 2.2.2). When constraints are enforced on the anatomy- and pose-encoding variables of the skeleton model (Section 2.2.4 to 2.2.7), the skeletal poses in a behavioral sequence can in principle be learned from images recorded via multiple cameras by minimizing the discrepancies between ground truth and projected surface marker locations (Section 2.2.9).

However, learning the pose-encoding variables for each individual time point of a behavioral sequence independently from each other is problematic, since poses are consistent in time, i.e. poses of consecutive time points are correlated and therefore similar. To account for time-

consisted skeletal poses, this section introduces a state space model, which enables reconstructing skeletal poses in a probabilistic manner (Section 2.3.1). Given automatically detected two-dimensional surface marker locations from a trained deep neural network (Section 2.3.2), the state space model allows for inferring the pose-encoding variables via a Bayesian filter or smoother (Section 2.3.4 and 2.3.5), which implicitly enforces temporal constraints. Since this newly introduced inference scheme does not allow for enforcing box constraints on the pose-encoding variables to guarantee that modeled bone rotations are coherent with physiological joint angle limits, the initial state space model is further modified by introducing new model variables to achieve the same outcome (Section 2.3.6).

2.3.1 The state space model

In the proposed pose reconstruction framework, a state space model is used to simulate the dynamics of pose changes over time (Figure 2.12). In mathematical terms, the state space model is given by a transition and an emission equation:

$$z_\tau = \mathrm{f}\left(z_{\tau-1}\right) + \epsilon_z = z_{\tau-1} + \epsilon_z \tag{2.18}$$

$$x_\tau = \mathrm{g}\left(z_\tau\right) + \epsilon_x. \tag{2.19}$$

For each time point $\tau \in \{1, ..., T\}$ of a behavioral sequence of length T, the pose-encoding variables of the skeleton model are stored within a state variable $z_\tau \in \mathbb{R}^{n_z}$ and the observable two-dimensional surface marker locations are stored within a measurement variable $x_\tau \in \mathbb{R}^{n_x}$. Therefore, z_τ contains the elements of the Rodrigues vectors r_τ, which store the bone rotations of each bone, as well as the three elements of the global translation vector t_τ, which stores the three-dimensional location of the root joint of the modeled skeleton. Thus, the dimensionality of z_τ only depends on the number of modeled bones n_{bones}, i.e. $n_z = 3\left(n_{\mathrm{bones}} + 1\right)$, whereas the dimensionality of x_τ is a function of the number of used cameras n_{cam} and surface markers n_{marker}, i.e. $n_x = 2\, n_{\mathrm{cam}}\, n_{\mathrm{marker}}$.

The transition equation (Equation 2.18) describes how the state variables change between two consecutive time points $\tau - 1$ and τ. Here, the transition function f and the noise variable ϵ_z are used to compute the current state variable z_τ from the previous one $z_{\tau-1}$. In the proposed

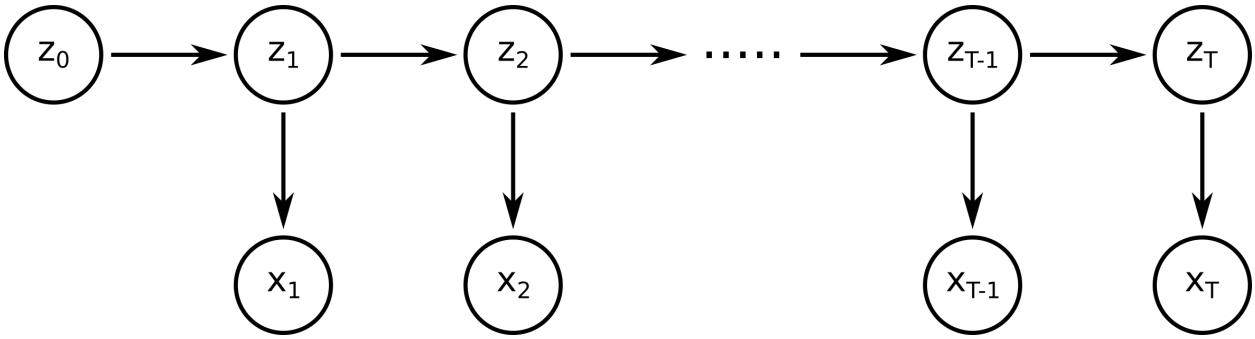

Figure 2.12: Graphical illustration of the state space model, which is used to model how skeletal poses of a freely-moving animal change over time. For each time point of a behavioral sequence a state variable z_τ encodes the pose of an animal and gives rise to a measurement variable x_τ, which is calculated using an emission function g and a noise variable ϵ_x, such that $x_\tau = \mathrm{g}\left(z_\tau\right) + \epsilon_x$. The measurement variable x_τ stores directly observable two-dimensional surface marker locations, which are recorded via a multi-camera setup, whereas the noise variable $\epsilon_x \sim \mathcal{N}\left(0, V_x\right)$ simulates measurement noise. The temporal progression of poses, which describes how a pose at time point $\tau - 1$ gives rise to a pose at the subsequent time point τ, is modeled via an additional noise variable $\epsilon_z \sim \mathcal{N}\left(0, V_z\right)$, such that $z_\tau = z_{\tau-1} + \epsilon_z$. The random variables ϵ_z and ϵ_x together with the initial state variable $z_0 \sim \mathcal{N}\left(\mu_0, V_0\right)$ account for the probabilistic nature of the state space model.

Algorithm 5: The emission function of the state space model.

function $\mathrm{g}(z_\tau)$
  $t \leftarrow s_t(z_{\tau_1}, z_{\tau_2}, z_{\tau_3})^T$ ▷ Obtain global translation t
  for $i \in \{1, .., n_{\mathrm{bone}}\}$ **do**
    $r_i \leftarrow s_r\left(z_{\tau_{3i+1}}, z_{\tau_{3i+2}}, z_{\tau_{3i+3}}\right)^T$ ▷ Obtain bone rotations r_i
  $m_{3\mathrm{D}} \leftarrow \mathrm{f}_{\mathrm{surface}}(t, r, l, v)$ ▷ Obtain 3D marker locations $m_{3\mathrm{D}}$
  for $i \in \{1, .., n_{\mathrm{cam}}\}$ **do**
    for $j \in \{1, .., n_{\mathrm{marker}}\}$ **do**
      $m_{2\mathrm{D}} \leftarrow \mathrm{f}_{3\mathrm{D}\rightarrow 2\mathrm{D}}\left(m_{3\mathrm{D}_j}, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i\right)$ ▷ Obtain 2D marker locations $m_{2\mathrm{D}}$
      $m^*_{n_{\mathrm{marker}}(i-1)+j} \leftarrow \left(\frac{m_{2\mathrm{D}_1}}{s_x} - 1, \frac{m_{2\mathrm{D}_2}}{s_y} - 1\right)^T$ ▷ Normalize x- and y-coordinates
  $x^*_\tau \leftarrow \mathrm{cat}\left(m^*_1, m^*_2, ..., m^*_{n_{\mathrm{cam}} n_{\mathrm{marker}}}\right)$ ▷ Obtain noise-free x^*_τ via concatenation
  return x^*_τ

pose reconstruction framework it is assumed that the transition function f is equal to the identity function, i.e. $\mathrm{f}(z_{\tau-1}) = z_{\tau-1}$, such that only the noise variable ϵ_z governs the dynamics of pose changes over time.

Computing x_τ given z_τ is performed via the emission equation (Equation 2.19) containing the emission function g and the noise variable ϵ_x. Here, the emission function g extracts the pose-encoding variables, i.e. the global translation vector t_τ and the bone rotations r_τ, from the state variable z_τ, propagates them through the functions $\mathrm{f}_{\mathrm{surface}}$ (Algorithm 2) and $\mathrm{f}_{3\mathrm{D}\rightarrow 2\mathrm{D}}$ (Equation 2.7) to obtain the noise-free two-dimensional surface marker locations and stores this final result within a n_x-dimensional vector (Algorithm 5). Thus, to calculate x_τ from z_τ it is necessary to learn the skeletal anatomy of an animal subject and calibrate the deployed multi-camera setup beforehand. Here, learning the skeletal anatomy gives the bone-lengths l and the join-to-marker-translation vectors v, whereas calibrating the multi-camera setup gives the intrinsic and extrinsic parameters $\tilde{r}_i$, $\tilde{t}_i$, $\tilde{k}_i$ and $\tilde{A}_i$ for each camera i. These quantities are required input parameters for the functions $\mathrm{f}_{\mathrm{surface}}$ and $\mathrm{f}_{3\mathrm{D}\rightarrow 2\mathrm{D}}$.

The two noise variables $\epsilon_z \in \mathbb{R}^{n_z}$ and $\epsilon_x \in \mathbb{R}^{n_x}$ as well as the initial state variable z_0 account for the probabilistic nature of the state space model, since they are assumed to be random variables drawn from normal distributions, i.e. $\epsilon_z \sim \mathcal{N}(0, V_z)$, $\epsilon_x \sim \mathcal{N}(0, V_x)$ and $z_0 \sim \mathcal{N}(\mu_0, V_0)$. Consequently, the dynamics of changing skeletal poses are described in their entirety by the probabilistic hyper-parameters of the state space model, i.e. the model parameters $\Theta = \{\mu_0, V_0, V_z, V_x\}$.

When the model parameters Θ and all measurement variables $x = \{x_1, x_2, ..., x_T\}$ of a behavioral sequence are given, deploying the state space model allows for inferring the state variables $z = \{z_0, z_1, ..., z_T\}$ and therefore reconstructing poses. To achieve this, the measurement variables of the entire behavioral sequence are obtained via a trained deep neural network, which automatically estimates the two-dimensional surface marker locations in the recorded video data (Section 2.3.2). The actual inference of z is performed via a Bayesian filter (Section 2.3.4) or smoother (Section 2.3.5), which implicitly incorporates temporal constraints into the pose reconstruction framework.

2.3.2 Detecting surface marker locations via deep neural networks

Using the state space model given by Equation 2.18 and 2.19 to infer the pose-encoding state variables z requires full knowledge of the measurement variables x, i.e. the two-dimensional surface marker locations in the recorded video data. Obtaining the surface marker locations solely via manually annotated labels is infeasible, since the video data is generated using multiple cameras,

which record images at high frame rates, e.g. 200 Hz, to allow for reconstructing poses with high temporal resolution. Consequently, reconstructing poses of a behavioral sequence, which is only a few seconds long, already places the total number of frames, for which labels are required, in the order of thousands, e.g. 4000 images in total for a 5 s long sequence recorded via 4 cameras. This yields a considerable amount of data, whose sheer quantity deems manual labeling impractical. Thus, it is necessary to reduce the manual labor involved in the process of obtaining the two-dimensional surface marker locations by shifting to automatized methods.

An established tool for automatically classifying, segmenting and detecting distinct objects or features from images are artificial neural networks, which are trained via supervised learning [116–118]. In the proposed pose reconstruction framework the used artificial neural network for this task is DeepLabCut [76], which is a deep convolutional neural network [55, 56] specifically designed for the automatized extraction of anatomical surface features, e.g. the snout of a freely-moving rat, from recorded video data. Particularly, a DeepLabCut network is a deep residual neural network [119] and builds on the concepts introduced by it's predecessors networks, i.e. DeepCut [57] and DeeperCut [58], whose original purposes are situated in the realm of human pose estimation (Section 1.2.1).

For training a DeepLabCut network only a comparably small number of manually labeled frames, i.e. a few hundred frames, is sufficient to ensure that the network's feature detection capabilities generalize to unseen data, such that detecting the two-dimensional surface marker locations in the recorded video data can be automated. After a DeepLabCut network is trained successfully, processing a previously unseen image via the trained network yields a score map for each surface marker, which contains a score value for each pixel of the input image. The score value of a single pixel indicates the probability of a respective two-dimensional surface marker being located at this pixel. By choosing the pixel location with the highest score vale for each individual surface marker, all surface marker locations are automatically detected. Furthermore, since a comparably small score value indicates a small certainty with respect to the detection accuracy, the risk for detecting incorrect surface marker locations is reduced by requiring the score value to be above a predefined threshold, the pcutoff-value, in order to deem a detection successful (Figure 2.13). In the proposed pose reconstruction framework this pcutoff-value equals 0.9.

However, while this approach provides an efficient way for automatically processing a high number of images, it does not necessarily guarantee that all surface marker locations are detected accurately. In fact, even though the detected surface marker locations given by a trained DeepLabCut network are treated as actual measurements within the state space model, the detection accuracy can be low, irrespective of the corresponding score values. Thus, the detected surface marker locations need to be treated with caution, which is accounted for by using a Bayesian filter or smoother for inferring the state variables in the state space model.

Since the Bayesian filter and smoother indirectly incorporate temporal constraints into the pose reconstruction scheme (Section 2.3.4 and 2.3.5), the influence of isolated misdetections on the reconstructed poses can be reduced. Furthermore, frequently occurring body part occlusions can cause surface markers to be temporarily invisible in the recorded video data, which leads to undetectable surface marker locations. As a consequence the underlying state space model contains missing measurements, i.e. a lack of detections for some of the surface marker locations. Thus, the measurement variables x are incomplete, i.e. the entries of a single measurement variable x_τ remain empty, when they correspond to missing measurements. Due to frequent body part occlusions it is necessary to formulate the algorithm for the Bayesian filter, such that it accounts for the dynamically changing visibility of the surface markers in a recorded behavioral sequence (Section 2.3.4).

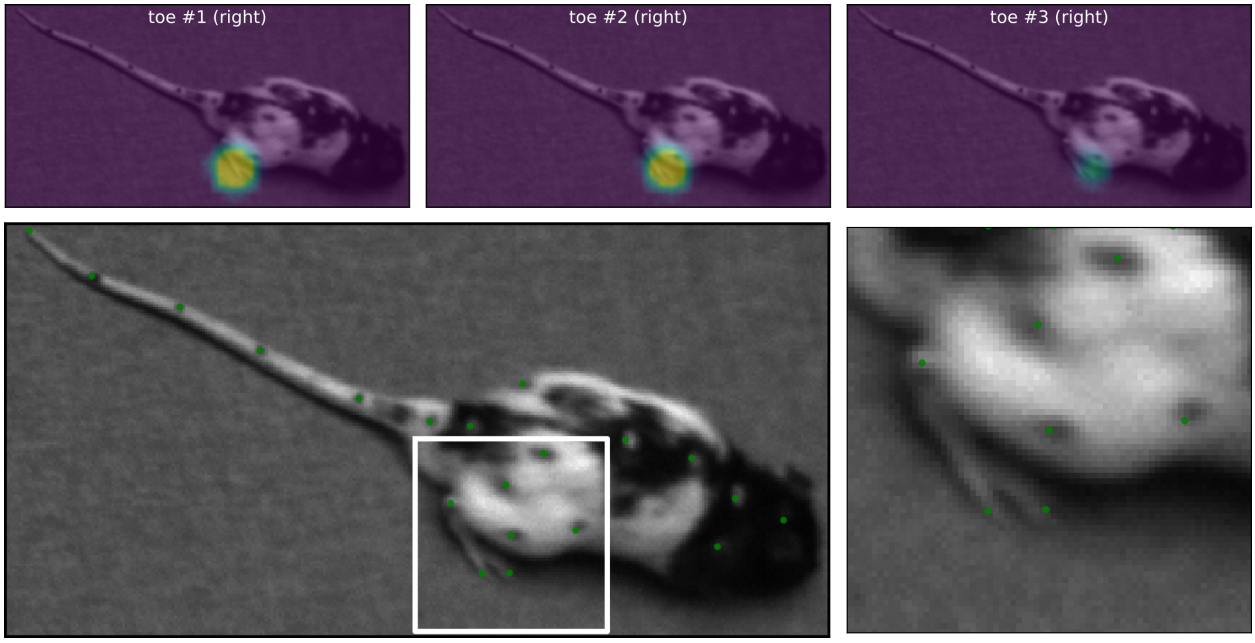

Figure 2.13: Example for how a trained DeepLabCut network is used to process a recorded image to automatically detect surface marker locations. For each surface marker the network gives a score map, which indicates the most likely location of a surface marker in the image. The respective score maps of three different surface markers at the right hind paw are shown (top row). The surface marker locations (green dots) are obtained from these score maps by choosing the pixel with the highest score value above a predefined threshold (bottom row). If the score value is below this threshold the corresponding surface marker is classified as not detected (here: toe #3), which leads to incomplete measurement variables within the state space model.

2.3.3 The unscented transform

To perform inference of the sate variables z in the state space model via a Bayesian filter or smoother, it is necessary to approximate arbitrary with normal probability distributions. In the state space model the initial state variable z_0 as well as the two noise variables ϵ_z and ϵ_x are assumed to be normally distributed, which causes each state variable z_τ at a time point τ to be normally distributed as well, due to the linearity of the transition equation [120, 121]. However, this property is lost for each measurement variable x_τ, due to the non-linearity of the emission function g. In fact, the underlying probability density function associated with the probability distribution of x_τ is unknown. Nevertheless, by drawing samples from the unknown distribution of x_τ, subsequently calculating the expectation value and covariance matrix of these samples and then using both quantities to parameterize a normal distribution, it is possible to approximate the unknown distribution of x_τ with a normal distribution [122, 123].

Thus, in order to approximate an arbitrary with a normal probability distribution, it is necessary to approximate an expectation value

$$\mathbb{E}\left[\mathrm{h}\left(y\right)\right] = \int \mathrm{p}\left(y\right) \mathrm{h}\left(y\right) dy \tag{2.20}$$

and a covariance matrix

$$\mathrm{Var}\left(\mathrm{h}\left(y\right)\right) = \mathbb{E}\left[\left(\mathrm{h}\left(y\right) - \mathbb{E}\left[\mathrm{h}\left(y\right)\right]\right)\left(\mathrm{h}\left(y\right) - \mathbb{E}\left[\mathrm{h}\left(y\right)\right]\right)^T\right], \tag{2.21}$$

where h is an arbitrary function and $y \in \mathbb{R}^d$ is a normally distributed random variable, i.e. $y \sim \mathcal{N}\left(\mu_y, \Sigma_y\right)$, of arbitrary size, i.e. $d \in \mathbb{N}$. To generate samples $\mathcal{Y} \in \mathbb{R}^{2d+1 \times d}$ from the normal distribution $\mathcal{N}\left(\mu_y, \Sigma_y\right)$, the unscented transform $\mathrm{f_{ut}}$ is used, such that $\mathcal{Y} = \mathrm{f_{ut}}\left(\mu_y, \Sigma_y\right)$ [122, 123]. The

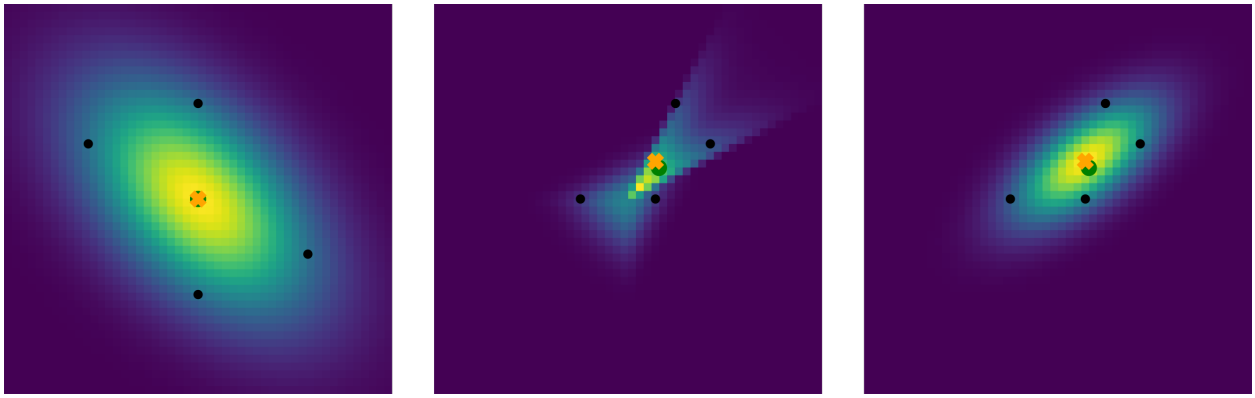

Figure 2.14: Two-dimensional example for how the unscented transform $\mathrm{f_{ut}}$ is used to approximate an arbitrary probability density function. From an initial two-dimensional normal distribution $\mathcal{N}(\mu_y, \Sigma_y)$ with expectation value $\mu_y = (0,0)^T$ (green dot) and covariances $\Sigma_{y_{11}} = 1$, $\Sigma_{y_{22}} = 1$ and $\Sigma_{y_{12}} = \Sigma_{y_{21}} = -0.5$, sigma points $\mathcal{Y}$ (black dots) are calculated, such that $\mathcal{Y} = \mathrm{f_{ut}}(\mu_y, \Sigma_y)$. Calculating the predicted expectation value (orange cross) from the sigma points with non-zero weights, i.e. $\mathcal{Y}_2$, $\mathcal{Y}_3$, $\mathcal{Y}_4$ and $\mathcal{Y}_5$, gives the true expectation value μ_y of the initial distribution (left). However, applying a non-linear function $\mathrm{h}(y) = 0.5\left(\mathrm{abs}(y) + (y_1 + y_2, y_1 + y_2)^T\right)$ to samples from the initial normal distribution $\mathcal{N}(\mu_y, \Sigma_y)$ yields a new sample distribution (center). This new probability distribution is not longer normal and has a new expectation value equal to $(0.39, 0.39)^T$, which is different from μ_y. Likewise, applying h to the sigma points $\mathcal{Y}$ also changes the predicted expectation value $\mu_{\mathrm{h}(y)}$. The normal distribution $\mathcal{N}\left(\mu_{\mathrm{h}(y)}, \Sigma_{\mathrm{h}(y)}\right)$ is then used to approximate the new sample distribution, where $\mu_{\mathrm{h}(y)}$ and $\Sigma_{\mathrm{h}(y)}$ are calculated from the new sigma points $\mathrm{h}(\mathcal{Y}_2)$, $\mathrm{h}(\mathcal{Y}_3)$, $\mathrm{h}(\mathcal{Y}_4)$ and $\mathrm{h}(\mathcal{Y}_5)$ giving $\mu_{\mathrm{h}(y)} = (0.35, 0.48)^T$, $\Sigma_{\mathrm{h}(y)_{11}} = 0.37$, $\Sigma_{\mathrm{h}(y)_{22}} = 0.26$ and $\Sigma_{\mathrm{h}(y)_{12}} = \Sigma_{\mathrm{h}(y)_{21}} = 0.20$ (right).

samples $\mathcal{Y}$ are called sigma points and are representative of the probability distribution $\mathcal{N}(\mu_y, \Sigma_y)$ in the sense that they are systematically spread around μ_y (Figure 2.14). The required unscented transform $\mathrm{f_{ut}}$ is defined in Algorithm 6 [123]. Here, $\mathrm{f_{cholesky}}(\Sigma_y)$ denotes the Cholesky decomposition of the covariance matrix Σ_y, which computes a lower triangular matrix L, such that $LL^T = \Sigma_y$ [124], and λ is a scalar value defined as

$$\lambda = \alpha^2 (d + \kappa) - d, \tag{2.22}$$

where the parameters are choosen as $\alpha = 1$ and $\kappa = 0$, such that $\lambda = 0$ [123, 125].

Using the sigma points $\mathcal{Y}$, the expectation value $\mathbb{E}[\mathrm{h}(y)]$ and the covariance matrix $\mathrm{Var}(\mathrm{h}(y))$ are approximated as

$$\mathbb{E}[\mathrm{h}(y)] \approx \sum_{i=1}^{2d+1} w_i \, \mathrm{h}(\mathcal{Y}_i) = \mu_{\mathrm{h}(y)} \tag{2.23}$$

and

$$\mathrm{Var}(\mathrm{h}(y)) \approx \sum_{i=1}^{2d+1} w_i \big(\mathrm{h}(\mathcal{Y}_i) - \mu_{\mathrm{h}(y)}\big)\big(\mathrm{h}(\mathcal{Y}_i) - \mu_{\mathrm{h}(y)}\big)^T = \Sigma_{\mathrm{h}(y)}, \tag{2.24}$$

Algorithm 6: The unscented transform.

function $\mathrm{f_{ut}}(\mu_y, \Sigma_y)$
  $L \leftarrow \mathrm{f_{cholesky}}(\Sigma_y)$ ▷ Obtain lower triangular matrix L
  $\mathcal{Y}_1 \leftarrow \mu_y$ ▷ Set first unsymmetrical sigma point $\mathcal{Y}_1$
  for $i \in \{2, ..., d+1\}$ **do**
    $\mathcal{Y}_i \leftarrow \mu_y + \sqrt{d + \lambda}\, L^T{}_i$ ▷ Compute first half of symmetrical sigma points
  for $i \in \{d+2, ..., 2d+1\}$ **do**
    $\mathcal{Y}_i \leftarrow \mu_y - \sqrt{d + \lambda}\, L^T{}_i$ ▷ Compute second half of symmetrical sigma points
  return $\mathcal{Y}$

where the weights $w \in \mathbb{R}^{2d+1}$ are defined as

$$w_1 = \frac{\lambda}{d+\lambda} = 0 \tag{2.25}$$

$$w_i = \frac{1}{2\left(d+\lambda\right)} = \frac{1}{2d} \; \forall i \in \{2, ..., 2d+1\}. \tag{2.26}$$

By assuming $\mathrm{h}\left(y\right) \sim \mathcal{N}\left(\mu_{\mathrm{h}(y)}, \Sigma_{\mathrm{h}(y)}\right)$, the parameters $\mu_{\mathrm{h}(y)}$ and $\Sigma_{\mathrm{h}(y)}$ are then used to parameterize a normal distribution, which approximates the unknown probability density function from which the transformed random variable $\mathrm{h}\left(y\right)$ is drawn from (Figure 2.14).

2.3.4 Bayesian filtering

Inferring the skeletal pose of an animal for an arbitrary time point τ of a behavioral sequence via the state space model and a Bayesian filter is equivalent to estimating the parameters of the normal distribution from which the pose-encoding state variable z_τ is drawn from, i.e. the filtered estimates $\tilde{\mu}_\tau$ and $\tilde{V}_\tau$ for the distribution's expectation value and covariance matrix. A seminal work in this context was published by Rudolf Kálmán in 1960, who introduced a formulation of a Bayesian filter, which is now named after him, i.e. the Kalman filter [126]. The Kalman filter allows for inferring the state variables in a purely linear state space model. However, in the proposed pose reconstruction framework the emission equation of the state space model is non-linear due to the emission function g. Consequently, the Bayesian filter used in the proposed pose reconstruction framework is the unscented Kalman filter [122], which is a modified version of the original Kalman filter.

By utilizing the unscented transform (Section 2.3.3), the unscented Kalman filter allows for inferring $\tilde{\mu}_\tau$ and $\tilde{V}_\tau$, even though the underlying state space model is not purely linear. Like the ordinary Kalman filter, the unscented Kalman filter is an iterative algorithm and uses it's estimates $\tilde{\mu}_{\tau-1}$ and $\tilde{V}_{\tau-1}$ from the previous time point $\tau - 1$, the covariance matrix of the state and measurement noise V_z and V_x as well as the measurement variable x_τ from the current time point τ to calculate $\tilde{\mu}_\tau$ and $\tilde{V}_\tau$. The corresponding inference scheme of the unscented Kalman filter is given by Algorithm 7 [123, 127].

Starting from the filter estimate for the distribution parameters of the state variable $z_{\tau-1}$ at time point $\tau - 1$, i.e. $\tilde{\mu}_{\tau-1}$ and $\tilde{V}_{\tau-1}$, the unscented Kalman filter generates sigma points via the unscented transform $\mathrm{f_{ut}}$ (Algorithm 6) to calculate predictions for how the state variable z_τ and the measurement variable x_τ at time point τ are distributed. These predicted distributions are normal distributions given by $\mathcal{N}\left(\bar{z}, P\right)$ for z_τ and $\mathcal{N}\left(\bar{x}, S\right)$ for x_τ. After theses predictions are calculated, the filter gain K is computed, which allows for updating the predicted distribution $\mathcal{N}\left(\bar{z}, P\right)$. This yields the final filter estimate for the distribution of the state variable z_τ, i.e. $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ (Figure 2.15). Basing the computation of the final distribution $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ at time point τ on the already inferred distribution $\mathcal{N}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}\right)$ from the previous time point $\tau - 1$ establishes a one-sided dependency of $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ on $\mathcal{N}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}\right)$, i.e. $\mathcal{N}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}\right)$ acts as a prior probability distribution for the state variables at time point τ. In practice, this limits the extent to which $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ differs from $\mathcal{N}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}\right)$ and therefore implicitly incorporates temporal constraints into the state space model.

Furthermore, Algorithm 8 accounts for missing measurements by performing computations, which are equivalent to reducing the dimensionality of the measurement space to an approriate size [127]. For instance, assuming the last two elements of the measurement variable x_τ correspond to a missing measurement, i.e. the respective surface marker location could not be detected

Algorithm 7: A single step of the unscented Kalman filter.

function $\mathrm{f}_{\mathrm{ukf\_step}}(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}, V_z, V_x)$
  $\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}\right)$ ▷ Form sigma points $\mathcal{Z}$
  $\mathcal{Z} \leftarrow \mathrm{f}\left(\mathcal{Z}\right)$ ▷ Propagate sigma points through transition function f
  $\bar{z} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{Z}_i$ ▷ Compute predicted mean $\bar{z}$
  $P \leftarrow V_z + \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \bar{z})(\mathcal{Z}_i - \bar{z})^T$ ▷ Compute predicted covariance matrix P
  $\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}\left(\bar{z}, P\right)$ ▷ Form sigma points $\mathcal{Z}$
  $\mathcal{X} \leftarrow \mathrm{g}\left(\mathcal{Z}\right)$ ▷ Propagate sigma points through emission function g
  $\bar{x} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{X}_i$ ▷ Compute predicted mean $\bar{x}$
  $S \leftarrow V_x + \sum_{i=1}^{2n_z+1} w_i (\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^T$ ▷ Compute predicted covariance matrix S
  for $i \in \{1, ..., n_x\}$ **do**
    if $x_{\tau i}$ is missing measurement **then**
      for $j \in \{1, ..., n_x\}$ **do**
        $S_{ij} \leftarrow 0$ ▷ Set rows of missing measurements to 0
        $S_{ji} \leftarrow 0$ ▷ Set columns of missing measurements to 0
      $S_{ii} \leftarrow 1$ ▷ Set diagonal entries to 1 to allow for computing S^{-1}
  $C \leftarrow \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \bar{z})(\mathcal{X}_i - \bar{x})^T$ ▷ Compute cross-covariance matrix C
  for $i \in \{1, ..., n_x\}$ **do**
    if $x_{\tau i}$ is missing measurement **then**
      for $j \in \{1, ..., n_z\}$ **do**
        $C_{ji} \leftarrow 0$ ▷ Set columns of missing measurements to 0
  $K \leftarrow CS^{-1}$ ▷ Compute filter gain K
  $r \leftarrow x_\tau - \bar{x}$ ▷ Compute residual r
  for $i \in \{1, ..., n_x\}$ **do**
    if $x_{\tau i}$ is missing measurement **then**
      $r_i \leftarrow 0$ ▷ Set entries of missing measurements to 0
  $\tilde{\mu}_\tau \leftarrow \bar{z} + Kr$ ▷ Compute filtered mean $\tilde{\mu}_\tau$
  $\tilde{V}_\tau \leftarrow P - KC^T$ ▷ Compute filtered covariance matrix $\tilde{V}_\tau$
  return $\tilde{\mu}_\tau, \tilde{V}_\tau$

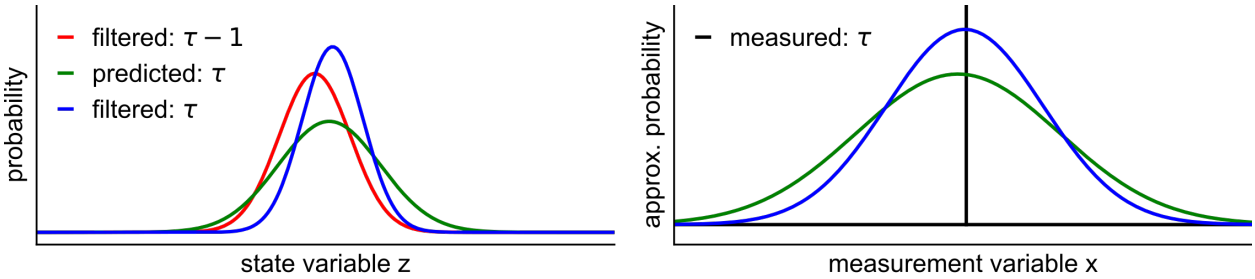

Figure 2.15: One-dimensional example for a single step of the unscented Kalman filter, showing predicted and filtered probability distributions for the state and measurement variables. Based on the distribution of the state variable $z_{\tau-1}$ from the previous time point $\tau - 1$, a prediction for the distribution of the state variable z_τ at the current time point τ is calculated. This predicted distribution is then updated to compute the final output distribution, i.e. the filtered distribution of the state variable z_τ at the current time point τ.

Algorithm 8: The unscented Kalman filter.

function $\mathrm{f}_{\mathrm{ukf}}(\mu_0, V_0, V_z, V_x)$
  $\tilde{\mu}_0 \leftarrow \mu_0$
  $\tilde{V}_0 \leftarrow V_0$
  for $\tau \in \{1, ..., T\}$ **do**
    $\tilde{\mu}_\tau, \tilde{V}_\tau \leftarrow \mathrm{f}_{\mathrm{ukf\_step}}\left(\tilde{\mu}_{\tau-1}, \tilde{V}_{\tau-1}, V_z, V_x\right)$
  return $\tilde{\mu}, \tilde{V}$

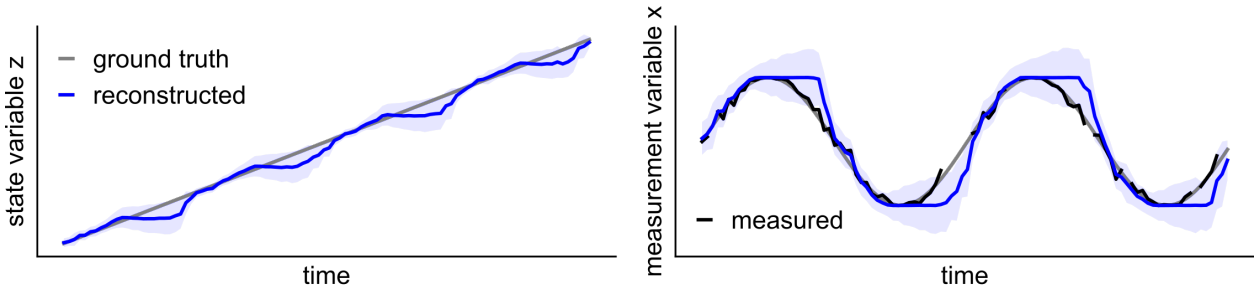

Figure 2.16: One-dimensional example for how the unscented Kalman filter is used to infer state variables z from incomplete measurement variables x for a sequence containing 100 time steps, where 10% of the measurement variables are missing. The underlying state space model is given by the transition function $\mathrm{f}\left(z_{\tau-1}\right) = z_{\tau-1} + \frac{\pi}{25}$ and the emission function $\mathrm{g}\left(z_\tau\right) = \sin\left(z_\tau\right)$ as well as the random variables $z_0 \sim \mathcal{N}\left(0, 0.1\right)$, $\epsilon_z \sim \mathcal{N}\left(0, 0.1\right)$ and $\epsilon_x \sim \mathcal{N}\left(0, 0.1\right)$. The state variables z are assumed to be equal to the inferred expectation values $\tilde{\mu}$, i.e. $z = \tilde{\mu}$, such that the measurement variables x are reconstructed as $x = \sin\left(\tilde{\mu}\right)$. The uncertainty intervals with respect to the state and measurement variables are given by the corresponding standard deviations (blue shading), which are calculated by generating 1000 samples from the inferred normal distribution $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ for each time point τ. Note how the uncertainty intervals increase and the reconstruction quality decreases near the extrema of the emission function.

accurately (Section 2.3.2). In this situation the computations in Algorithm 8 are equivalent to calculating the reduced Kalman gain $K^* \in \mathbb{R}^{n_z \times n_x - 2}$ and using it to update the predicted distribution $\mathcal{N}\left(\bar{z}, P\right)$, such that only accurately detected surface marker locations influence the computation.

To obtain the filter estimates for the expectation values $\tilde{\mu} = \{\tilde{\mu}_0, ..., \tilde{\mu}_T\}$ and covariance matrices $\tilde{V} = \{\tilde{V}_0, ..., \tilde{V}_T\}$ of an entire behavioral sequence, the inference scheme given by Algorithm 7 has to be executed sequentially for each time point $\tau \in \{1, ..., T\}$ according to Algorithm 8 [123]. Computing all inferred expectation values $\tilde{\mu}$ allows for reconstructing z by assuming equality between both quantities, i.e. $z = \tilde{\mu}$. Consequently, reconstructing the measurement variables x is achieved by propagating $\tilde{\mu}$ through the emission function g, i.e. $x = \mathrm{g}\left(\tilde{\mu}\right)$. Since the unscented Kalman filter infers the probability distributions from which the state variables are drawn from, it is also possible to calculate uncertainty intervals with respect to the reconstructed measurement variables, by sampling from the inferred distributions (Figure 2.16).

2.3.5 Bayesian smoothing

In principle, utilizing the unscented Kalman filter already allows for inferring the state variables z (Section 2.3.4). However, the used inference scheme in the proposed pose reconstruction framework is actually a Bayesian smoother, i.e. the unscented Rauch-Tung-Striebel (RTS) smoother [128, 129]. Unlike the unscented Kalman filter, whose inference scheme is solely based on incorporating information from past time points, i.e. $\{0, 1, ..., \tau - 1\}$, the unscented RTS smoother uses information from an entire behavioral sequence, including past and future time points, i.e. $\{0, 1, ..., T\}$, to infer the parameters of the underlying distribution from which a state variable z_τ at time point τ is drawn from. For the unscented RTS smoother these parameters are equal to

Algorithm 9: A single step of the unscented RTS smoother.

function $\mathrm{f}_{\mathrm{uks\_step}}(\tilde{\mu}_\tau, \tilde{V}_\tau, \hat{\mu}_{\tau+1}, \hat{V}_{\tau+1}, V_z)$

$\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$ ▷ Form sigma points $\mathcal{Z}$

$\mathcal{Z} \leftarrow \mathrm{f}\left(\mathcal{Z}\right)$ ▷ Propagate sigma points through transition function f

$\bar{z} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{Z}_i$ ▷ Compute predicted mean $\bar{z}$

$P \leftarrow V_z + \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \bar{z})(\mathcal{Z}_i - \bar{z})^T$ ▷ Compute predicted covariance matrix P

$D \leftarrow \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \tilde{\mu}_\tau)(\mathcal{Z}_i - \bar{z})^T$ ▷ Compute cross-covariance matrix D

$G_\tau \leftarrow DP^{-1}$ ▷ Compute smoother gain G_τ

$\hat{\mu}_\tau \leftarrow \tilde{\mu}_\tau + G_\tau \left(\hat{\mu}_{\tau+1} - \bar{z}\right)$ ▷ Compute smoothed mean $\hat{\mu}_\tau$

$\hat{V}_\tau \leftarrow \tilde{V}_\tau + G_\tau \left(\hat{V}_{\tau+1} - P\right) {G_\tau}^T$ ▷ Compute smoothed covariance matrix $\hat{V}_\tau$

return $\hat{\mu}_\tau, \hat{V}_\tau, G_\tau$

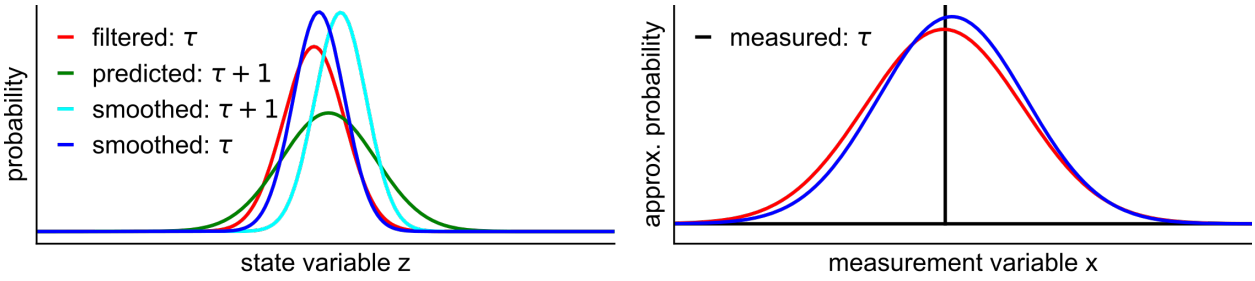

Figure 2.17: One-dimensional example for a single step of the backward pass of the unscented RTS smoother, showing the filtered, predicted and smoothed probability distributions of the state and measurement variables. Based on the filtered distribution of the state variable z_τ from the current time point τ, a prediction for the distribution of the state variable $z_{\tau+1}$ at the subsequent time point $\tau + 1$ is calculated. Computing the differences between this predicted distribution and the smoothed estimate for the distribution of the state variable $z_{\tau+1}$ allows for updating the filtered distribution. This yields the final output distribution, i.e. the smoothed distribution of the state variable z_τ.

the smoothed estimates $\hat{\mu}_\tau$ and $\hat{V}_\tau$ for the distribution's expectation value and covariance matrix respectively.

Using the unscented RTS smoother, $\hat{\mu}_\tau$ and $\hat{V}_\tau$ are inferred by iterating through a given behavioral sequence twice via a forward and a backward pass. In the forward pass the iteration starts at the beginning of the behavioral sequence and terminates at its end. Thus, executing the forward pass is identical to obtaining the filtered estimates $\tilde{\mu}$ and $\tilde{V}$ via the unscented Kalman filter. In the subsequent backward pass the direction of the iteration is reversed, i.e. the iteration starts at the end and terminates at the beginning of the behavioral sequence.

To generate the smoothed estimates $\hat{\mu}_\tau$ and $\hat{V}_\tau$, the unscented RTS smoother uses its own estimates $\hat{\mu}_{\tau+1}$ and $\hat{V}_{\tau+1}$ at the subsequent time point $\tau + 1$, the estimates of the unscented Kalman filter $\tilde{\mu}_\tau$ and $\tilde{V}_\tau$ at the current time point τ and the covariance matrix of the state noise V_z. The respective inference scheme of the unscented RTS smoother's backward pass for a single time point τ is given by Algorithm 9 [123].

Similarly to the unscented Kalman filter, in a single step of the backward pass of the unscented RTS smoother the filter estimate for the distribution of the state variable z_τ at time point τ, i.e. $\mathcal{N}\left(\tilde{\mu}_\tau, \tilde{V}_\tau\right)$, is used to calculate a prediction for the distribution of the state variable $z_{\tau+1}$ at time point $\tau + 1$, i.e. $\mathcal{N}\left(\bar{z}, P\right)$. Subsequently, the smoother gain G is computed and used to update the initial filter estimates $\tilde{\mu}_\tau$ and $\tilde{V}_\tau$, which yields the smoother's final output distribution $\mathcal{N}\left(\hat{\mu}_\tau, \hat{V}_\tau\right)$ of the state variable z_τ (Figure 2.17). In contrast to the unscented Kalman Filter, the unscented RTS smoother does not need access to the measurement variables x.

Algorithm 10: The unscented RTS smoother.

function $\mathrm{f}_{\mathrm{uks}}(\mu_0, V_0, V_z, V_x)$
  $\tilde{\mu}, \tilde{V} \leftarrow \mathrm{f}_{\mathrm{ukf}}\left(\mu_0, V_0, V_z, V_x\right)$
  $\hat{\mu}_T \leftarrow \tilde{\mu}_T$
  $\hat{V}_T \leftarrow \tilde{V}_T$
  for $\tau \in \{T-1, ..., 0\}$ **do**
    $\hat{\mu}_\tau, \hat{V}_\tau, G_\tau \leftarrow \mathrm{f}_{\mathrm{uks\_step}}\left(\tilde{\mu}_\tau, \tilde{V}_\tau, \hat{\mu}_{\tau+1}, \hat{V}_{\tau+1}, V_z\right)$
  return $\hat{\mu}, \hat{V}, G$

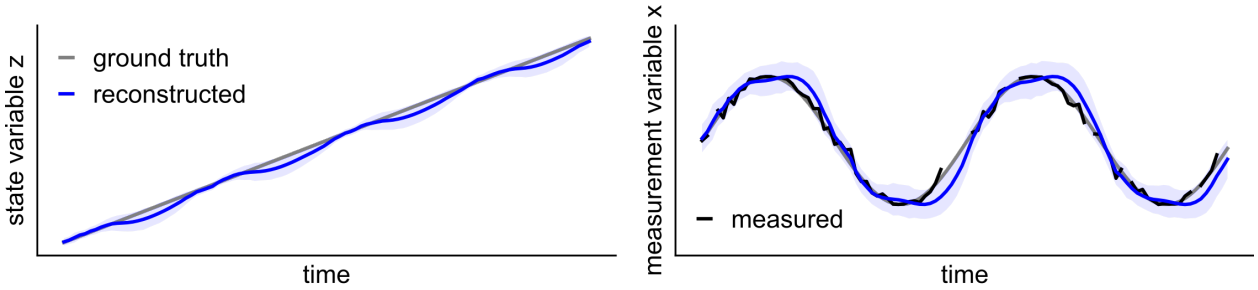

Figure 2.18: One-dimensional example for how the unscented RTS smoother is used to infer state variables z from incomplete measurement variables x. All model parameters and the sequence itself are identical to Figure 2.16. The state variables z are assumed to be equal to the inferred expectation values $\hat{\mu}$, i.e. $z = \hat{\mu}$, such that the measurement variables x are reconstructed as $x = \sin(\hat{\mu})$. The uncertainty intervals with respect to the reconstructed state and measurement variables are given by the corresponding standard deviations (blue shading), which are calculated by generating 1000 samples from the inferred normal distribution $\mathcal{N}\left(\hat{\mu}_\tau, \hat{V}_\tau\right)$ for each time point τ. Note how the uncertainty intervals decrease and the reconstruction quality increases near the extrema of the emission function, when compared to the corresponding results given by the unscented Kalman filter (Figure 2.16).

To obtain the smoother estimates for all expectation values $\hat{\mu} = \{\hat{\mu}_0, ..., \hat{\mu}_T\}$ and covariance matrices $\hat{V} = \{\hat{V}_0, ..., \hat{V}_T\}$ of an entire behavioral sequence, Algroithm 9 has to be executed sequentially according to Algorithm 10 [123]. Using the unscented RTS smoother to infer the state variables yields smoother results with higher reconstruction accuracy for the state and measurement variables, when compared to the unscented Kalman filter (Figure 2.18).

2.3.6 Constraining bone rotations in the state space model

When using a state space model to describe how skeletal poses change over time, constraining the pose-encoding state variables z to guarantee that modeled bone rotations are in agreement with anatomical joint angle limits (Section 2.2.7) becomes more complex. When the skeletal anatomy and poses of an animal subject are learned via gradient descent optimization, joint angle limits are enforced by introducing respective box constraints into the optimization scheme (Section 2.2.9). In contrast to that, using the unscented RTS smoother to infer the state variables z (Section 2.3.5) does not allow for constraining the corresponding smoothed estimates $\hat{\mu}$ to stay within predefined intervals via box constraints. However, box constraints are incorporated into the state space model indirectly, by introducing a redefined state variable $z^*_\tau \in \mathbb{R}^{n_z}$ and a function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}$ according to

$$z^*_\tau = \mathrm{f}\left(z^*_{\tau-1}\right) + \epsilon_z = z^*_{\tau-1} + \epsilon_z \tag{2.27}$$

$$x_\tau = \mathrm{g}\left(\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}\left(z^*_\tau\right)\right) + \epsilon_x. \tag{2.28}$$

Function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}$ is given by Algorithm 11 and used to map each element of the redefined state variable z^*_τ, which corresponds to an entry of a Rodrigues vector r_i, to the respective lower and

Algorithm 11: Constraining bone rotations in the state space model.

function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}(z_\tau^*)$
  $t^* \leftarrow \left(z_{\tau_1}^*, z_{\tau_2}^*, z_{\tau_3}^*\right)^T$ ▷ Obtain normalized global translation t^*
  $r_1^* \leftarrow \left(z_{\tau_4}^*, z_{\tau_5}^*, z_{\tau_6}^*\right)^T$ ▷ Obtain normalized global rotation r_1^*
  for $i \in \{2, .., n_{\mathrm{bone}}\}$ **do**
    $r_i^{**} \leftarrow \left(z_{\tau_{3i+1}}^*, z_{\tau_{3i+2}}^*, z_{\tau_{3i+3}}^*\right)^T$ ▷ Obtain redefined bone rotation r_i^{**}
    for $j \in \{1, .., 3\}$ **do**
      $n \leftarrow \frac{1}{2}\left(\mathrm{f}_{\mathrm{sigmoid}}\left(r_{i_j}^{**}\right) + 1\right)$ ▷ Map $r_{i_j}^{**} \in (-\inf, \inf)$ to $n \in (0, 1)$
      $r_{i_j} \leftarrow b_{0_{ij}} + \left(b_{1_{ij}} - b_{0_{ij}}\right) n$ ▷ Compute bone rotation $r_{i_j} \in \left(b_{0_{ij}}, b_{1_{ij}}\right)$
    $r_i^* \leftarrow \frac{r_i}{s_r}$ ▷ Obtain normalized bone rotation r_i^*
  $z_\tau \leftarrow \mathrm{cat}\left(t^*, r_1^*, r_2^*, ..., r_{n_{\mathrm{bone}}}^*\right)$ ▷ Obtain z_τ via concatenation
  return z_τ

upper bound of the associated bone rotation, such that joint angle limits are enforced. Thus, b_{0ij} and b_{1ij} in Algorithm 11 denote the lower and upper bound, which correspond to entry j of a Rodrigues vector r_i encoding the rotation of bone i. The sigmoid function $\mathrm{f}_{\mathrm{sigmoid}}(y)$ in Algorithm 11 maps an input parameter $y \in \mathbb{R}$ from the interval $(-\inf, \inf)$ to the interval $(-1, 1)$, such that

$$\lim_{y \to -\inf} \mathrm{f}_{\mathrm{sigmoid}}(y) = -1 \tag{2.29}$$

$$\lim_{y \to \inf} \mathrm{f}_{\mathrm{sigmoid}}(y) = 1. \tag{2.30}$$

To ensure that the sigmoid function $\mathrm{f}_{\mathrm{sigmoid}}$ is as similar as possible to the identity function, the following properties are advantageous:

$$\mathrm{f}_{\mathrm{sigmoid}}(0) = 0 \tag{2.31}$$

$$\mathrm{f}_{\mathrm{sigmoid}}{}'(0) = 1, \tag{2.32}$$

with $\mathrm{f}_{\mathrm{sigmoid}}{}'(y) = \frac{\partial}{\partial y}\mathrm{f}_{\mathrm{sigmoid}}(y)$. Here, the intention for minimizing the discrepancy between $\mathrm{f}_{\mathrm{sigmoid}}$ and the identity function is to reduce the effect of the additional non-linearities introduced by $\mathrm{f}_{\mathrm{sigmoid}}$, such that the modified state space model (Equation 2.27 and 2.28) remains as similar as possible to the original state space model (Equation 2.18 and 2.19). For the proposed pose reconstruction framework the following functions represent candidates for the sigmoid function $\mathrm{f}_{\mathrm{sigmoid}}$ [120, 121]:

$$\mathrm{f}_{\mathrm{atan}}(y) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}y\right) \tag{2.33}$$

$$\mathrm{f}_{\mathrm{logistic}}(y) = \frac{2}{1 + \exp(-2y)} - 1 \tag{2.34}$$

$$\mathrm{f}_{\mathrm{erf}}(y) = \mathrm{erf}\left(\frac{\sqrt{\pi}}{2}y\right), \tag{2.35}$$

where the error function erf is defined as

$$\mathrm{erf}(y) = \frac{2}{\sqrt{\pi}}\int_0^y \exp\left(-a^2\right) da. \tag{2.36}$$

Here, candidate function $\mathrm{f}_{\mathrm{erf}}$ is the least different from the identity function in the sense that it minimizes the cumulative error $\int \mathrm{abs}\left(\mathrm{f}_{\mathrm{sigmoid}}(y) - y\right) dy$, when compared to the other two candidate

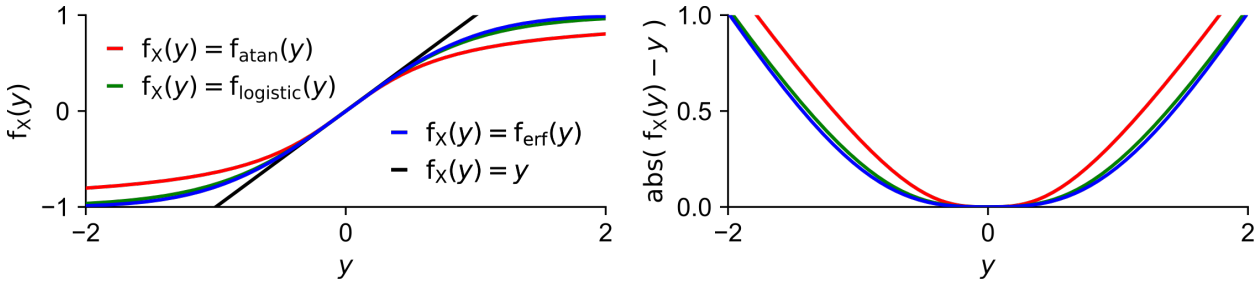

Figure 2.19: Curve progressions (left) and corresponding error values (right) in the vicinity of the origin for the sigmoid candidate functions $\mathrm{f}_{\mathrm{atan}}$, $\mathrm{f}_{\mathrm{logistic}}$ and $\mathrm{f}_{\mathrm{erf}}$, which allow for enforcing joint angle limits in the state space model. The shown error values indicate discrepancies between the identity function and the candidate functions. Note how every input value from the interval $(-\inf, \inf)$ is mapped to the output interval $(-1, 1)$, which is used to enforce box constraints corresponding to joint angle limits. Additionally, note how the error values for candidate function $\mathrm{f}_{\mathrm{erf}}$ are overall smaller compared to those from candidate functions $\mathrm{f}_{\mathrm{atan}}$ and $\mathrm{f}_{\mathrm{logistic}}$.

functions (Figure 2.19). Thus, the required sigmoid function is defined as $\mathrm{f}_{\mathrm{sigmoid}}\,(y) = \mathrm{f}_{\mathrm{erf}}\,(y)$. Consequently, instead of using the unscented RTS smoother to infer the state variables $z = \{z_0, ..., z_T\}$ in the original state space model, the smoother is deployed in the context of the modified state space model to infer the redefined state variables $z^* = \{z_0^*, ..., z_T^*\}$ in order to account for joint angle limits in the proposed pose estimation framework.

2.4 Learning the model's probabilistic hyper-parameters

The previous section introduced a state space model to describe how skeletal poses change over time within a behavioral sequence (Section 2.3.1). Additionally, the previous section described how the pose-encoding state variables $z = \{z_0, z_1, ..., z_T\}$ in the state space model are inferred from the corresponding measurement variables $x = \{x_1, x_2, ..., x_T\}$ via a Bayesian smoother, i.e. an unscented RTS smoother (Section 2.3.5), such that the resulting reconstructed bone configurations are guaranteed to comply with physiological joint angle limits (Section 2.3.6).

When the unscented RTS smoother is used to infer the state variables z, the dynamics of changing skeletal poses are determined by the model parameters $\Theta = \{\mu_0, V_0, V_z, V_x\}$. These model parameters are given by the expectation value μ_0 and the covariance matrix V_0 of the underlying normal distribution, from which the initial state variable z_0 is drawn from, as well as the covariance matrices V_z and V_x, which determine the magnitude of the transition and measurement noise ϵ_z and ϵ_x respectively. Thus, given that there exists a reasonably good estimate for the model parameters Θ, deploying the unscented RTS smoother to infer the state variables z from the measurement variables x allows for reconstructing the unknown skeletal poses in a behavioral sequence. However, while the measurement variables x are automatically obtained via a trained deep neural network (Section 2.3.2), the model parameters Θ are unknown in practice.

In principle, the initial parameter values for the expectation value μ_0 and the covariance matrix V_0 are updated by the unscented RTS smoother, yielding the smoothed estimates $\hat{\mu}_0$ and $\hat{V}_0$. However, since the resulting values for $\hat{\mu}_0$ and $\hat{V}_0$ are highly dependent on how μ_0 and V_0 are initialized, it is essential to actually use reasonable initial parameter values to obtain a decent reconstruction quality with respect to the estimated skeletal poses. Furthermore, the parameter values for the covariance matrices V_z and V_x are not updated at all by the unscented RTS smoother. Consequently, obtaining reasonable values for these two quantities is even more challenging. Particularly, finding reasonable values for V_z and V_x is non-trivial, since the associated transition and measurement noise are assumed to be normally distributed, which only approximates the true dynamics governing how an animal behaves and adjusts its skeletal pose. In fact, the circumstance that all

random variables involved in the state space model, i.e. the initial state variable z_0, the transition noise ϵ_z and the measurement noise ϵ_x, are normally distributed is a mere model assumption. The actual mechanics and associated distributions, which would accurately describe the dynamics of skeletal pose changes, are complex and unknown. Thus, an appropriate method for estimating the model parameters Θ of the state space model is required, such that the true dynamics of changing skeletal poses can at least be approximated reasonably well.

A method for learning the model parameters Θ is given by the iterative expectation-maximization (EM) algorithm [130], which allows for calculating a set of model parameters, such that a lower bound of the state space model's evidence, i.e. the evidence lower bound (ELBO), is maximized [120, 121]. Each iteration of the EM algorithm contains an expectation step (E-step), in which the state variables z are inferred, and a maximization step (M-step), in which a new set of model parameters is calculated, such that the ELBO is maximized. Thus, the E-step is identical to infering the state variables z via the unscented RTS smoother, whereas the computations of the M-step are aimed at solving a respective optimization problem in closed-form [125].

This section covers the theoretical bases of the EM algorithm (Section 2.4.1) and describes how it is used to learn the model parameters Θ of the state space model (Section 2.4.2). Additionally, this section describes how convergence of the EM algorithm is defined (Section 2.4.3) and how the algorithm is implemented in the context of animal pose estimation (Section 2.4.4).

2.4.1 The expectation-maximization algorithm

While the general form of the EM algorithm was first introduced by Arthur Dempster, Nan Laird and Donald Rubin in 1977 [130], this section follows the concepts and notations stated by Christopher Bishop [120] and Kevin Murphy [121].

The general idea behind the EM algorithm is to incrementally increase the state space model's marginal likelihood $\mathrm{p}(x)$, i.e. the model evidence, by maximizing the ELBO. To understand the theoretical background of this maximization scheme, it is beneficial to recall two fundamental concepts on which probability theory is build on. Firstly, the model's joint distribution $\mathrm{p}(x, z)$ is equal to the product of the model's likelihood $\mathrm{p}(x|z)$ and prior $\mathrm{p}(z)$ [120, 121]:

$$\mathrm{p}(x, z) = \mathrm{p}(x|z)\,\mathrm{p}(z)\,. \tag{2.37}$$

Secondly, the mutual dependency of the model's marginal likelihood $\mathrm{p}(x)$, posterior $\mathrm{p}(z|x)$, likelihood $\mathrm{p}(x|z)$ and prior $\mathrm{p}(z)$ is given by Bayes' theorem [120, 121]:

$$\mathrm{p}(z|x)\,\mathrm{p}(x) = \mathrm{p}(x|z)\,\mathrm{p}(z)\,. \tag{2.38}$$

To start building an understanding for how the EM algorithm works, at first an arbitrary probability density function $\mathrm{q}(z)$ over the state variables z is defined, such that the following statement is true by definition:

$$\int \mathrm{q}(z)\,dz = 1. \tag{2.39}$$

Multiplying Equation 2.39 with an arbitrary constant c yields

$$c \int \mathrm{q}(z)\,dz = \int c\,\mathrm{q}(z)\,dz = c. \tag{2.40}$$

However, replacing the constant c with a function, which is independent of the state variables z, is a valid mathematical operation and can be performed without loss of generality. Choosing this function to be the model's marginal log-likelihood $\ln \mathrm{p}(x)$ yields

$$\int \mathrm{q}(z) \ln \mathrm{p}(x)\,dz = \ln \mathrm{p}(x)\,. \tag{2.41}$$

Here, it is advantageous to memorize that the marginal log-likelihood $\ln \mathrm{p}(x)$ is independent of the probability density function $\mathrm{q}(z)$ to understand the upcoming theoretical considerations, which are essential for understanding the EM algorithm.

Now, Equations 2.37 and 2.38 are used to obtain a relationship between the marginal log-likelihood $\ln \mathrm{p}(x)$, the Kullback–Leibler (KL) divergence $\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p})$ and the ELBO $\mathcal{L}$. Starting from Equation 2.41, the following equations are derived:

$$\ln \mathrm{p}(x) = \int \mathrm{q}(z) \ln \mathrm{p}(x)\, dz \tag{2.42}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x)\, \mathrm{p}(x)}{\mathrm{p}(z|x)} dz \tag{2.43}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x|z)\, \mathrm{p}(z)}{\mathrm{p}(z|x)} dz \tag{2.44}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x, z)}{\mathrm{p}(z|x)} dz \tag{2.45}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x, z)\, \mathrm{q}(z)}{\mathrm{p}(z|x)\, \mathrm{q}(z)} dz \tag{2.46}$$

$$= \int \mathrm{q}(z) \left(\ln \frac{\mathrm{p}(x, z)}{\mathrm{q}(z)} - \ln \frac{\mathrm{p}(z|x)}{\mathrm{q}(z)} \right) dz \tag{2.47}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x, z)}{\mathrm{q}(z)} dz - \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x)}{\mathrm{q}(z)} dz \tag{2.48}$$

$$= \mathcal{L} + \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}), \tag{2.49}$$

with $\mathcal{L} = \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x,z)}{\mathrm{q}(z)} dz$ and $\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) = -\int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x)}{\mathrm{q}(z)} dz$. The KL divergence is a distance measure between the probability density functions q and p. Consequently, it is always larger or equal to zero:

$$\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \geq 0, \tag{2.50}$$

with equality $\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) = 0$ if $\mathrm{q} = \mathrm{p}$ [120, 121]. When the ELBO $\mathcal{L}$ is added to Equation 2.50 and the result is combined with the derived definition of the marginal log-likelihood $\ln \mathrm{p}(x)$ (Equation 2.49), it becomes clear that the ELBO $\mathcal{L}$ is indeed a lower bound of the marginal log-likelihood $\ln \mathrm{p}(x)$:

$$\ln \mathrm{p}(x) = \mathcal{L} + \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \geq \mathcal{L}. \tag{2.51}$$

Now, noticing that $\ln \mathrm{p}(x)$ is actually conditioned on the model parameters Θ finally yields

$$\ln \mathrm{p}(x|\Theta) = \mathcal{L}(\mathrm{q}, \Theta) + \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \tag{2.52}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x, z|\Theta)}{\mathrm{q}(z)} dz - \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x, \Theta)}{\mathrm{q}(z)} dz \tag{2.53}$$

$$= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x, \Theta)\, \mathrm{p}(x|\Theta)}{\mathrm{q}(z)} dz - \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x, \Theta)}{\mathrm{q}(z)} dz \tag{2.54}$$

$$= \left(\int \mathrm{q}(z) \ln \mathrm{p}(x|\Theta)\, dz + \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x, \Theta)}{\mathrm{q}(z)} dz \right) - \int \mathrm{q}(z) \ln \frac{\mathrm{p}(z|x, \Theta)}{\mathrm{q}(z)} dz \tag{2.55}$$

$$= \left(\int \mathrm{q}(z) \ln \mathrm{p}(x|\Theta)\, dz - \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \right) + \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \tag{2.56}$$

$$\geq \mathcal{L}(\mathrm{q}, \Theta) \tag{2.57}$$

$$\mathcal{L}(\mathrm{q}, \Theta) = \int \mathrm{q}(z) \ln \mathrm{p}(x|\Theta)\, dz - \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) \tag{2.58}$$

$$= \ln \mathrm{p}(x|\Theta) - \mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}). \tag{2.59}$$

These last equations mark a cornerstone for developing an understanding of the EM algorithm, in which the E- and M-step are sequentially executed in every iteration of the algorithm.

In the E-step the model parameters Θ are held constant and the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$ is maximized with respect to $\mathrm{q}(z)$, i.e. the probability density functions $\mathrm{p}(z|x, \Theta_k)$ of the state variables z are inferred, given a current estimate Θ_k of the model parameters Θ at iteration k of the EM algorithm. As a consequence $\mathrm{q}(z)$ becomes equal to $\mathrm{p}(z|x, \Theta_k)$, i.e. $\mathrm{q}(z) = \mathrm{p}(z|x, \Theta_k)$, which results in the KL divergence $\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p})$ becoming equal to zero, i.e. $\mathrm{KL}(\mathrm{q} \,||\, \mathrm{p}) = \mathrm{KL}(\mathrm{p} \,||\, \mathrm{p}) = 0$, and the marginal log-likelihood $\ln \mathrm{p}(x|\Theta_k)$ becoming equal to the ELBO $\mathcal{L}(\mathrm{q}, \Theta_k)$, i.e. $\ln \mathrm{p}(x|\Theta_k) = \mathcal{L}(\mathrm{q}, \Theta_k)$. For understanding this step it is helpful to remember that the KL divergence is always greater than zero (Equation 2.50) and that the marginal log-likelihood $\ln \mathrm{p}(x|\Theta_k)$ is actually independent of the probability density function $\mathrm{q}(z)$.

In the subsequent M-step $\mathrm{q}(z)$ is held constant and $\mathcal{L}(\mathrm{q}, \Theta)$ is maximized with respect to the model parameters Θ in order to obtain a new estimate Θ_{k+1} of the model parameters Θ for the next iteration $k+1$ of the EM algorithm. Here, the fact that the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$ is maximized with respect to the model parameters Θ yields the inequality $\mathcal{L}(\mathrm{q}, \Theta_{k+1}) \geq \mathcal{L}(\mathrm{q}, \Theta_k)$. Therefore, the M-step leads to an increased marginal log-likelihood:

$$\ln \mathrm{p}(x|\Theta_{k+1}) \geq \mathcal{L}(\mathrm{q}, \Theta_{k+1}) \geq \mathcal{L}(\mathrm{q}, \Theta_k) = \ln \mathrm{p}(x|\Theta_k). \tag{2.60}$$

Thus, the starting point in the M-step is the following:

$$\begin{aligned}
\ln \mathrm{p}(x|\Theta) &= \mathcal{L}(\mathrm{q}, \Theta) && (2.61)\\
&= \int \mathrm{q}(z) \ln \frac{\mathrm{p}(x, z|\Theta)}{\mathrm{q}(z)} dz && (2.62)\\
&= \int \mathrm{p}(z|x, \Theta_k) \ln \frac{\mathrm{p}(x, z|\Theta)}{\mathrm{p}(z|x, \Theta_k)} dz && (2.63)\\
&= \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(x, z|\Theta)\, dz - \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(z|x, \Theta_k)\, dz && (2.64)\\
&= \mathcal{Q}(\Theta, \Theta_k) - \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(z|x, \Theta_k)\, dz, && (2.65)
\end{aligned}$$

with $\mathcal{Q}(\Theta, \Theta_k) = \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(x, z|\Theta)\, dz$. Since the objective of the M-step is to optimize the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$ with respect to Θ, the latter term is omitted due to its independence of Θ. Consequently, it is sufficient to maximize function $\mathcal{Q}(\Theta, \Theta_k)$ in order to maximize the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$. Furthermore, function $\mathcal{Q}(\Theta, \Theta_k)$ has the form of an expectation value, such that $\mathcal{Q}(\Theta, \Theta_k)$ is obtained by calculating the expectation of $\ln \mathrm{p}(x, z|\Theta)$ with respect to z:

$$\mathcal{Q}(\Theta, \Theta_k) = \mathbb{E}\left[\ln \mathrm{p}(x, z|\Theta)\right], \tag{2.66}$$

where it is a requirement that x and Θ_k are given quantities, which is fulfilled in the M-step. Finally, this leads to the sole purpose of the M-step, i.e. maximizing $\mathcal{Q}(\Theta, \Theta_k)$ with respect to Θ in order to obtain a new estimate Θ_{k+1} of the model parameters Θ:

$$\Theta_{k+1} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta_k). \tag{2.67}$$

By successively executing the E-step and the M-step for several iterations, the model parameters Θ of the state space model are learned via the EM algorithm. The learned model parameters are then used to reconstruct skeletal poses by inferring the state variables z via the unscented RTS smoother.

2.4.2 The maximization step

In the previously described M-step the function $\mathcal{Q}(\Theta, \Theta_k)$ is maximized in order to compute a new estimate $\Theta_{k+1} = \{\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1}\}$ of the model parameters Θ of the state space model for iteration $k+1$ of the EM algorithm. The details of this computation are described in this section, which follows the concepts stated by Juho Kokkala, Arno Solin and Simo Särkkä [125].

In the state space model all state variables z fulfill the Markov property, i.e. each state variable z_τ only depends on the previous one $z_{\tau-1}$ [120, 121]. This sequential structure is exploited when maximizing function $\mathcal{Q}(\Theta, \Theta_k)$ in the M-step. Particularly, it leads to the following formulation of the model's joint distribution [120, 121, 125]:

$$\mathrm{p}(x, z) = \mathrm{p}(z_0) \prod_{\tau=1}^{T} \mathrm{p}(z_\tau | z_{\tau-1}) \, \mathrm{p}(x_\tau | z_\tau) \,. \tag{2.68}$$

Taking the logarithm of the joint distribution $\mathrm{p}(x, z)$ and accounting for the fact that it is actually conditioned on the model parameters Θ yields

$$\ln \mathrm{p}(x, z|\Theta) = \ln \mathrm{p}(z_0|\mu_0, V_0) + \sum_{\tau=1}^{T} \ln \mathrm{p}(z_\tau | z_{\tau-1}, V_z) + \sum_{\tau=1}^{T} \ln \mathrm{p}(x_\tau | z_\tau, V_x) \,. \tag{2.69}$$

However, to maximize $\mathcal{Q}(\Theta, \Theta_k)$ the expectation value of $\ln \mathrm{p}(x, z|\Theta)$ needs to be considered:

$$\begin{aligned} \mathcal{Q}(\Theta, \Theta_k) &= \mathbb{E}\left[\ln \mathrm{p}(x, z|\Theta)\right] && (2.70) \\ &= \mathbb{E}\left[\ln \mathrm{p}(z_0|\mu_0, V_0)\right] + \sum_{\tau=1}^{T} \mathbb{E}\left[\ln \mathrm{p}(z_\tau | z_{\tau-1}, V_z)\right] + \sum_{\tau=1}^{T} \mathbb{E}\left[\ln \mathrm{p}(x_\tau | z_\tau, V_x)\right] && (2.71) \\ &= I_0 + I_z + I_x, && (2.72) \end{aligned}$$

with $I_0 = \mathbb{E}\left[\ln \mathrm{p}(z_0|\mu_0, V_0)\right]$, $I_z = \sum_{\tau=1}^{T} \mathbb{E}\left[\ln \mathrm{p}(z_\tau | z_{\tau-1}, V_z)\right]$ and $I_x = \sum_{\tau=1}^{T} \mathbb{E}\left[\ln \mathrm{p}(x_\tau | z_\tau, V_x)\right]$. When acknowledging that all random variables in the state space model are assumed to be normally distributed, i.e. $z_\tau \sim \mathcal{N}\left(\hat{\mu}_\tau, \hat{V}_\tau\right)$, it becomes clear that computing $\mathcal{Q}(\Theta, \Theta_k)$ only involves evaluating the expectation values of log-transformed normal distributions (Appendix A.1). Consequently, simplified terms for I_0, I_z and I_x can be obtained by using the outputs of the unscented RTS smoother, i.e. the expectation values $\hat{\mu}$, the covariance matrices $\hat{V}$ and the smoother gains G. Respective parameter values for these three quantities are computed in the preceding E-step based on the current estimate $\Theta_k = \{\mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}\}$ of the model parameters Θ at iteration k of the EM algorithm. [125].

A respectively simplified expression for I_0 is given by

$$\begin{aligned} I_0 &= -\frac{1}{2} \ln \det(2\pi V_0) - \frac{1}{2} \operatorname{tr}\left({V_0}^{-1} \mathbb{E}\left[(z_0 - \mu_0)(z_0 - \mu_0)^T \right] \right) && (2.73) \\ &= -\frac{1}{2} \ln \det(2\pi V_0) - \frac{1}{2} \operatorname{tr}\left({V_0}^{-1} \left(\hat{V}_0 + (\hat{\mu}_0 - \mu_0)(\hat{\mu}_0 - \mu_0)^T \right) \right) . && (2.74) \end{aligned}$$

To obtain a simplified expression for I_z it is necessary to use pairwise sigma points $\mathcal{P}_\tau$, since there are two different random variables involved, i.e. $z_\tau \sim \mathcal{N}\left(\hat{\mu}_\tau, \hat{V}_\tau\right)$ and $z_{\tau-1} \sim \mathcal{N}\left(\hat{\mu}_{\tau-1}, \hat{V}_{\tau-1}\right)$, when evaluating the expectation values of the log-transformed normal distributions in I_z [125]. For each of the T transition steps in the state space model the pairwise mean vector $\check{\mu}_\tau \in \mathbb{R}^{2n_z}$ is generated by concatenating $\hat{\mu}_\tau$ and $\hat{\mu}_{\tau-1}$:

$$\check{\mu}_\tau = \operatorname{cat}(\hat{\mu}_\tau, \hat{\mu}_{\tau-1}) = \operatorname{cat}\begin{pmatrix} \hat{\mu}_\tau \\ \hat{\mu}_{\tau-1} \end{pmatrix} . \tag{2.75}$$

Likewise, the pairwise covariance matrix $\check{V}_\tau \in \mathbb{R}^{2n_z \times 2n_z}$ is generated as

$$\check{V}_\tau = \mathrm{cat}\begin{pmatrix} \hat{V}_\tau & \hat{V}_\tau {G_{\tau-1}}^T \\ G_{\tau-1}\hat{V}_\tau & \hat{V}_{\tau-1} \end{pmatrix}, \tag{2.76}$$

such that the upper left, upper right, lower left and lower right entries of matrix $\check{V}_\tau$ are given by $\hat{V}_\tau \in \mathbb{R}^{n_z \times n_z}$, $\hat{V}_\tau {G_{\tau-1}}^T \in \mathbb{R}^{n_z \times n_z}$, $G_{\tau-1}\hat{V}_\tau \in \mathbb{R}^{n_z \times n_z}$ and $\hat{V}_{\tau-1} \in \mathbb{R}^{n_z \times n_z}$ respectively. Using $\check{\mu}_\tau$ and $\check{V}_\tau$ allows for deploying the unscented transform $\mathrm{f_{ut}}$ (Algorithm 6) to calculate the pairwise sigma points $\mathcal{P}_\tau$:

$$\mathcal{P}_\tau = \mathrm{cat}\begin{pmatrix} \mathcal{B}_\tau & \mathcal{A}_\tau \end{pmatrix} = \mathrm{f_{ut}}\left(\check{\mu}_\tau, \check{V}_\tau\right). \tag{2.77}$$

Here, the incomplete pairwise sigma points $\mathcal{B}_\tau \in \mathbb{R}^{4n_z+1 \times n_z}$ and $\mathcal{A}_\tau \in \mathbb{R}^{4n_z+1 \times n_z}$ are defined, such that concatenating them along their second dimension yields $\mathcal{P}_\tau \in \mathbb{R}^{4n_z+1 \times 2n_z}$. The respective weights $\check{w} \in \mathbb{R}^{4n_z+1}$, which are required within the unscented transform $\mathrm{f_{ut}}$, are given as follows:

$$\check{w}_1 = 0 \tag{2.78}$$

$$\check{w}_i = \frac{1}{4n_z} \forall i \in \{2, ..., 4n_z + 1\}. \tag{2.79}$$

A simplified expression for I_z is then given by

$$I_z = -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{1}{2}\sum_{\tau=1}^{T}\mathrm{tr}\left({V_z}^{-1}\mathbb{E}\left[\left(z_\tau - z_{\tau-1}\right)\left(z_\tau - z_{\tau-1}\right)^T\right]\right) \tag{2.80}$$

$$= -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{T}{2}\mathrm{tr}\left({V_z}^{-1}\left(\frac{1}{T}\sum_{\tau=1}^{T}\mathbb{E}\left[\left(z_\tau - z_{\tau-1}\right)\left(z_\tau - z_{\tau-1}\right)^T\right]\right)\right) \tag{2.81}$$

$$\approx -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{T}{2}\sum_{\tau=1}^{T}\mathrm{tr}\left({V_z}^{-1}\left(\frac{1}{T}\sum_{i=1}^{4n_z+1}\check{w}_i\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)^T\right)\right). \tag{2.82}$$

In contrast, simplifying the expression for I_x only requires using the ordinary sigma points $\mathcal{Z}_\tau = \mathrm{f_{ut}}\left(\hat{\mu}_t, \hat{V}_t\right)$, which need to be propagated through the emission function g (Algorithm 5):

$$I_x = -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{1}{2}\sum_{\tau=1}^{T}\mathrm{tr}\left({V_x}^{-1}\mathbb{E}\left[\left(x_\tau - \mathrm{g}\left(z_\tau\right)\right)\left(x_\tau - \mathrm{g}\left(z_\tau\right)\right)^T\right]\right) \tag{2.83}$$

$$= -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{T}{2}\mathrm{tr}\left({V_x}^{-1}\left(\frac{1}{T}\sum_{\tau=1}^{T}\mathbb{E}\left[\left(x_\tau - \mathrm{g}\left(z_\tau\right)\right)\left(x_\tau - \mathrm{g}\left(z_\tau\right)\right)^T\right]\right)\right) \tag{2.84}$$

$$\approx -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{T}{2}\sum_{\tau=1}^{T}\mathrm{tr}\left({V_x}^{-1}\left(\frac{1}{T}\sum_{i=1}^{2n_z+1}w_i\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)^T\right)\right). \tag{2.85}$$

To obtain the new estimate Θ_{k+1} of the model parameters Θ for iteration $k+1$ of the EM algorithm, $\mathcal{Q}\left(\Theta, \Theta_k\right)$ still needs to be differentiated with respect to μ_0, V_0, V_z and V_x. Setting the resulting derivatives to zero and solving them for μ_0, V_0, V_z and V_x respectively finally yields the new estimate Θ_{k+1} of the model parameters Θ. The expressions for these derivatives are given

as follows (Appendix A.2):

$$\frac{\partial}{\partial \mu_0}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{\partial}{\partial \mu_0} I_0 \tag{2.86}$$

$$= {V_0}^{-1}\left(\hat{\mu}_0 - \mu_0\right) \tag{2.87}$$

$$\frac{\partial}{\partial V_0}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{\partial}{\partial V_0} I_0 \tag{2.88}$$

$$= -\frac{1}{2}{V_0}^{-1} + \frac{1}{2}{V_0}^{-1}\left(\hat{V}_0 + \left(\hat{\mu}_0 - \mu_0\right)\left(\hat{\mu}_0 - \mu_0\right)^T\right){V_0}^{-1} \tag{2.89}$$

$$\frac{\partial}{\partial V_z}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{\partial}{\partial V_z} I_z \tag{2.90}$$

$$= -\frac{T}{2}{V_z}^{-1} + \frac{T}{2}\sum_{\tau=1}^{T}{V_z}^{-1}\left(\frac{1}{T}\sum_{i=1}^{4n_z+1}\check{w}_i\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)^T\right){V_z}^{-1} \tag{2.91}$$

$$\frac{\partial}{\partial V_x}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{\partial}{\partial V_x} I_x \tag{2.92}$$

$$= -\frac{T}{2}{V_x}^{-1} + \frac{T}{2}\sum_{\tau=1}^{T}{V_x}^{-1}\left(\frac{1}{T}\sum_{i=1}^{2n_z+1} w_i\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)^T\right){V_x}^{-1}. \tag{2.93}$$

Setting these derivatives to zero and solving for μ_0, V_0, V_z and V_x respectively yields

$$\mu_0 = \hat{\mu}_0 \tag{2.94}$$

$$V_0 = \hat{V}_0 + \left(\hat{\mu}_0 - \mu_0\right)\left(\hat{\mu}_0 - \mu_0\right)^T = \hat{V}_0 \tag{2.95}$$

$$V_z = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{i=1}^{4n_z+1}\check{w}_i\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)^T \tag{2.96}$$

$$V_x = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{i=1}^{2n_z+1} w_i\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)\left(x_\tau - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)\right)^T. \tag{2.97}$$

The parameter values for $\mu_{0,k+1}$, $V_{0,k+1}$ and $V_{z,k+1}$ are then given by Equation 2.94, 2.95 and 2.96 respectively. To obtain parameter values for $V_{x,k+1}$ Equation 2.97 still needs to be modified to account for missing measurements, i.e. incomplete entries of the measurement variables x (Section 2.3.2). Besides, in the proposed pose reconstruction framework the covariance matrix V_x of the measurement noise is restricted to only have diagonal entries. Consequently, only the diagonal entries $\mathrm{diag}\left(V_x\right)$ of V_x are updated in the M-step. Thus, the final solution for a single diagonal entry $j \in \{1, ..., n_x\}$ of V_x is given by

$$\mathrm{diag}\left(V_x\right)_j = \frac{1}{T_j}\sum_{\tau=1}^{T}\delta_{\tau j}\sum_{i=1}^{2n_z+1} w_i\Big(x_{\tau j} - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)_j\Big)^2, \tag{2.98}$$

where δ_{tj} indicates if at time point τ entry j of measurement variable x_τ is associated with a missing measurement, i.e. $\delta_{\tau j} = 0$, or not, i.e. $\delta_{\tau j} = 1$, and T_j is the total number of observed measurements for entry j in the entire behavioral sequence, i.e. $T_j = \sum_{\tau=1}^{T}\delta_{\tau j}$. Therefore, the parameter values for the diagonal entries of $V_{x,k+1}$ are given by Equation 2.98.

Successively learning new model parameters in each iteration of the EM algorithm results in gradually improving the inference results for the probability distributions, from which the state variables are drawn from. As a consequence, the reconstruction quality with respect to the measurement variables is improved as well (Figure 2.20).

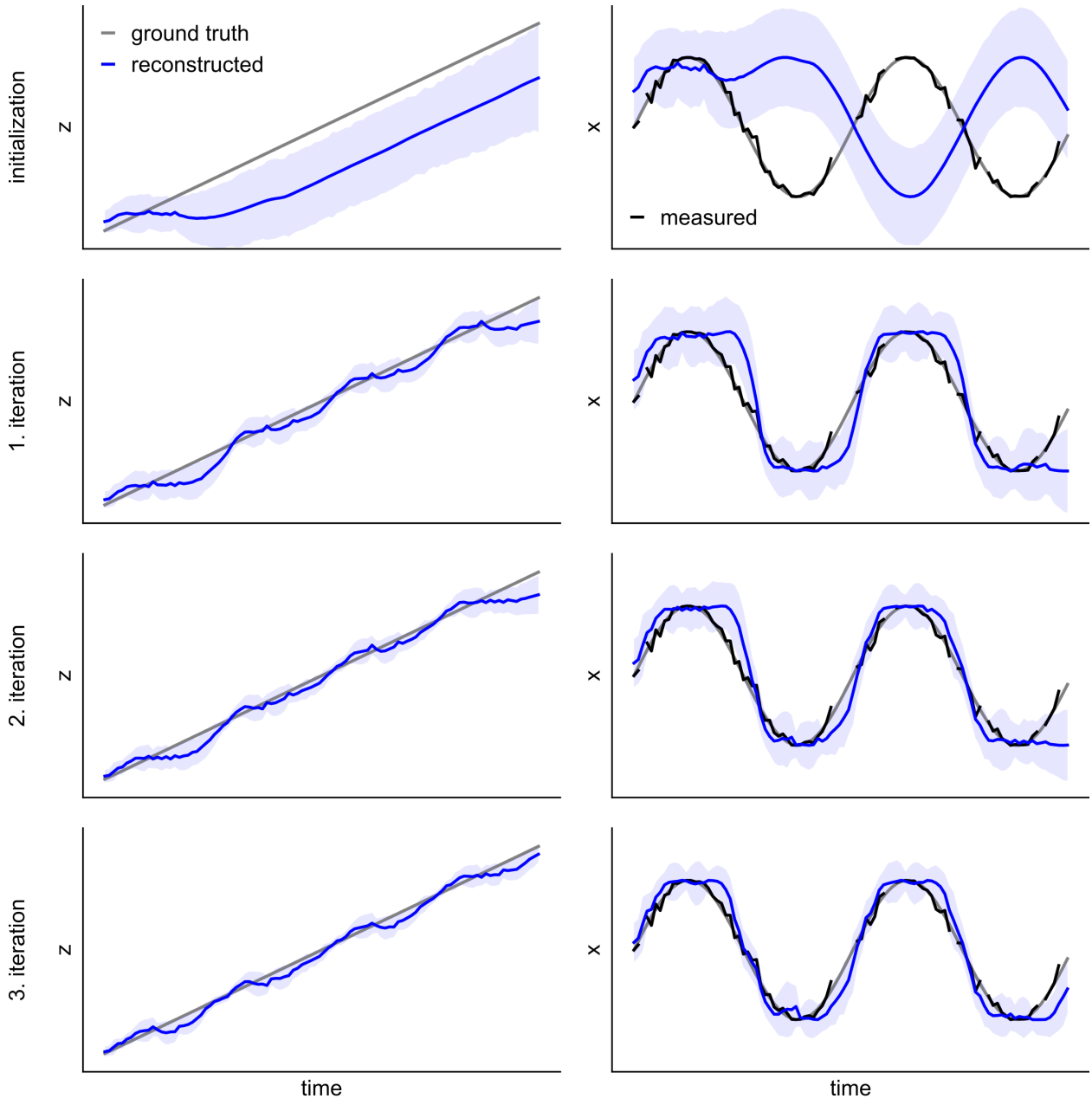

Figure 2.20: One-dimensional example for how the EM algorithm is used to learn unknown model parameters $\Theta = \{\mu_0, V_0, V_z, V_x\}$ of a state space model. Here, all model parameters and the sequence itself are identical to Figure 2.16 and 2.18. The EM algorithm is initialized with $\mu_{0,0} = \frac{\pi}{4}$, $V_{0,0} = 0.5$, $V_{z,0} = 0.5$ and $V_{x,0} = 0.5$. The uncertainty intervals with respect to z and x (blue shading) are calculated as described in Figure 2.18. In each iteration of the EM algorithm a new estimate of the model parameters is computed. This estimate is used to infer the underlying normal distributions of the state variables z via the unscented RTS smoother. Based on these inferred normal distributions the measurement variables x are reconstructed as described in Figure 2.18. Note how the reconstruction quality increases with each iteration.

2.4.3 Convergence criterion

To determine at which iteration the EM algorithm is going to be terminated, it is necessary to define a respective convergence criterion. One possibility for evaluating if the EM algorithm converged is given by calculating the change in the learned model parameters Θ_k in each iteration k and assuming convergence once it falls below a predefined threshold [120].

In the proposed pose reconstruction framework the vectors $\Delta\mu_0 \in \mathbb{R}^{n_z}$, $\Delta \operatorname{diag}(V_0) \in \mathbb{R}^{n_z}$, $\Delta \operatorname{diag}(V_z) \in \mathbb{R}^{n_z}$ and $\Delta \operatorname{diag}(V_x) \in \mathbb{R}^{n_x}$, which contain the relative changes of the model parameters μ_0, V_0, V_z and V_x respectively, are computed in each iteration as follows:

$$\Delta\mu_{0_i} = \operatorname{abs}\left(\frac{\mu_{0,k_i} - \mu_{0,k-1_i}}{\mu_{0,k-1_i}}\right) \; \forall\, i \in \{1, ..., n_z\} \tag{2.99}$$

$$\Delta \operatorname{diag}(V_0)_i = \operatorname{abs}\left(\frac{V_{0,k_{ii}} - V_{0,k-1_{ii}}}{V_{0,k-1_{ii}}}\right) \; \forall\, i \in \{1, ..., n_z\} \tag{2.100}$$

$$\Delta \operatorname{diag}(V_z)_i = \operatorname{abs}\left(\frac{V_{z,k_{ii}} - V_{z,k-1_{ii}}}{V_{z,k-1_{ii}}}\right) \; \forall\, i \in \{1, ..., n_z\} \tag{2.101}$$

$$\Delta \operatorname{diag}(V_x)_i = \operatorname{abs}\left(\frac{V_{x,k_{ii}} - V_{x,k-1_{ii}}}{V_{x,k-1_{ii}}}\right) \; \forall\, i \in \{1, ..., n_x\}. \tag{2.102}$$

Here, only the diagonal entries of the covariance matrices V_0, V_z and V_x are considered, since a fraction of their off-diagonal entries is expected to be zero, e.g. due to uncorrelated bone rotations. Furthermore, for V_x all off-diagonal entries are zero by definition, since it is modeled as a diagonal matrix (Section 2.4.2). Using these vectors allows for generating an additional vector $\Delta v \in \mathbb{R}^{3n_z+n_x}$, which contains all relative changes, via concatenation:

$$\Delta v = \operatorname{cat}\left(\Delta\mu_0, \Delta \operatorname{diag}(V_0), \Delta \operatorname{diag}(V_z), \Delta \operatorname{diag}(V_x)\right). \tag{2.103}$$

The EM algorithm is assumed to have reached convergence, when the mean $\Delta\bar{v}$ of Δv falls below a threshold ϵ_{tol}:

$$\Delta\bar{v} = \frac{1}{3n_z + n_x} \sum_{i=1}^{3n_z+n_x} \Delta v_i < \epsilon_{\text{tol}} = 0.05. \tag{2.104}$$

By deploying this convergence criterion, the unknown model parameters Θ of the state space model are learned, which allows for inferring the state variables z and reconstructing the measurement variables x (Figure 2.21).

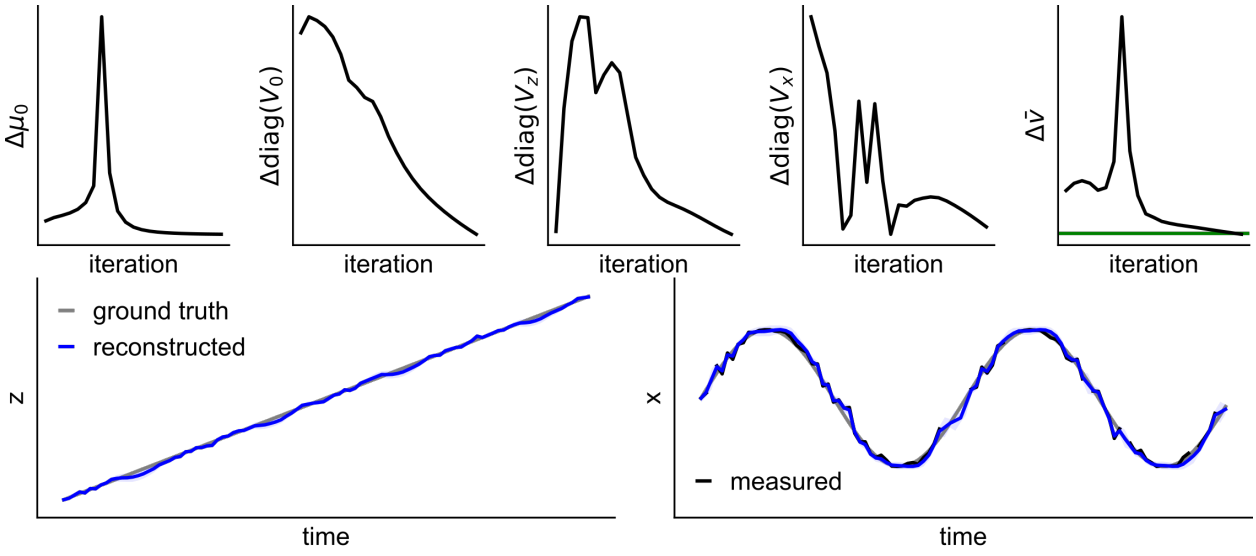

Figure 2.21: Convergence behavior of the EM algorithm for a one-dimensional example (top row) as well as final reconstruction results for the state and measurement variable z and x respectively (bottom row). Here, all model parameters and the sequence itself are identical to Figure 2.16, 2.18 and 2.20. The uncertainty intervals with respect to z and x (blue shading) are calculated as described in Figure 2.18. Convergence is reached after 22 iterations, such that $\Delta\bar{v} = 0.25\left(\Delta\mu_0 + \Delta \operatorname{diag}(V_0) + \Delta \operatorname{diag}(V_z) + \Delta \operatorname{diag}(V_x)\right) < \epsilon_{\text{tol}} = 0.05$. The respective threshold value $\epsilon_{\text{tol}} = 0.05$ is highlighted (upper right, green line). Note how the measured and reconstructed values for x are almost identical.

While the choice for the value of ϵ_{tol} is arbitrary in principle, setting it to 0.05 leads to reasonable results in the context of animal pose reconstruction (Chapter 3).

2.4.4 Implementation

To deploy the EM algorithm within the context of the proposed pose reconstruction framework, an initial estimate $\Theta_0 = \{\mu_{0,0}, V_{0,0}, V_{z,0}, V_{x,0}\}$ of the model parameters Θ is used to initiate the first iteration of the EM algorithm. To obtain $\mu_{0,0}$ the objective function given in Equation 2.17 is minimized, while the bone lengths as well as the surface marker positions are kept constant. Here, n_{time} is set to 1 to ensure that only the first time point of the respective behavioral sequence is included in the optimization scheme. Furthermore, the covariance matrices $V_{0,0}$, $V_{x,0}$ and $V_{z,0}$ are initialized as diagonal matrices, whose diagonal and off-diagonal entries equal 0.001 and 0 respectively.

To learn the model parameters μ_0, V_0, V_x and V_z the EM algorithm is then executed according to Algorithm 12. Here, in accordance to the concepts stated in Section 2.4.2 and 2.4.3, function f_{M}, given by Algorithm 13, performs the M-step and function f_{tol}, given by Algorithm 14, computes the mean $\Delta\bar{v}$ of the relative changes of the model parameters Θ. The implementation of the EM algorithm allows for reconstructing skeletal poses of freely-moving animals with high temporal as well as spatial resolution with respect to the resulting three-dimensional joint positions (Figure 2.22, Chapter 3).

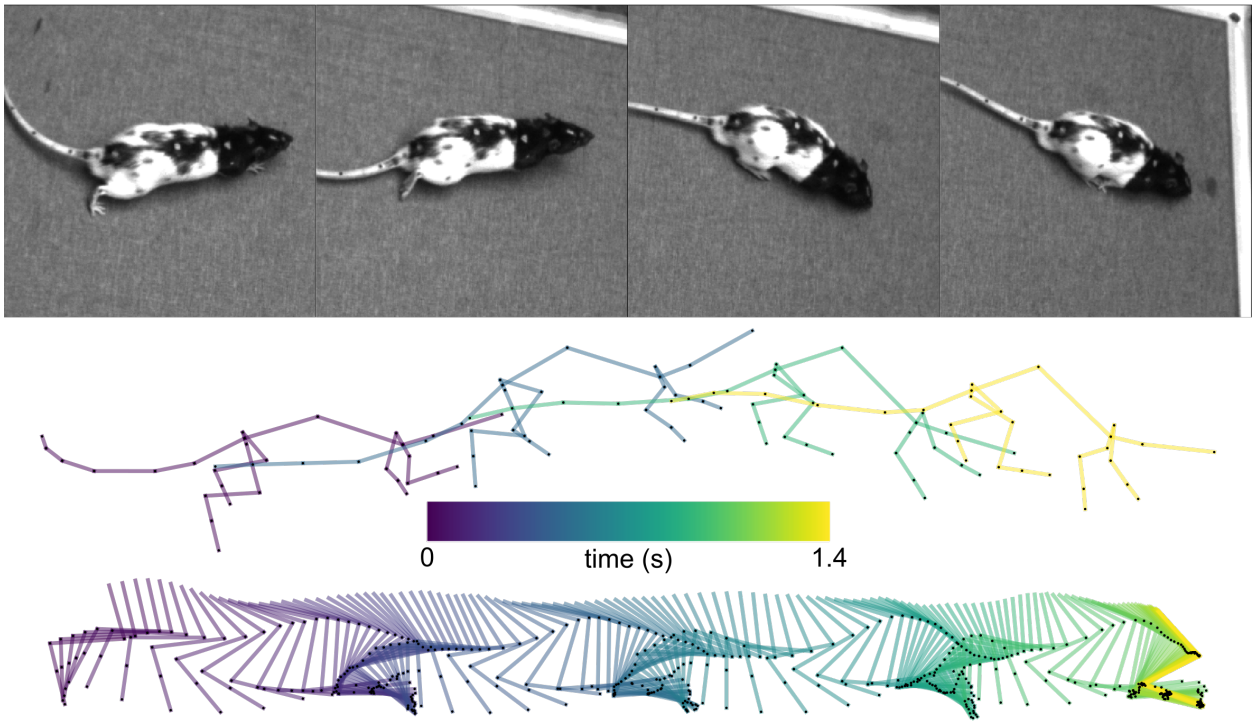

Figure 2.22: Example for how the EM algorithm is used to reconstruct skeletal poses of a freely-moving animal. Based on two-dimensional images recorded at different time points (top), three-dimensional joint positions are reconstructed (center). Resulting reconstructed skeletal poses are consistent in time and allow for extracting fine-grained motions of individual limbs, e.g. the right hind limb (bottom). The shown skeletal poses (center) correspond to the time points at which the displayed images were captured. The shown motion trajectory of the right hind limb (bottom) also corresponds to the behavioral sequence depicted in the displayed images. Here, the additional skeletal leg poses were reconstructed based on in-between time points, where further images were recorded.

Algorithm 12: The EM algorithm.

function $\mathrm{f_{EM}}(\mu_{0,0}, V_{0,0}, V_{z,0}, V_{x,0})$
  $k \leftarrow 0$ ▷ Initialize iteration number k
  $\Delta\bar{v} \leftarrow \inf$ ▷ Initialize changes in Θ
  while $\Delta\bar{v} \geq \epsilon_{\mathrm{tol}}$ **do**
    $\hat{\mu}, \hat{V}, G \leftarrow \mathrm{f_{uks}}\left(\mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}\right)$ ▷ Perform E-step
    $\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1} \leftarrow \mathrm{f_M}\left(\hat{\mu}, \hat{V}, G\right)$ ▷ Perform M-step
    $k \leftarrow k + 1$ ▷ Increase iteration number k
    $\Delta\bar{v} \leftarrow \mathrm{f_{tol}}\left(\mu_{0,k-1}, V_{0,k-1}, V_{z,k-1}, V_{x,k-1}, \mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}\right)$ ▷ Compute changes in Θ
  return $\mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}$

Algorithm 13: The M-step of the EM algorithm.

function $\mathrm{f_M}(\hat{\mu}, \hat{V}, G)$
  for $\tau \in \{1, ..., T\}$ **do**
    $\mathrm{cat}\begin{pmatrix}\mathcal{B}_\tau & \mathcal{A}_\tau\end{pmatrix} \leftarrow \mathrm{f_{ut}}\left(\mathrm{cat}\begin{pmatrix}\hat{\mu}_\tau \\ \hat{\mu}_{\tau-1}\end{pmatrix}, \mathrm{cat}\begin{pmatrix}\hat{V}_\tau & \hat{V}_\tau {G_{\tau-1}}^T \\ G_{\tau-1}\hat{V}_\tau & \hat{V}_{\tau-1}\end{pmatrix}\right)$
    $\mathcal{Z}_\tau \leftarrow \mathrm{f_{ut}}\left(\hat{\mu}_\tau, \hat{V}_\tau\right)$
  $\mu_{0,k+1} \leftarrow \hat{\mu}_0$
  $V_{0,k+1} \leftarrow \hat{V}_0$
  $V_{z,k+1} \leftarrow \frac{1}{T}\sum_{\tau=1}^{T}\sum_{i=1}^{4n_z+1} \check{w}_i \left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)\left(\mathcal{B}_{\tau i} - \mathcal{A}_{\tau i}\right)^T$
  for $j \in \{1, ..., n_x\}$ **do**
    ${V_{x,k+1}}_{jj} \leftarrow \frac{1}{T_j}\sum_{\tau=1}^{T} \delta_{tj} \sum_{i=1}^{2n_z+1} w_i \left(x_{\tau j} - \mathrm{g}\left(\mathcal{Z}_{\tau i}\right)_j\right)^2$
  return $\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1}$

Algorithm 14: Computing the convergence criterion of the EM algorithm.

function $\mathrm{f_{tol}}(\mu_{0,k-1}, V_{0,k-1}, V_{z,k-1}, V_{x,k-1}, \mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k})$
  for $i \in \{1, ..., n_z\}$ **do**
    $\Delta\mu_{0_i} \leftarrow \mathrm{abs}\left(\frac{\mu_{0,k_i} - \mu_{0,k-1_i}}{\mu_{0,k-1_i}}\right)$
    $\Delta\,\mathrm{diag}\left(V_0\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{0,k}}_{ii} - {V_{0,k-1}}_{ii}}{{V_{0,k-1}}_{ii}}\right)$
    $\Delta\,\mathrm{diag}\left(V_z\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{z,k}}_{ii} - {V_{z,k-1}}_{ii}}{{V_{z,k-1}}_{ii}}\right)$
  for $i \in \{1, ..., n_x\}$ **do**
    $\Delta\,\mathrm{diag}\left(V_x\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{x,k}}_{ii} - {V_{x,k-1}}_{ii}}{{V_{x,k-1}}_{ii}}\right)$
  $\Delta v \leftarrow \mathrm{cat}\left(\Delta\mu_0, \Delta\,\mathrm{diag}\left(V_0\right), \Delta\,\mathrm{diag}\left(V_z\right), \Delta\,\mathrm{diag}\left(V_x\right)\right)$
  $\Delta\bar{v} \leftarrow \frac{1}{3n_z+n_x}\sum_{i=1}^{3n_z+n_x} \Delta v_i$
  return $\Delta\bar{v}$

Chapter 3

Results

3.1 Evaluating learned skeleton anatomies

To evaluate the performance of the numerical optimization scheme used for learning the skeletal anatomy (Section 2.2.9), video data of six freely-moving animals was recorded, whereas surface markers were painted onto the body of each animal in a symmetrical pattern prior to the recordings (Section 2.2.2). The six animals were rats of different sizes, whose weights spanned an order of magnitude, i.e. the individual animals weighted 174 g (animal #1), 178 g (animal #2), 71 g (animal #3), 72 g (animal #4), 735 g (animal #5) and 699 g (animal #6). To generate the video data, all animals were allowed to move freely in an open area of size 105x80 cm^2 (animal #1 and #2) or 60x60 cm^2 (animal #3, #4, #5 and #6), while their behavior was recorded via four different overhead cameras. All overhead cameras were calibrated prior to the experiments, recorded the images synchronously, had a resolution of 1280x1024 px^2 and were operated at either 100 Hz (animal #1 and #2) or 200 Hz (animal #3, #4, #5 and #6) with an acquisition time of 2.5 ms.

After the video data was recorded the two-dimensional positions of the surface markers were manually located and labeled in each camera view for all animals, such that either each 50th (animals #1 and #2) or 200th (animals #3, #4, #5 and #6) time point of the recorded behavioral sequences was subjected to labeling. In total, this procedure yielded 2404 (animal #1 and #2), 752 (animal #3), 1100 (animal #4), 992 (animal #5) and 1128 (animal #6) annotated frames, corresponding to 601 (animal #1 and #2), 188 (animal #3), 275 (animal #4), 248 (animal #5) and 282 (animal #6) different time points in the recorded behavioral sequences. The labeled two-dimensional surface marker locations were then used to learn the anatomy-encoding variables, i.e. the reduced bone lengths l^* (Section 2.2.4) and the reduced joint-to-marker-translation vectors v^* (Section 2.2.5), for each animal by minimizing the objective function given in Equation 2.17. This allowed for reconstructing the skeletal anatomy of each animal (Figure 3.1).

To evaluate the accuracy of the learned skeletal anatomies, three-dimensional scans of all animals were obtained via magnetic resonance imaging (MRI), such that each MRI scan had a resolution of 0.4x0.4x0.4 mm^3. In order to identify the three-dimensional locations of the painted surface markers in the MRI data, half-spherical MRI markers were attached to the bodies of the animals prior to the scans, such that the position of each MRI marker coincided with the location of a painted surface marker. Ground truth data on joint and surface marker positions of the animals was then obtained by manually labeling the positions of individual joints and MRI markers in the MRI scans (Figure 3.2). However, during one of the scans a single MRI marker fell off the body of an animal (right hind paw marker, animal #1), such that the three-dimensional location of the corresponding painted surface marker was not recoverable. In this particular case the three-dimensional position on the animal's body closest to the location of the seceded MRI marker was labeled instead. Furthermore, the ground truth joint positions of the left and right hind paw joints

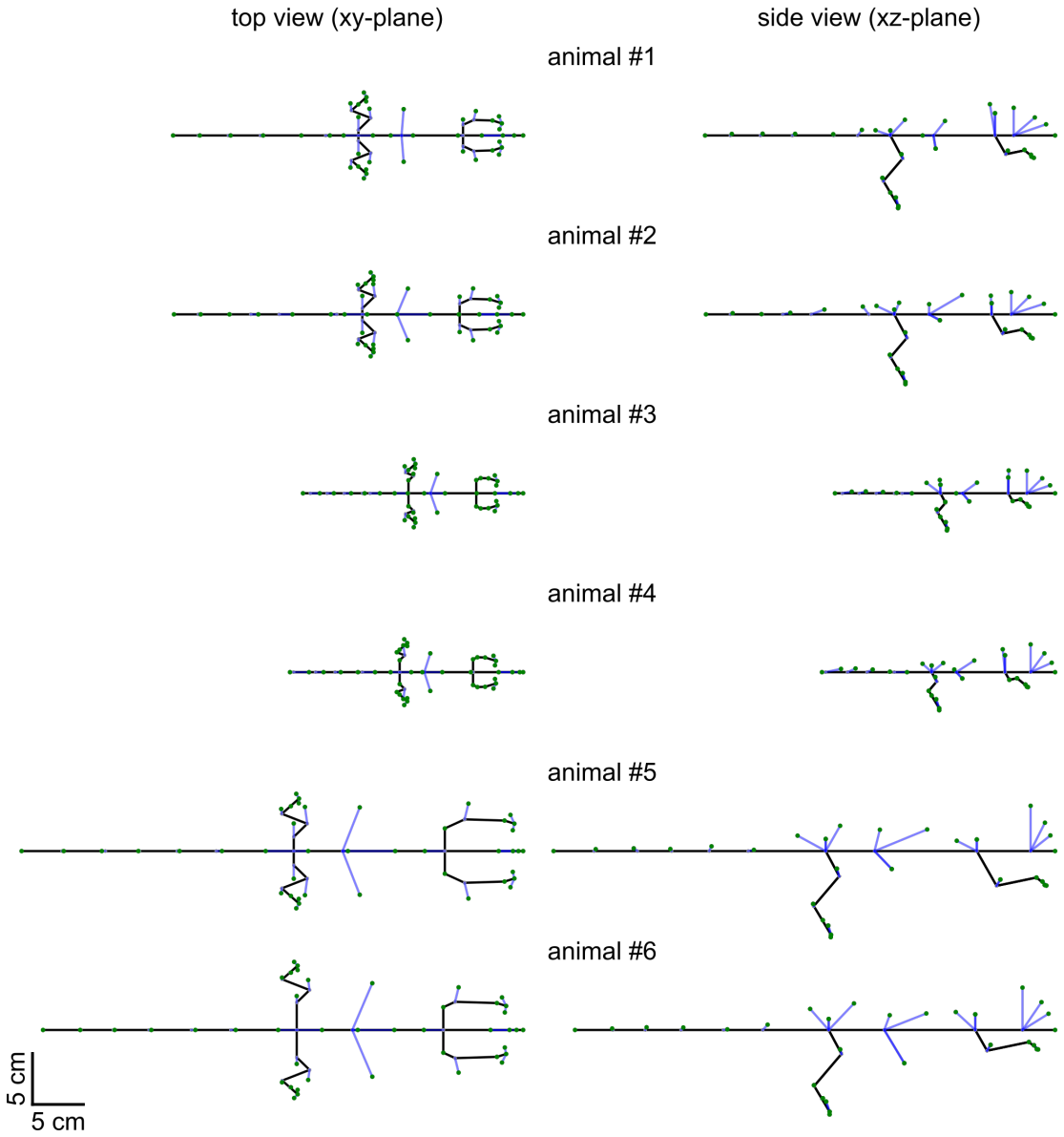

Figure 3.1: Learned skeletal anatomies of six different animal subjects seen in the xy- (left column) and xz-plane (right column). The individual bones (black lines), three-dimensional surface marker locations (green dots) as well as the rigid connections between them (blue lines) are shown. To highlight the different bone lengths and surface marker positions, the skeletal poses of all animal subjects are identical.

could not be identified in the MRI scan (12 joint locations in total, 2 for each animal), such that the missing locations were assumed to be identical to the positions of the corresponding left and right hind paw markers.

After all joint and surface marker positions were labeled, the obtained three-dimensional surface marker locations were used to align learned skeleton anatomies with ground truth skeletal poses for each animal by minimizing an optimization problem similar to the one given by Equation 2.17:

$$\underset{t^*,r^*}{\arg\min} \sum_{j=1}^{n_{\text{marker}}} \left|\left| \mathrm{f}_{\text{surface}}\left(s_t\ t^*, s_r\ r^*, l, v\right)_j - m_j \right|\right|^2, \tag{3.1}$$

with $l = \mathrm{f}_{\mathrm{l}^* \to \mathrm{l}}\left(l^*\right)$ (Algorithm 3), $v = \mathrm{f}_{\mathrm{v}^* \to \mathrm{v}}\left(v^*\right)$ (Algorithm 4) and m_j the labeled three-dimensional

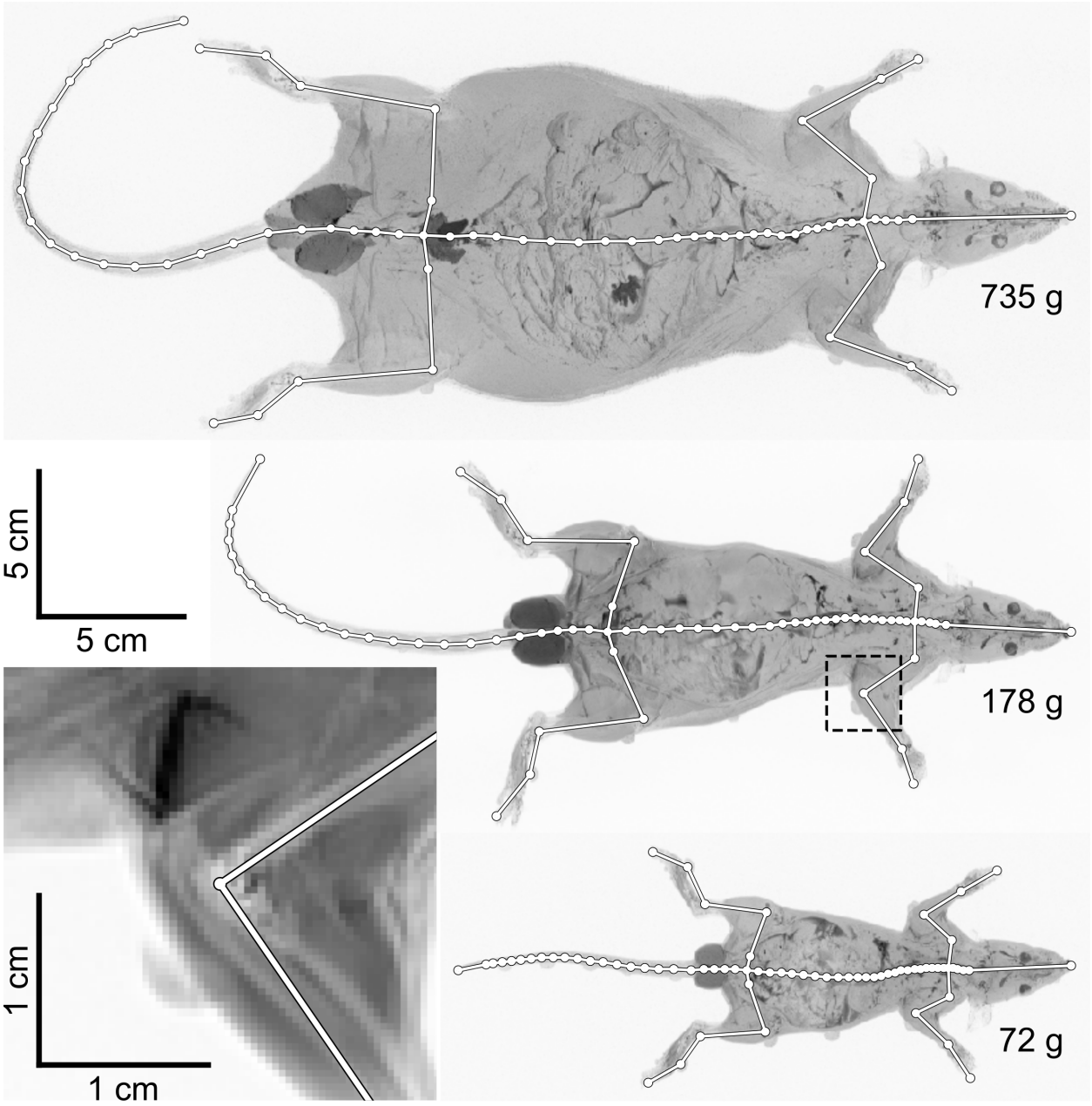

Figure 3.2: MRI scans of three differently-sized animal subjects (maximum projection), where manually labeled bone (white lines) and joint (white dots) positions are highlighted. Additionally, an enlarged section (dashed black box) provides a more detailed view of the right elbow joint of the medium-sized animal subject (lower left, mean projection). Note the half-spherical MRI marker in the enlarged area. Also note how the weights of the displayed animal subjects differ by an order of magnitude.

position of marker j. Aligning learned skeleton anatomies with ground truth poses enabled reconstructing the static skeletal pose of each animal during the MRI scan.

To determine correspondences between the spine joints in the modeled skeletal anatomies and the MRI scans, vertebrae were counted in the MRI scans, such that each spine segment in the skeleton model matched its anatomical counterpart [131]. As a result the modeled spine segments were equivalent to the spinal column's cervical, thoracic and lumbar sections as well as the sacrum.

After skeletal anatomies were learned, the resulting bone lengths in the skeleton model were

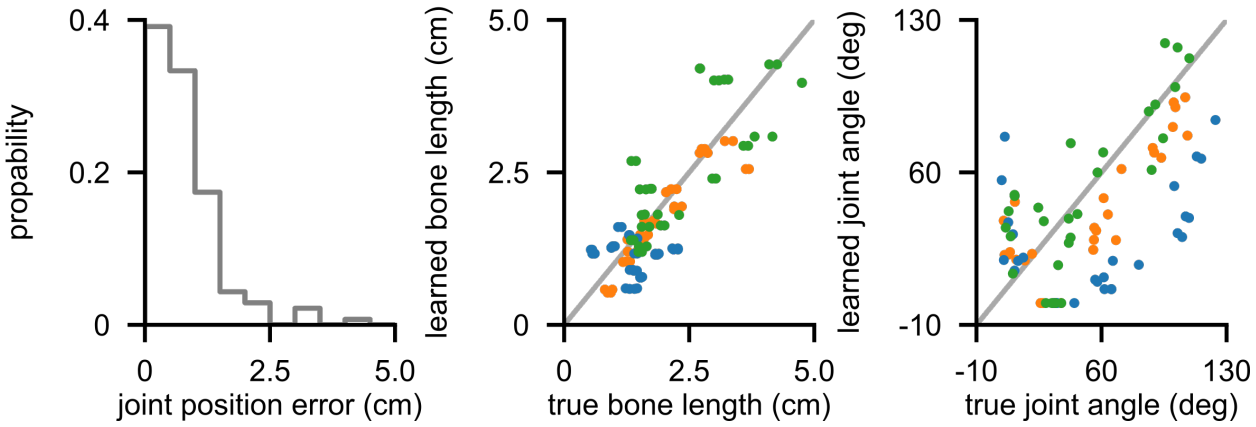

Figure 3.3: Histogram of joint position error probabilities (left) as well as scatter plots showing learned vs. true limb bone lengths (center) and limb joint angles (right). The different colors represent small (blue), medium (orange) and large (green) animal sizes (blue: 71 g and 72 g, orange: 174 g and 178 g, green: 699 g and 735 g).

compared to those measured in the MRI scans. Additionally, reconstructing the skeletal poses of the animals during the MRI scans allowed for computing the distribution of the three-dimensional joint position errors as well as the discrepancies between learned and true limb joint angles (Figure 3.3). Here, joint position errors were defined as the Euclidean distances between reconstructed and true joint positions in three-dimensional space, taking into account all joints except those along the tail. Furthermore, when comparing learned with true bone lengths and joint angles, only the bones and joints of the limbs were considered, since the number of modeled bones along the spine and tail in the skeleton model was limited to ten. Consequently, there did not necessarily exist a one-to-one correspondence between modeled and anatomical spine and tail joints, since a multitude of anatomical spine and tail bones were approximated with only a few bones in the skeleton model (Section 2.2.1). Thus, including the spine and tail joints for the comparison of learned and measured bone lengths and joint angles was infeasible.

The majority of the computed joint position errors were below 1 cm (138 joint positions in total, joint position error: 0.79 +/- 0.69 and 0.65 cm [avg. +/- s.d. and median]). Additionally, inferred limb bone lengths and limb bone angles were not significantly different from those measured in the MRI scans (108 limb bone lengths in total, range of measured limb bone lengths: 0.53 cm to 4.76 cm, limb bone length error: 0.46 +/- 0.34 and 0.36 cm [avg. +/- s.d. and median], Spearman correlation coefficient: 0.75, two-tailed p-value testing for non-correlation: 5.00×10^{-21}; 84 limb bone angles in total, range of measured limb bone angles: 4.13 deg to 123.77 deg, limb bone angle error: 27.80 +/- 18.98 and 26.72 deg [avg. +/- s.d. and median], Spearman correlation coefficient: 0.47, two-tailed p-value testing for non-correlation: 5.29×10^{-6}).

Altogether, this evaluation demonstrated that learned skeleton anatomies were accurate when compared with true skeleton anatomies, while accuracy was invariant to different animal sizes. Additionally, the fact that accurate three-dimensional joint positions were reconstructed for the static animal poses during the MRI scans indicated that joint positions can be successfully recovered, when three-dimensional ground truth surface marker locations are available or, equivalently, when they are inferred from correctly detected two-dimensional surface marker locations in recorded video data.

3.2 Assessing how constraints affect pose reconstruction accuracy

The proposed pose reconstructing framework enforces anatomical constraints in the form of physiological joint angle limits (Section 2.2.7 and 2.3.6) as well as temporal constraints, which are implicitly incorporated via a state space model (Section 2.3.1) and the usage of an unscented RTS smoother (Section 2.3.5). To evaluate the influence of both constraint types on the pose reconstruction accuracy, behavioral sequences of freely-moving animals were recorded, while additional ground truth data on paw positions and orientations was obtained using a frustrated total internal reflection (FTIR) imaging approach [132, 133].

Particularly, the same six animals, which were described in Section 3.1, were recorded via four different overhead cameras, while they were allowed to move freely on a transparent FTIR plate of size 60x60 cm^2. Two infrared LED-strips were mounted at the edges of the FTIR plate, such that infrared light could propagate through the plate from two opposing sites. This construction made the paws of the animals appear bright in images recorded via additional cameras located underneath the FTIR plate, whenever the paws were placed in close vicinity to the surface of the plate, e.g. when they touched it. To ensure that the respective camera sensors only detected infrared light emitted from the LED-strips, infrared filters were mounted onto the lenses of the cameras. All cameras used in these experiments were calibrated beforehand, recorded the images synchronously, had a resolution of 1280x1024 px^2 and were operated at 200 Hz with an acquisition time of 2.5 ms. Using this approach, 29 behavioral sequences were recorded, which accumulated to 36250 frames per camera and a total duration of 181.25 s.

To allow for the automated detection of two-dimensional surface marker locations in the images, which were recorded via the four overhead cameras, an individual DeepLabCut network was trained for each animal. Particularly, 4068 (animal #1), 3980 (animal #2), 752 (animal #3), 1100 (animal #4), 992 (animal #5) and 1128 (animal #6) images were used for training. These training images were not part of the data set, which was analyzed to assess the effect of the enforced constraints on the pose reconstruction accuracy. The automatically detected surface marker locations were then used to reconstruct skeletal poses of freely-moving animals via the proposed pose reconstruction framework.

Additional ground truth information on the positioning of individual paws was obtained by manually labeling the two-dimensional locations of paw center points, fingers and toes in every 40th image, which was recorded with the underneath cameras. Consequently, manually labeled anatomical landmarks, i.e. paw center points, fingers and toes, corresponded to modeled surface markers, i.e. the left- as well as right-sided front and hind paw markers as well as the finger and toe markers #1 to #3 (Figure 2.9). By comparing the two-dimensional positions of manually labeled anatomical landmarks with the corresponding surface marker positions on the FTIR plate obtained by reconstructing poses, discrepancies with respect to ground truth and reconstructed paw positions and orientations were computed, i.e. the position and angle errors of the paws (Figure 3.4).

To obtain the two-dimensional positions of the reconstructed surface markers on the FTIR plate, their inferred three-dimensional positions were projected on the FTIR plate by dismissing their last component. This projection was valid, since the coordinate system of the FTIR plate was the used frame of reference during the reconstruction of skeletal poses, such that inferred three-dimensional surface marker positions were given in the coordinate system of the FTIR plate. The corresponding two-dimensional positions of the manually labeled anatomical landmarks were obtained by projecting the respective labels into three-dimensional space using a pinhole camera model (Section 2.1.2) and subsequently computing the intersections of the resulting three-dimensional lines with the FTIR plate.

To finally assess the effect of the anatomical and temporal constraints on the pose reconstruction accuracy, skeletal poses were reconstructed via four different reconstruction models,

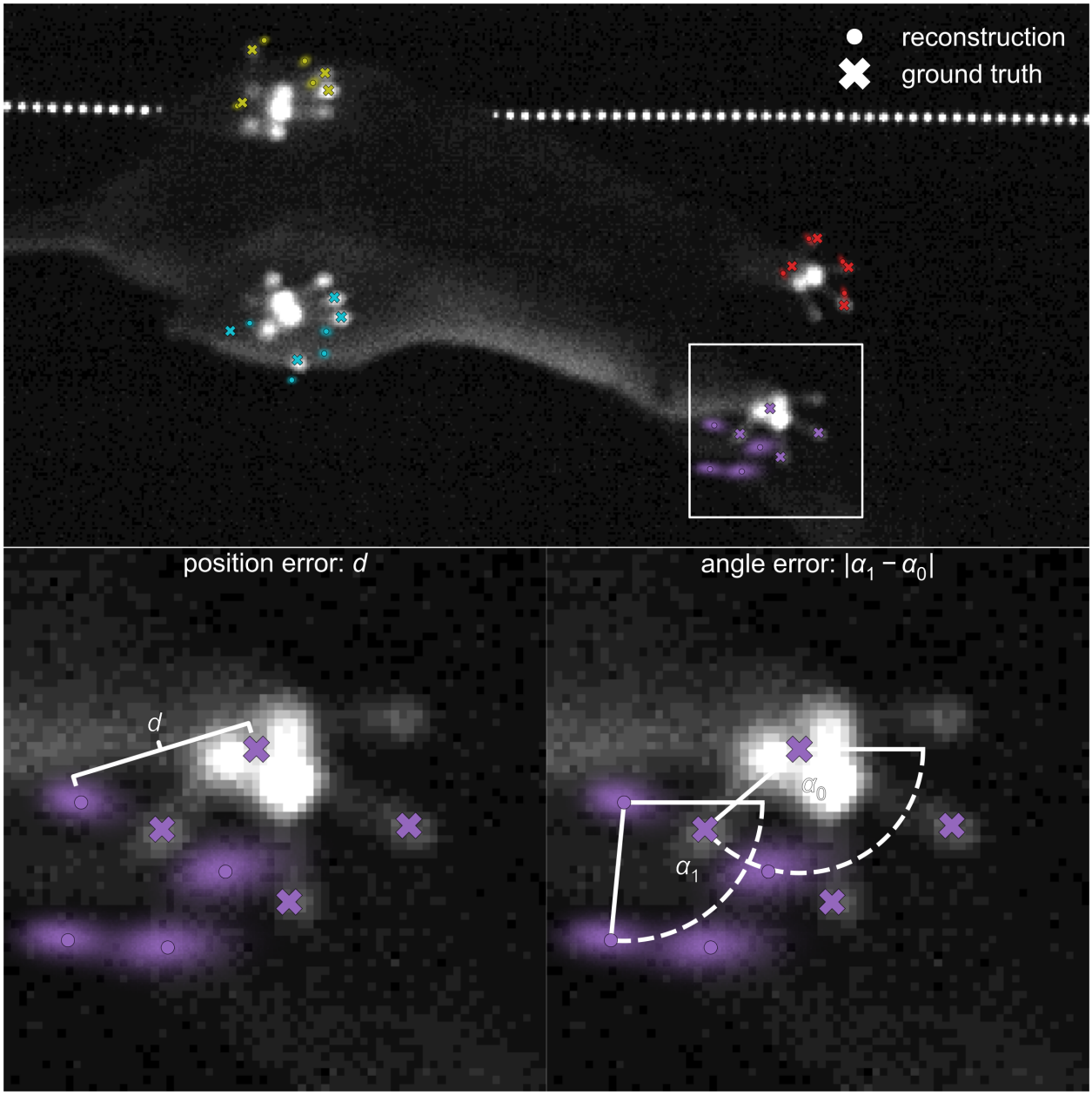

Figure 3.4: Single image recorded with an underneath camera, showing a freely-moving animal subject on a transparent FTIR plate (top). The reconstructed (colored circles) and ground truth (colored crosses) xy-positions of the center points, fingers and toes of the four different paws are shown (purple: left front paw, red: right front paw, cyan: left hind paw, yellow: right hind paw). Large point clouds in the vicinity of the reconstructed xy-positions indicate high reconstruction uncertainty. Enlarged views of the left front paw furthermore illustrate how the position and angle errors are computed (bottom).

namely the full, temporal, anatomical and naive model. In the full model anatomical and temporal constraints were enforced using the state space model in combination with the unscented RTS smoother and the EM algorithm. This was also the case for the temporal model but here joint angle limits for the limb joints were set to $[-180, 180]$, whenever the respective limit in the full model did not equal $[0, 0]$ already (Table 2.3). This enabled full 360 deg bone rotations at the respective limb joints, such that effectively no physiological joint angle limits were enforced in the temporal model. The pose-encoding parameters in the full and temporal model were initialized by reconstructing the skeletal pose of only the first time point of an analyzed behavioral sequence by minimizing the

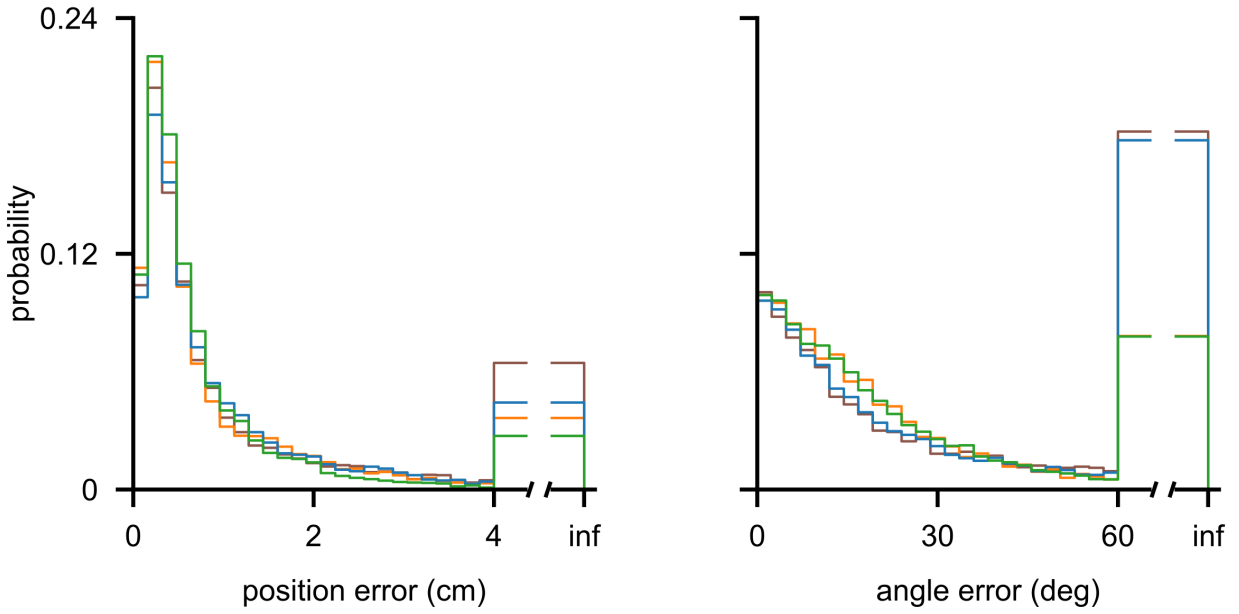

Figure 3.5: Probability histograms of the position (left) and angle (right) error obtained via four different reconstruction models, i.e. the full (green), temporal (blue), anatomical (orange) and naive (brown) model.

objective function given in Equation 2.17 via gradient descent optimization. Here, automatically detected instead of manually labeled two-dimensional surface marker locations were used in the objective function to calculate discrepancies between them and the corresponding reconstructed surface marker positions. This optimization scheme was also used for reconstructing poses via the anatomical and the naive model. As a consequence, no temporal constraints were enforced in the anatomical and the naive model, since the state space model and the unscented RTS smoother were not deployed here. To initialize the pose-encoding parameters in the anatomical or naive model for a given time point of an analyzed behavioral sequence, the restructured skeletal pose of the previous time point was used. Furthermore, to also remove the enforcement of anatomical constraints in the naive model, respective joint angle limits of limb joints were set to $[-180, 180]$, equivalently to the temporal model.

Comparing the position and angle errors of the four different reconstruction models with each other showed that the pose reconstruction accuracy varied depending on which reconstruction model was used (Figure 3.5). Particularly, the full model produced significantly smaller position errors compared to the other models (10410 position errors in total; p-values of one-sided Kolmogorov-Smirnov test: full vs. anatomical: $9.84\text{x}10^{-21}$; full vs. temporal: $4.38\text{x}10^{-35}$; full vs. naive: $9.03\text{x}10^{-37}$). However, angle errors were only significantly smaller when comparing the full to the temporal and naive model (7203 and 6969 angle errors in total for the full / anatomical and the temporal / naive model respectively; p-values of one-sided Kolmogorov-Smirnov test: full vs. temporal: $3.20\text{x}10^{-39}$; full vs. naive: $2.51\text{x}10^{-50}$). Additionally, the fraction of position errors exceeding 4 cm increased, when constraints were not enforced (fraction of position errors exceeding 4 cm: full: 2.72%; anatomical: 3.64%; temporal: 4.42%; naive: 6.44%). The same trend was observed for angle errors exceeding 60 deg (fraction of angle errors exceeding 60 deg: full: 7.78%; anatomical: 7.81%; temporal: 17.77%; naive: 18.22%).

While angle errors were significantly reduced by the anatomical constraints, enforcing temporal constraints limited abrupt pose changes over time (Figure 3.6). Particularly, this effect became evident when computing and comparing joint velocities and accelerations from skeletal poses reconstructed via the four different reconstruction models (Figure 3.7). In fact, joint velocities obtained

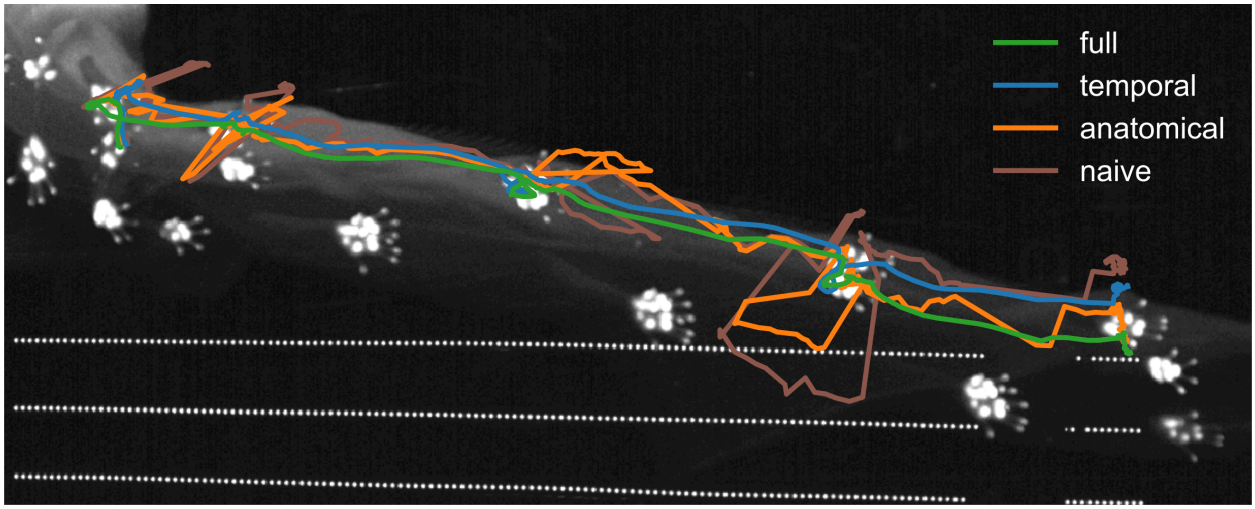

Figure 3.6: Maximum intensity projection of images recorded with an underneath camera, showing a 2.5 s long behavioral sequence of a freely-moving animal subject walking on a transparent FTIR plate. The colored trajectories show the reconstructed xy-positions of the right hind paw, given by the full (green), temporal (blue), anatomical (orange) and naive (brown) model. Note how the reconstruction quality occasionally decreases for the anatomical and the naive model, which both do not enforce temporal constraints (brown and orange trajectories).

via the full model were significantly smaller when compared to the other models (576288 joint velocities in total; p-value of one-sided Kolmogorov-Smirnov test: full vs. anatomical: numerically 0; full vs. temporal: numerically 0; full vs. naive: numerically 0). Furthermore, the same was true for joint accelerations (576288 joint accelerations in total; p-value of one-sided Kolmogorov-Smirnov test: full vs. anatomical: numerically 0; full vs. temporal: $3.71 \text{x} 10^{-90}$; full vs. naive: numerically 0). Enforcing temporal constraints also lowered the percentage of velocities exceeding 0.08 cm/ms (full: 3.29%; anatomical: 13.49%; temporal: 3.28%; naive: 13.85%). Additionally, this was also the case for accelerations exceeding 0.02 cm/ms^2 (full: 0.22%; anatomical: 23.43%; temporal: 0.25%; naive: 24.55%). Here, velocity and acceleration values were computed via central finite differences (order of accuracy: 8) [134] based on the reconstructed three-dimensional positions of surface markers, which corresponded to the paw center points, fingers and toes.

To also assess the effect of missing measurements (Section 2.3.2), position errors were computed for only those surface markers, whose two-dimensional locations were not successfully detected by the trained DeepLabCut networks, e.g. due to occlusions (Figure 3.8). Compared to all other models the full model produced significantly lower errors (2797 position errors in total; p-values of one-sided Kolmogorov-Smirnov test: full vs. anatomical: $9.67 \text{x} 10^{-23}$; full vs. temporal: $2.83 \text{x} 10^{-22}$; full vs. naive: $3.91 \text{x} 10^{-47}$). Additionally, the full model also produced the smallest number of errors above 4 cm (full: 9.36%; anatomical: 11.61%; temporal: 13.72%; naive: 19.12%). Besides, averaged position errors increased the longer a surface marker remained undetected for the full and the naive model (linear regression full: slope: 1.49 cm/s, intercept: 1.13 cm; linear regression naive: slope: 2.77 cm/s, intercept: 1.39 cm). However, averaged position errors obtained via the full model were significantly lower when compared to errors obtained via the naive model (p-value of one-sided Mann-Whitney rank test: full vs. naive: $3.91 \text{x} 10^{-47}$).

To accurately compare the averaged position errors of occluded surface markers for the full and the naive model, the error calculation was adjusted depending on which model was used. Time spans until the next successful detection were treated equally to time spans since the last successful detection, when calculating the averaged position errors of the full model, since here the used unscented RTS smoother incorporates information from past and future time points. As a consequence, the direction of time becomes irrelevant, when computing the time span for which

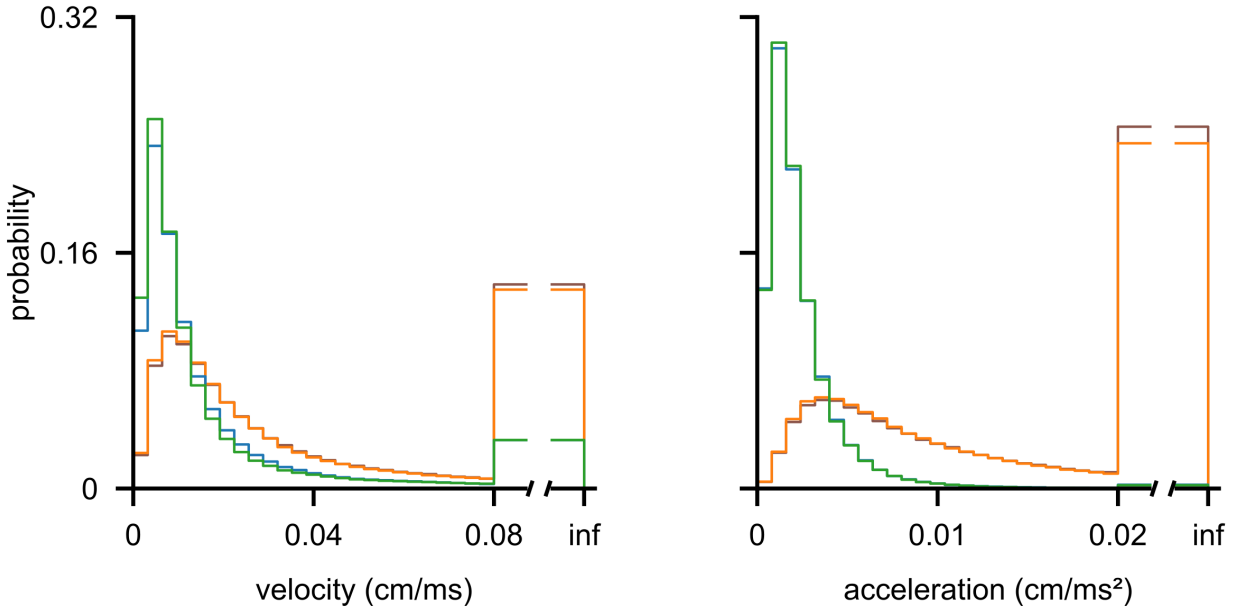

Figure 3.7: Histograms of the joint velocities (left) and accelerations (right) obtained from four different reconstruction models, i.e. the full (green), temporal (blue), anatomical (orange) and naive (brown) model. Note how velocities and accelerations are substantially higher, when no temporal constraints are enforced, i.e. when the anatomical or naive model is used for reconstructing skeletal poses.

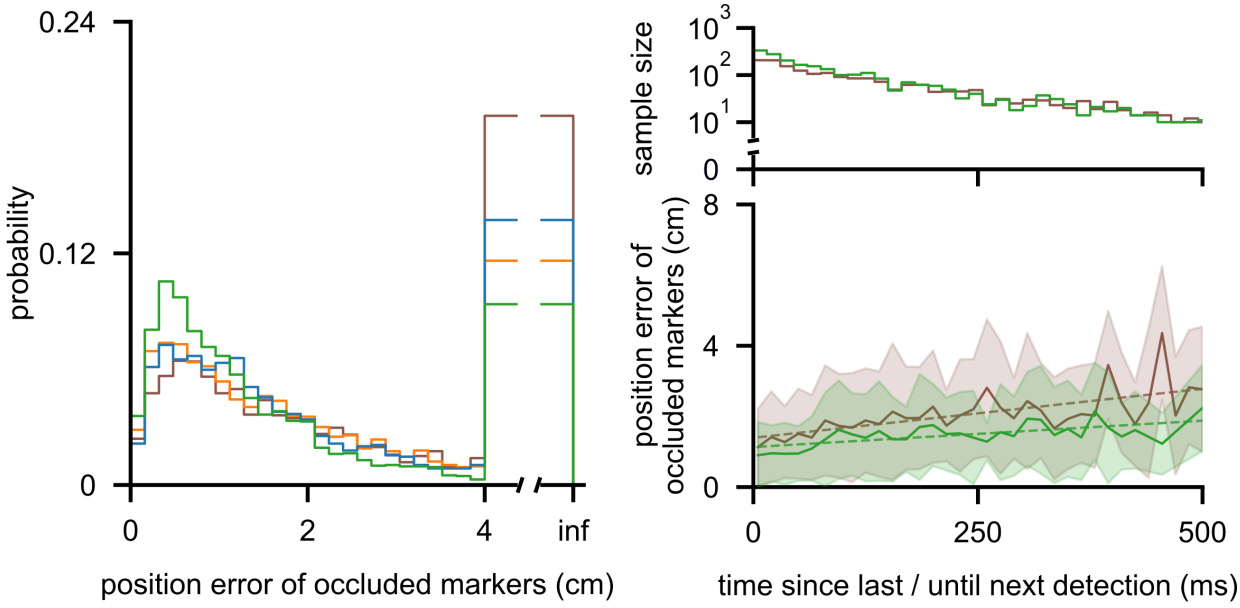

Figure 3.8: Probability histogram of the position error obtained via four different reconstruction models, i.e. the full (green), temporal (blue), anatomical (orange) and naive (brown) model, where only undetected surface markers are considered (left). Additionally, the averaged position errors of undetected surface markers (bottom right) and corresponding binned sample sizes (top right) as a function of time since the last or until the next successful surface marker detection are shown. The distinction between ”since the last” and ”until the next” successful detection is necessary due to the different nature of the full and the naive reconstitution model, i.e. the full model has access to detections in the past and future, whereas the naive model only has access to detections in the past. The shaded areas represent the computed standard deviations of the averaged position errors (bottom right).

a given surface marker remained undetected. Thus, in the full model a time span, which indicated how much time has passed since the last successful detection of a given surface marker, was defined as the smallest element of the set containing both, the time span until the next successful detection and the time span since the last successful detection. However, in the naive model only time spans since the last successful detection were considered for the respective computations, since the unscented RTS smoother was not used here, such that the naive model did not have the capacity to process pose information of future time points.

Altogether, this evaluation demonstrated the beneficial effects of simultaneously enforcing anatomical and temporal constraints during pose estimation. Particularly, considering both constraint types led to an overall increased pose reconstruction accuracy.

3.3 Quantifying periodic gait cycles

To assess to which extent the proposed pose reconstruction framework allows for extracting periodic gait cycles and how the enforced anatomical and temporal constraints affect the corresponding results, skeletal poses were reconstructed based on recorded behavioral sequences of freely-moving animals. Particularly, animal #1 and animal #2 (Section 3.1) were recorded via four different overhead cameras while they were allowed to move freely in an open arena of size 80x105 cm^2 with 50 cm high walls. All overhead cameras were calibrated prior to the experiments, recorded the images synchronously, had a resolution of 1280x1024 px^2 and were operated at 100 Hz with an acquisition time of 2.5 ms. The therefore generated data set consisted of 27 sequences with a total of 14650 frames in each of the four cameras and a total duration of 146.5 s.

To allow for the automated detection of two-dimensional surface marker locations in the images, which were recorded via the four overhead cameras, an individual DeepLabCut network was trained for each animal. Particularly, 2404 different images were used for training a respective DeepLabCut network for each animal subject. These training images were not part of the data set, which was analyzed to assess if periodic gait cycles could be extracted based on reconstructed poses. The automatically detected surface marker locations were then used to reconstruct skeletal poses of freely-moving animals via the proposed pose reconstruction framework.

While the animals were moving freely in the arena, sequences of gait were observed frequently. During these gait sequences animals crossed various distances on different paths in the arena (Figure 3.9). To extract cyclic gait patterns from the recorded gait sequences, skeletal poses were reconstructed and analyzed with respect to skeletal kinematics by computing four different kinematic metrics, namely the x-positions and -velocities of different joints as well as their angles and angular velocities (Figure 3.10).

To calculate the kinematic metrics, all reconstructed skeletal poses were aligned by applying a respective coordinate transformation, which translated and rotated the three-dimensional joint

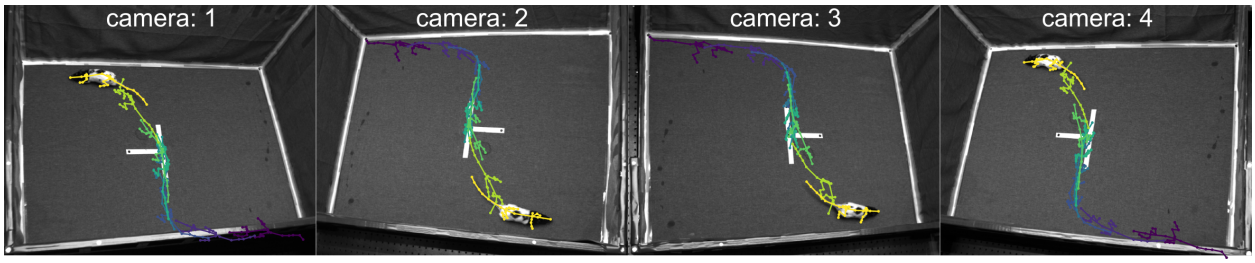

Figure 3.9: Images of a freely-moving animal recorded via four different overhead cameras. Reconstructed skeletal poses are shown for different time points of an analyzed gait sequence (colored lines). Time differences between the shown skeletal poses are equal to 1 s.

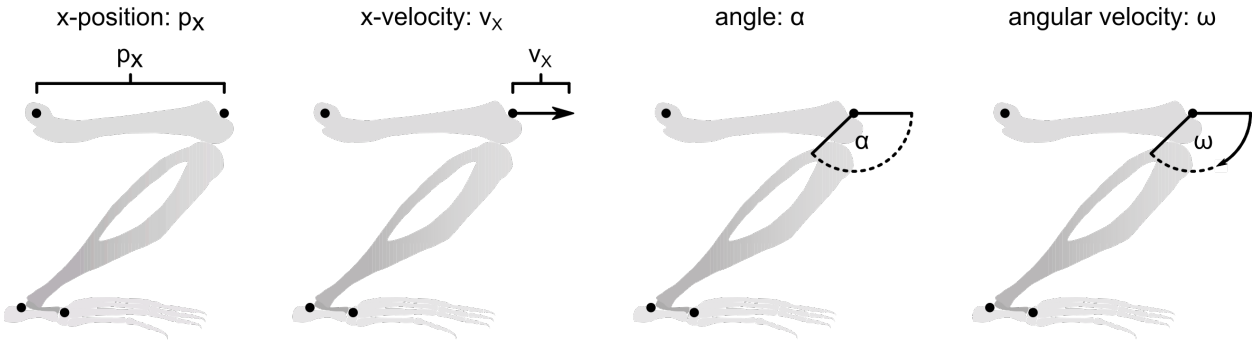

Figure 3.10: Schematic illustrations of kinematic metrics for a single joint, i.e. the x-position (left), x-velocity (center left), angle (center right), and angular velocity (right) of the knee joint. For the shown example of the knee joint, the x-position p_x denotes the distance from the pelvis, i.e. spine joint #2 (Figure 2.8), to the knee joint, whereas the angle α denotes the angle between the walking direction and the tibia, i.e. the bone connecting knee and ankle joints. The x-velocity v_x and angular velocity ω are computed as the first temporal derivatives of the x-position p_x and the angle α.

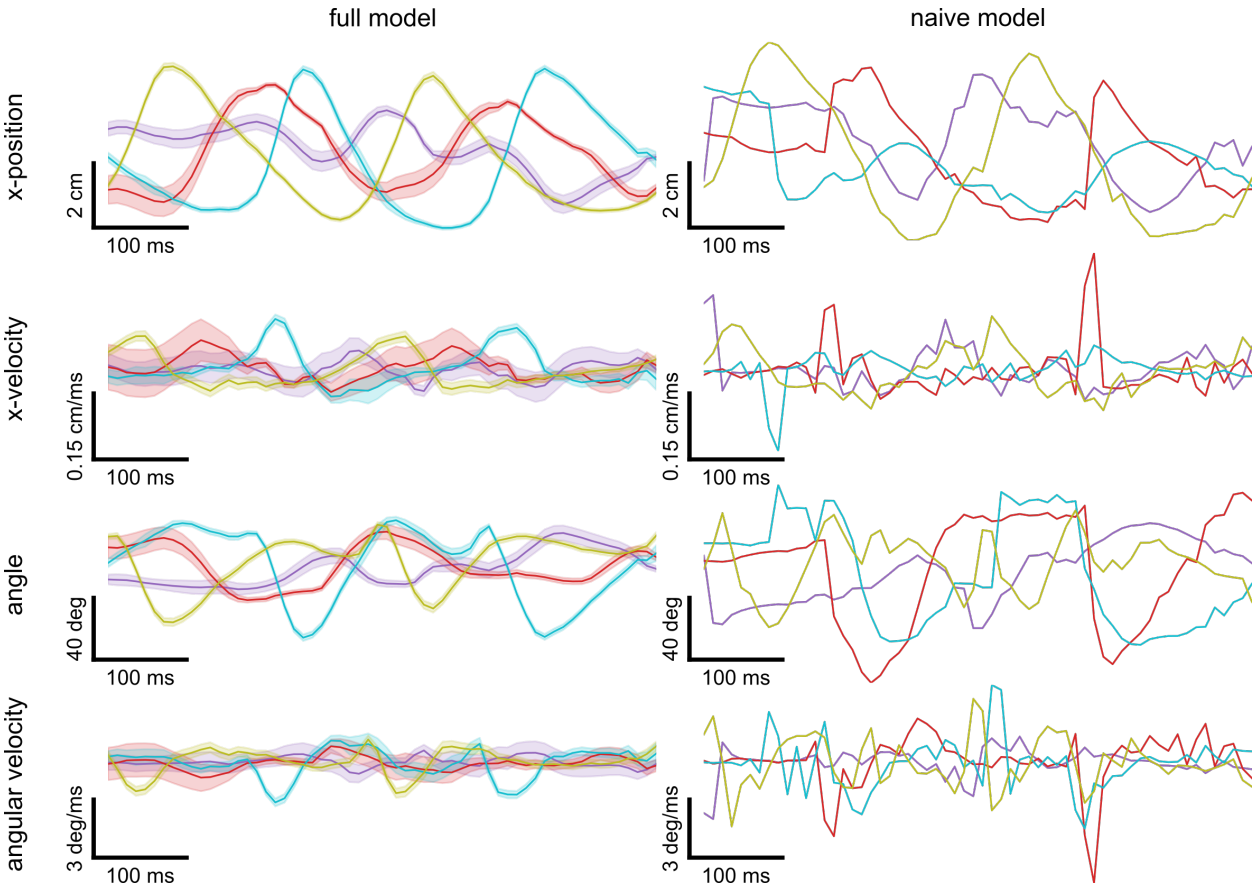

Figure 3.11: Exemplary traces of the x-position (top row), x-velocity (center top row), angle (bottom top row) and angular velocity (bottom row) of the left wrist (purple), right wrist (red), left ankle (cyan) and right ankle (yellow) joint as a function of time. The four kinematic metrics were computed based on reconstructed skeletal poses obtained via the full (left column) and naive (right column) reconstruction model for a single gait sequence. To obtain a concise overview, all traces were centered by subtracting their mean value. Shaded areas represent standard deviations.

positions. This coordinate transformation changed the origin of each reconstructed skeletal pose to spine joint #2, which connects the hind limb bones to the spine, and modified the x-direction, such that it pointed from the new origin joint, i.e. spine joint #2, to the xy-position of spine joint #4, which connects the front limb bones to the spine (Figure 2.8). As a consequence, the new x-direction always coincided with the walking direction of the respective animal subject. Thus, the x-position of a joint denoted the distance along the new x-direction from the respective joint to spine joint #2 and the joint angle measured the angle between the new x-direction and the bone, whose end-joint was identical to the respective joint. Based on these two quantities, the corresponding x-velocities and angular velocities of a joint were calculated via central finite differences (order of accuracy: 8) [134].

To compare the full to the naive reconstruction model (Section 3.2), skeletal poses were reconstructed via both models based on the recorded video data of gait sequences. Subsequently, the reconstructed poses were used to calculate the kinematic metrics. When analyzing the temporal progression of individual traces of these kinematic metrics, periodic gait cycles could be identified. Respective gait cycles formed self-similar patterns, whereas gait periodicity was more evident when skeletal poses were reconstructed via the full model instead of the naive model (Figure 3.11). In fact, individual traces generated via the naive model were dominated by noise, such that the periodic nature of gait was overall less apparent, when the naive model was used. However, when skeletal poses were reconstructed via the full model the self-similarity of gait cycles was evident, for instance, when computing auto-correlations of individual x-position traces as well as their associated frequency spectra (Figure 3.12; fitting auto-correlations via a damped sinusoid: frequency: 3.14 Hz, decay rate: 2.49 Hz, R^2-value: 0.90; frequency spectra of auto-correlations: max. peak at 3.33 Hz, sampling rate: 0.83 Hz).

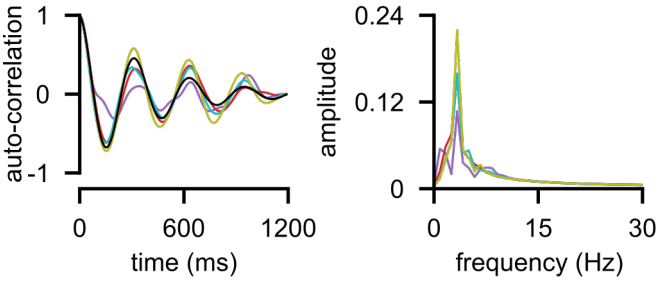

Figure 3.12: Auto-correlations of the x-position as a function of time (top) and corresponding Fourier-transformed data (bottom). Shortened time sections of the associated x-position traces, which were generated via the full model, are shown in Figure 3.11 using the same color-coding. Note how a fitted damped sinusoid (black) matches the auto-correlations and all Fourier-transformed auto-correlations have their maximum peak at the same frequency.

Furthermore, averaging the traces of the four different kinematic metrics across the entire population using all 27 gait sequences yielded population-averaged traces, which highlighted the beneficial effects of enforcing anatomical and temporal constraints via the full model (Figure 3.13, 3.14, A.1 and A.2). To obtain these population-averaged traces, swing phase mid-points of individual limbs were localized by identifying maximum peaks in individual x-velocity traces of different joints with values above 25 cm/s. Subsequently, population-averaged traces were calculated using 400 ms long sub-sections of the individual x-velocity traces, which all contained the mid-point of a single swing phase. Prior to averaging, the sub-sections were aligned, such that the swing phase mid-point of each sub-section was always located at the center of the resulting population-averaged trace, i.e. 200 ms before the end and after the beginning of each population-averaged trace.

Population-averaged traces of the four different limbs obtained via the full model were significantly less variable than those obtained via the naive model, i.e. standard deviations related to the full model were smaller than those related to the naive model for all time points of the population-averaged traces (Figure 3.13 and 3.14; p-value of one-sided Mann-Whitney rank test: x-position: 1.40×10^{-49}; x-velocity: 2.28×10^{-55}; angle: 1.42×10^{-55}; angular velocity: 1.44×10^{-55}). Additionally, the periodicity of gait cycles was apparent in the population-averaged traces in the form of equidis-

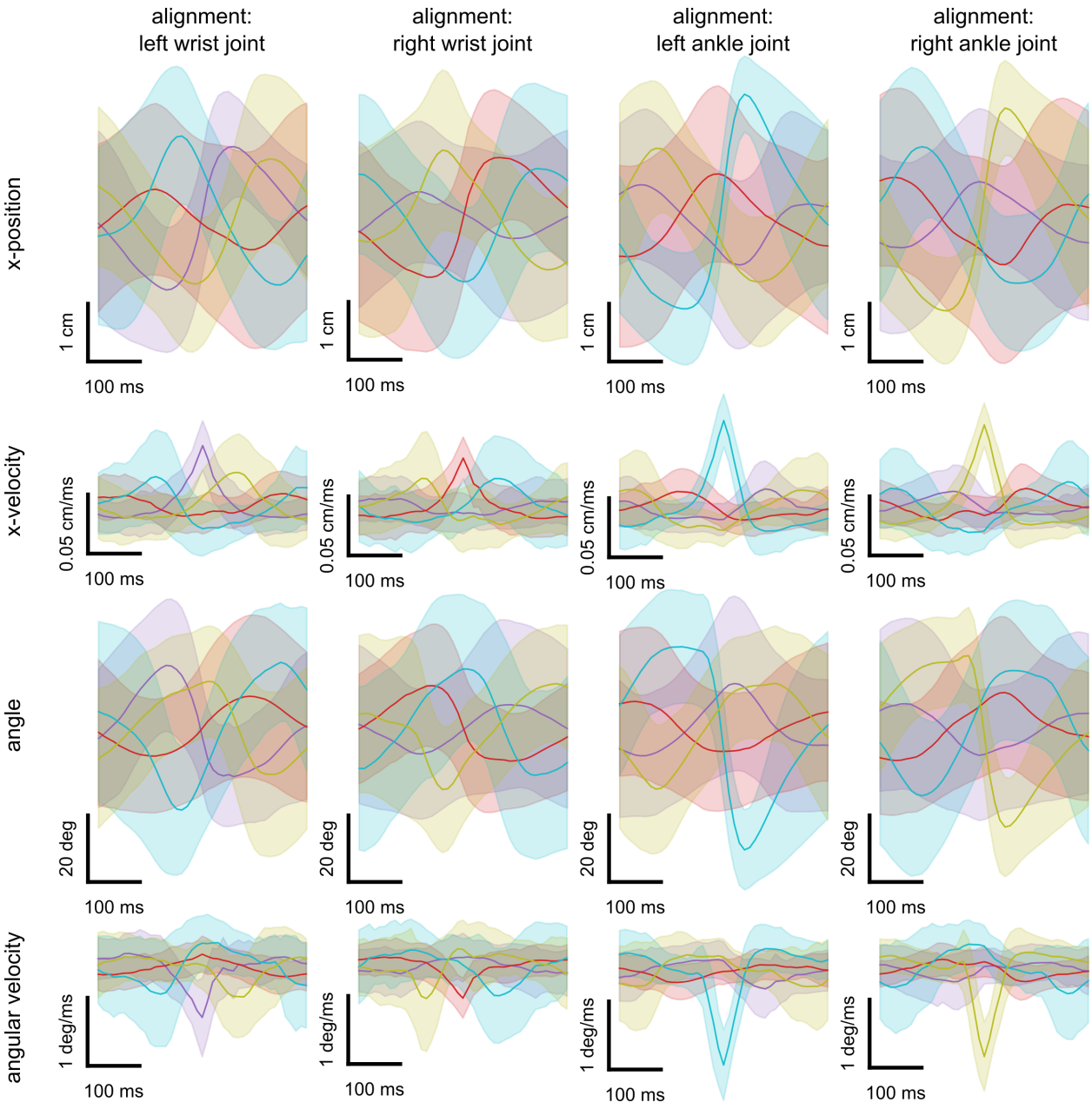

Figure 3.13: Population-averaged traces of the x-position (top row), x-velocity (center top row), angle (center bottom row) and angular velocity (bottom) as a function of time for the left wrist (purple), right wrist (red), left ankle (cyan) and right ankle (yellow) joint. To compute the population-averaged traces, individual traces were first obtained based on skeletal poses, which were reconstructed via the full reconstruction model. Then these individual traces were aligned to x-velocity peaks above 25 cm/s of the left wrist (left column), right wrist (center left column), left ankle (center right column) and right ankle joint (right column) before averaging across the entire population of individual traces. To obtain a concise overview, all population-averaged traces were centered by subtracting their mean value. Shaded areas represent standard deviations.

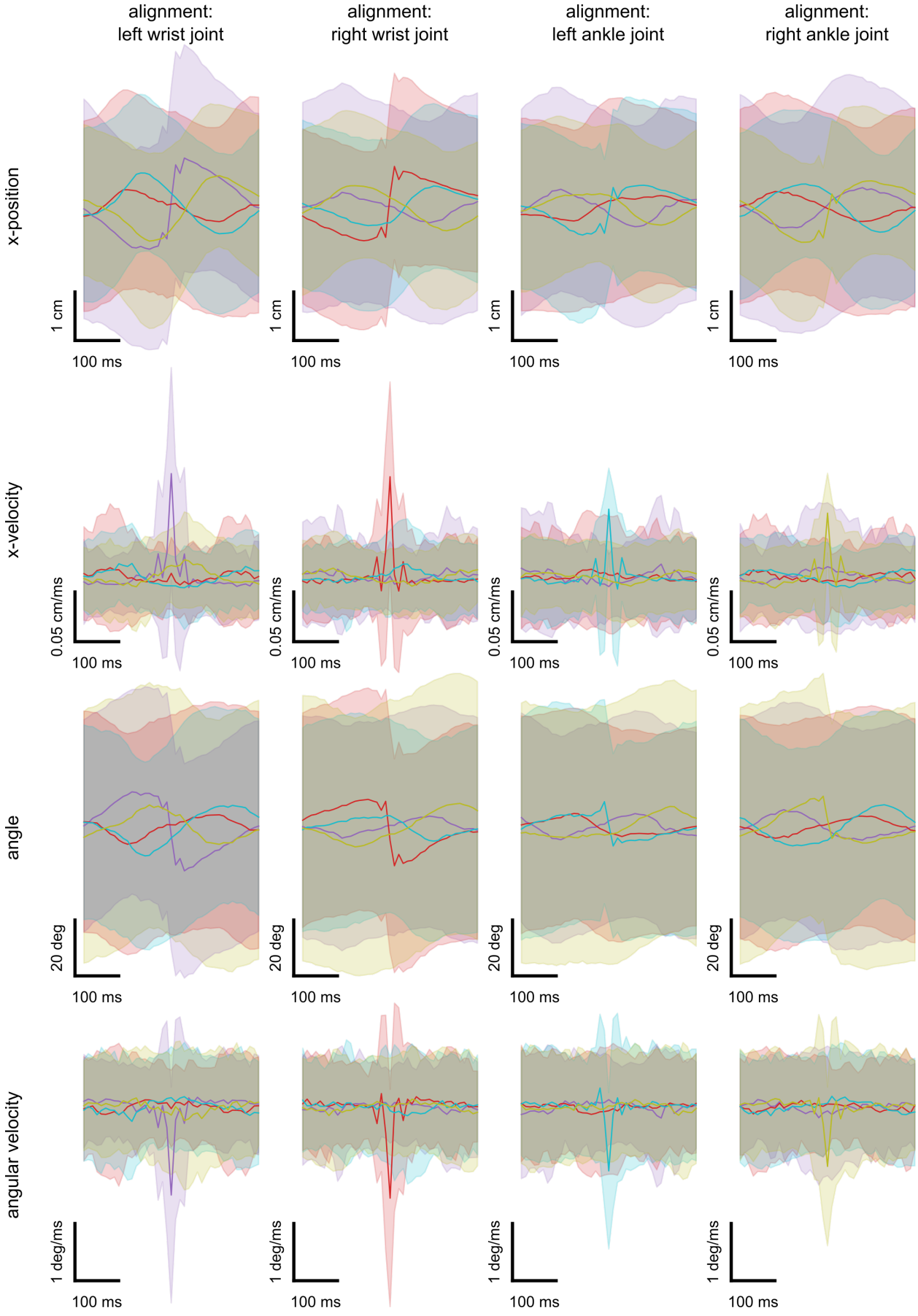

Figure 3.14: Same as Figure 3.13, except that traces were obtained via the naive reconstruction model.

tant maximum and minimum peaks, which corresponded to swing phase mid-points of individual limbs. Particularly, time differences between these peaks were less variable, when comparing population-averaged traces related to the full model with those related to the naive model (16 peaks resulting in 12 time differences in total; sampling rate: 10 ms; full: x-position [min. peaks]: 75.00 +/- 29.01 ms, x-velocity [max. peaks]: 78.33 +/- 10.67 ms, angle [max. peaks]: 78.33 +/- 23.74 ms, angular velocity [min. peaks]: 75.00 +/- 10.40 ms [avg. +/- s.d.]; naive: x-position [min. peaks]: 64.16 +/- 56.78 ms, x-velocity [max. peaks]: 80.83 +/- 54.99 ms, angle [max. peaks]: 74.16 +/- 33.53 ms, angular velocity [min. peaks]: 53.33 +/- 47.78 ms [avg. +/- s.d.]).

In contrast, high noise levels caused the periodicity of gait cycles to vanish in its entirety, when population-averaged traces were computed based on triangulated three-dimensional surface marker locations, since no underlying skeleton model was used in this case (Figure A.3 and A.4). Here, the three-dimensional surface marker locations were reconstructed via triangulation by directly using the corresponding two-dimensional surface marker locations, which were given by the trained DeepLabCut networks. Particularly, only the two most likely two-dimensional surface marker locations were taken into account for triangulation, i.e. the image locations with the highest probability values in the score maps of two different cameras (Section 2.3.2).

Altogether, this evaluation demonstrated that reconstructing skeletal poses via the full reconstruction model allows for accurately extracting and quantifying periodic gait cycles. However, when skeletal poses were reconstructed via the naive reconstruction model, extracting periodic gait cycles was less feasible. Furthermore, periodic gait cycles could not be observed based on trajectories of merely triangulated three-dimensional surface marker positions, whose two-dimensional counterparts were automatically detected via trained DeepLabCut networks.

3.4 Quantifying gap-crossing behaviors

To evaluate if the proposed pose reconstruction framework is suited for quantifying behaviors other than gait, skeletal poses were reconstructed based on recorded behavioral sequences of animals, which were subjected to a gap-crossing task. Particularly, animal #1 and animal #2 (Section 3.1) were recorded via four different overhead cameras, while they were crossing gaps of variable lengths. The respective gap-crossing track consisted of two 50x20 cm^2 platforms, mounted 120 cm off the ground on a slide mechanism to allow adjusting the distance between the platforms in the range of 10 to 30 cm. All overhead cameras were calibrated prior to the experiments, recorded the images synchronously, had a resolution of 1280x1024 px^2 and were operated at 200 Hz with an acquisition time of 2.5 ms. The therefore generated data set consisted of 44 sequences with a total of 8800 frames in each of the four cameras and a total duration of 44 s, such that each individual sequence had a length of 1 s.

To allow for automatically detecting two-dimensional surface marker locations in the recorded video data an individual DeepLabCut network was trained for each animal. In contrast to the previously described analyzes of animal behavior (Section 3.2 and 3.3), quantifying gap-crossing behavior required the training data set of each DeepLabCut network (Section 2.3.2) to be a subset of the finally analyzed data set, due to the limited number of gap-crossing events and recorded images. Particularly, 20% of the recorded images, which belonged to the finally analyzed data set, were used for training, i.e. every 5th image of each gap-crossing sequence. Thus, for each animal 3608 different images were used for training the DeepLabCut networks. Once the DeepLabCut networks were trained, they were used to automatically detect two-dimensional surface marker locations in all recorded images. These automatically detect surface marker locations were then used to reconstruct skeletal poses during gap-crossing.

Subjecting animals to gap-crossing tasks with varying gap lengths forced them to coordinate their body during the entire time of the jump and to re-estimate the jumped distance in each trial

in order to prevent falling of the track (Figure 3.15). Reconstructing skeletal poses during gap-crossing allowed for relating different kinematic quantities to the placement of individual paws, e.g. by computing the angle of spine joint #3 (Figure 2.8) at the onset of a jump in relation to hind paw positions upon landing (Figure 3.16). The respective joint angles were calculated as the angle between two connected bones and temporal kinematic quantities, i.e. spatial and angular velocity values of joints, were computed via central finite differences (order of accuracy: 8) [134]. To obtain kinematic quantities, reconstructed skeletal poses were aligned equivalently to the analyses of gait data (Section 3.3), i.e. a coordinate transformation was applied to the three-dimensional joint positions, such that the origin of each skeletal pose was identical to the location of spine joint #2 (Figure 2.8) and the x-direction pointed from this origin joint to the xy-position of spine joint #4 (Figure 2.8).

Similar to periodic gait cycles, gap-crossing behaviors appeared to follow a stereotypical pattern, which allowed for identifying specific behavioral decision points based on the reconstructed skeletal poses. These decision points were the start-, mid- and end-point of a jump. To obtain these points in each gap-crossing sequence, computed joint angles for all spine and hind limb joints were averaged, yielding traces of averaged joint angles over time for each jump (Figure 3.17). In the resulting averaged joint angle traces distinct minimum and maximum peaks were always present in the following order: local minimum, local maximum, global minimum, local maximum, local minimum. Identifying these extrema allowed for extracting the start- and end-point of a jump, since they coincided with the first and last local minimum respectively. Additionally, the mid-point of each jump was also extracted, since it coincided with the global minimum of the respective trace. The similar temporal progression of the averaged joint angle traces pointed towards a consistent gap-crossing behavior, which could be illustrated by computing a single globally-averaged joint angle trace as well as characteristic skeletal poses at the start-, mid- and end-point of a stereotypical jump (Figure 3.17).

Besides, high cross-correlations of kinematic quantities, i.e. spatial and angular limb velocities, indicated that joint motions were interdependent at the start-points of the jumps (Figure 3.18). For instance, significant correlations between the spatial velocities of the right wrist and elbow joints and the right and left knee joints were found in reconstructed skeletal poses (right wrist vs. right elbow: Spearman correlation coefficient: 0.95, two-tailed p-value testing for non-correlation: 5.40×10^{-24}; right knee vs. left knee: Spearman correlation coefficient: 0.93, two-tailed p-value testing for non-correlation: 6.79×10^{-20}). Here, spatial velocities were calculated as the absolute velocity values of the joints in three-dimensional space, i.e. the lengths of their velocity vectors.

Additionally, reconstructed skeletal poses allowed for weak predictions of behavioral outcomes, since kinematic quantities at time points prior to the end-points of the jumps were correlated with the jumped distances (Figure 3.19). Here, jumped distances were computed as the absolute xy-difference of hind paw positions at the start- and end-point of each jump and paw positions were defined as the average of the ankle, hind paw and toe joint positions. For instance, angular velocities of spine joint #3 (Figure 2.8) and z-velocities of spine joint #4 (Figure 2.8) were significantly correlated with the jumped distances 205 ms and 175 ms before the end-points of the jumps (Spearman correlation coefficients: -0.73 and 0.81, two-tailed p-values testing for non-correlation: 1.13×10^{-8} and 1.12×10^{-11}).

Altogether, this evaluation demonstrated that reconstructing skeletal poses enables quantitative analyses of skeletal kinematics during complex gap-crossing behaviors, where animals jump to cross gaps of various lengths. These analyses showed that skeletal movement patterns during gap-crossing behaviors were conserved across different trials and comprised interdependent joint motions at the onsets of the jumps. Furthermore, kinematic quantities of the skeleton were correlated with jumped distances, such that the kinematic quantities were weakly predictive of the distance the animals crossed via jumping.

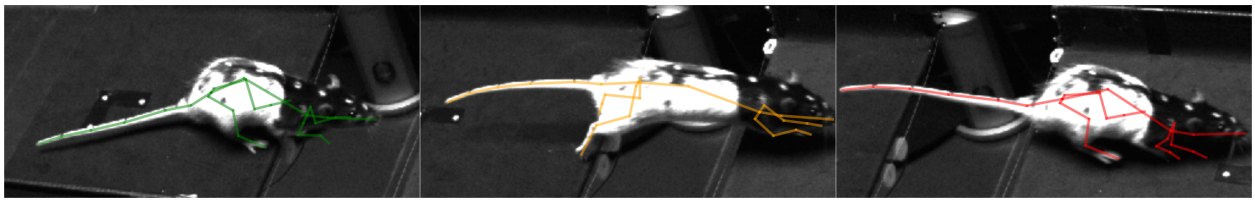

Figure 3.15: Images of an animal performing a gap-crossing task at the start (left), middle (center) and end (right) of a jump. Additionally, the reconstructed skeletal poses corresponding to the three different time points are shown (green, orange and red lines).

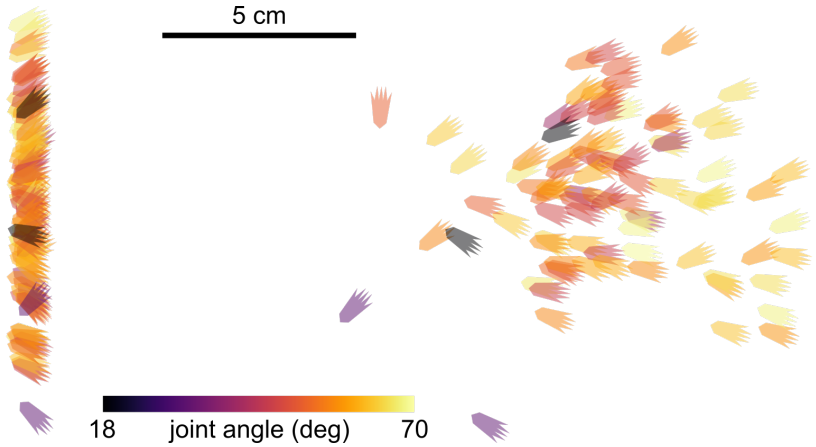

Figure 3.16: Reconstructed xy-positions of the hind paws at the start and end of jumps obtained from animals, which were subjected to a gap-crossing task. The displayed paw positions are color-coded by the joint angle of spine joint #3 (Figure 2.8). Paw positions at the start of the jumps (paws on the left) were aligned, such that their x-positions matched.

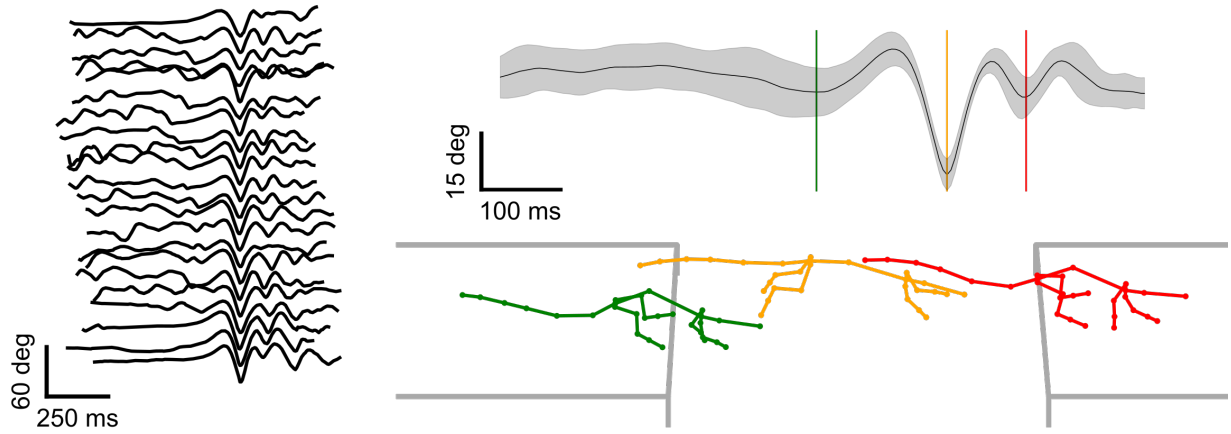

Figure 3.17: Averaged joint-angle traces containing all spine and hind limb joint angles for 22 out of 44 gap-crossing trials (left). Aligning all 44 traces to the mind-point of each jump (global minimum of each trace) and averaging again across all trials yielded a globally-averaged joint angle trace (top right), which illustrates stereotypical jumping behavior during gap-crossing. The respective start-, mid- and end-point of the resulting stereotypical jump are highlighted (vertical green, orange and red lines respectively). Similarly, averaging reconstructed skeletal poses of all 44 gap-crossing trials at the start- (green), mid- (orange) and end-point (red) of the individual jumps yielded characteristic skeletal poses for these time points (bottom right).

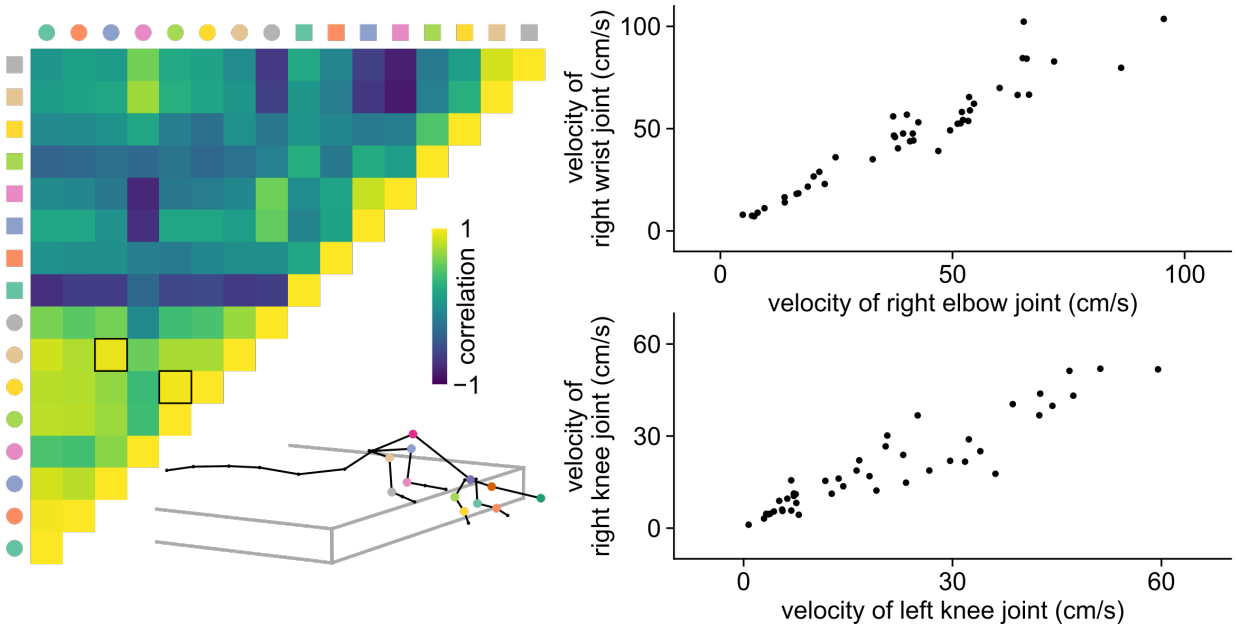

Figure 3.18: Cross-correlation matrix of spatial and angular velocities at the start-points of the jumps for different limb joints (left). Different marker shapes indicate whether rows and columns represent spatial or angular velocities (circles: spatial velocities, squares: angular velocities). An additional illustration of the skeletal pose of an animal at the start-point of a jump indicates which marker color corresponds to which joint. Examples for two high correlation values are highlighted with black rectangles in the cross-correlation matrix and displayed via scatter plots (right).

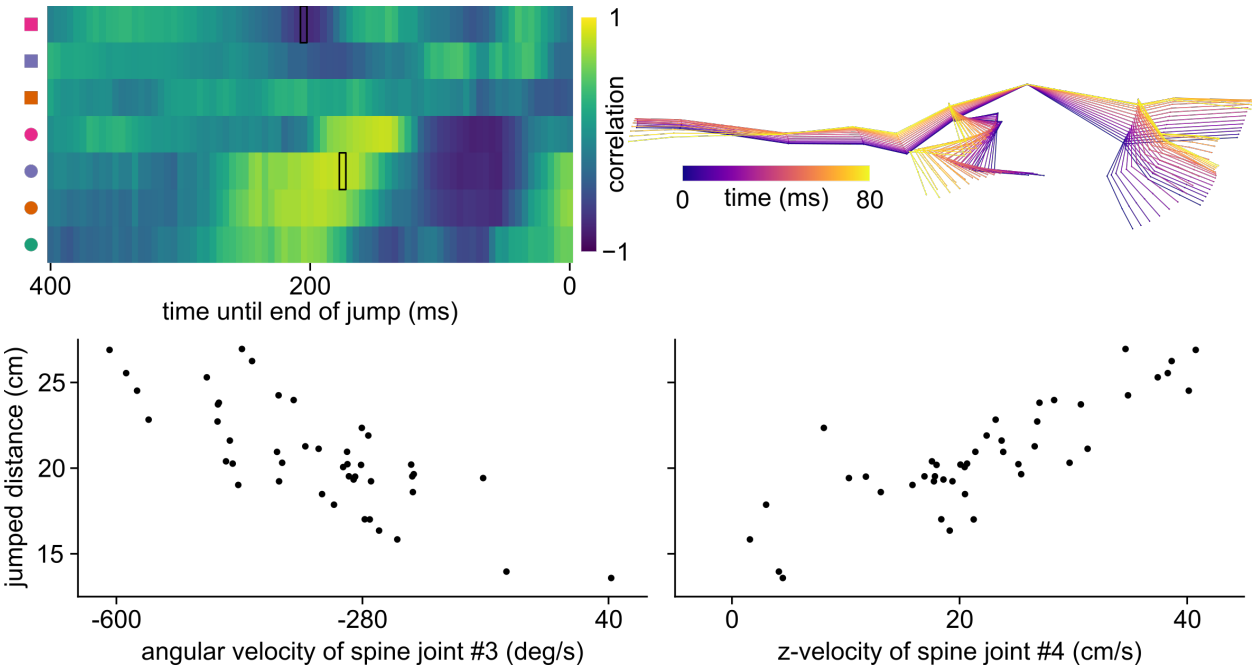

Figure 3.19: Cross-correlation matrix showing how the z- and angular velocities of head and spine joints are correlated with the jumped distances for time points up to 400 ms before the end-points of the jumps (top left). Conventions of marker colors and shapes are the same as in Figure 3.18. Examples for two high correlation values are highlighted with black rectangles in the cross-correlation matrix and displayed via scatter plots (bottom). Additionally, aligned and subsequently overlaid reconstructed skeletal poses 240 ms to 160 ms before the end of a single jump are shown, such that the positions for spine joint #3 (Figure 2.8) are identical for all poses (top right). The displayed skeletal poses were taken from a jump trial represented by the top left data point in the scatter plot at the lower left.

Chapter 4

Discussion

4.1 Conclusive summary

In the previous chapters of this thesis a framework for reconstructing skeletal poses of freely-moving animals at the resolution of single joints is presented. The lengths and rotation ranges of bones are constrained within an underlying skeleton model based on realistic anatomical principles [110, 111], such that only physiologically-feasible poses are reconstructed (Section 2.2). Additionally, skeletal poses are estimated in a probabilistic manner by deploying a state space model, which furthermore ensures that temporal constraints are accounted for, such that reconstructed poses are consistent in time (Section 2.3). The pose-encoding variables of the state space model are inferred via an unscented RTS smoother [128, 129], which processes pose information from past and future time points of a behavioral sequence, while an EM algorithm [125, 130] is used to learn the required probabilistic hyper-parameters of the unscented RTS smoother (Section 2.4).

The proposed pose reconstruction framework relies on video data of freely-moving animals, which needs to be recorded via multiple calibrated cameras. The respective calibration of the cameras is performed automatically without the need for generating any manual annotations (Section 2.1). Subsequently, the positions of surface markers located on the fur of the freely-moving animals are determined in the recorded video data, for which a published CNN architecture for detecting anatomical surface features in two-dimensional images, i.e. DeepLabCut [76], is used. Fusing these automatically generated detections with the anatomical and temporal constraints enforced by the proposed pose estimation framework enables reconstructing skeletal poses and quantifying skeletal kinematics of freely-moving animals of various sizes.

The results presented in this thesis show that the proposed pose reconstruction framework allows for learning bone lengths as well as three-dimensional joint and surface marker locations from recorded two-dimensional video data. Particularly, learned skeleton anatomies are validated by comparing them with ground truth data obtained via MRI scans (Section 3.1). Besides, further analyses show that simultaneously enforcing the implemented anatomical and temporal constraints is advantageous for the pose reconstruction accuracy of the proposed pose reconstruction framework and enables compensating for occasional occlusions of body parts and noisy detections of surface markers (Section 3.2). Additionally, the scientific potential of reconstructing skeletal poses within the scope of behavior quantification is demonstrated by analyzing periodic gait sequences (Section 3.3) as well as gap-crossing events (Section 3.4), both of which involve complex coordinated limb placements. Particularly, the proposed pose reconstruction framework offers the opportunity to relate these behaviors with the underlying skeleton. Respective analyses show that reconstructed skeletal poses follow characteristic movement patterns during gait and gap-crossing and that they comprise an interplay of interdependent skeletal kinematics, which are furthermore correlated to future behavioral outcomes, i.e. jumped distances during gap-crossing.

In contrast to related studies in the field of animal pose estimation, which solely rely on deep neural networks to approximate a black-box function mapping recorded images to anatomical feature locations [8,69,76,77], the work presented in this thesis takes a different direction to shed light into how the underlying skeleton gives rise to the movement dynamics of visible markers located at an animal's body surface. Using a realistic skeleton model to incorporate mechanistic knowledge about the physical world into the proposed pose reconstruction framework thereby allows for estimating interpretable bone rotations, while anatomical and temporal constraints are accounted for. Thus, the work presented in this thesis addresses the challenges of directly estimating hidden skeletal kinematics, which ultimately govern how an animal's body surface appears, instead of merely aiming at reconstructing the visible body surface itself. Consequently, the proposed pose estimation framework has the capacity to not only enhance the detail at which animal behavior can be studied, but also provides an opportunity for objectively quantifying underlying bone and joint movements in freely-behaving animals.

4.2 Limitations

While the results presented in this thesis indicate that the proposed pose estimation framework offers the unique potential to unravel skeletal kinematics of freely-moving animals, there exist limitations with respect to the framework itself as well as to how the framework was evaluated. Particularly, there exist limitations regarding the framework's overall performance and reconstruction capabilities, i.e. technical limitations, as well as the analyses, which were performed within the scope of this thesis, i.e. analytical limitations.

4.2.1 Technical limitations

A technical limitation of the proposed pose reconstruction framework is given by its dependence on automatically generated detections of surface marker positions, which are obtained via a trained CNN (Section 2.3.2). The respective detection accuracy is required to be high enough, such that the majority of surface marker positions fed into the framework is reasonably accurate and therefore reliable to a certain degree. For a very high number of incorrect detections the framework is expected to generate inaccurate pose reconstruction results. In fact, the framework does not provide an unlimited capability for compensating for incorrectly detected surface marker positions. This is particularly true when the overall fraction of erroneous surface marker detections becomes too large. Similarly, if the probabilistic certainties of the trained CNN with respect to the surface marker positions, i.e. the score values (Section 2.3.2), are too low, a majority of the surface marker positions are actually regarded as missing measurements (Section 2.3.4), such that reconstructing accurate skeletal poses becomes infeasible. For instance, in case a behavioral sequence contains missing measurements over a comparably long time span, the proposed pose reconstruction framework will increase the probabilistic uncertainties associated with the three-dimensional joint positions of a reconstructed skeletal poses, which effectively deems the reconstruction itself unreliable. Thus, rigorous training of the deployed CNN is a prerequisite for the proposed framework to be functional, since it ensures that automatically generated detections of surface marker positions comprise a minimum level of accuracy.

Another drawback of the proposed pose estimation scheme is the speed at which skeletal poses are reconstructed. Since the unscented RTS smoother fuses pose information of past and future time points (Section 2.3.5), live-processing of a behavioral sequence is infeasible, i.e. it is not possible to reconstruct poses of a freely-moving animal with only a very short time delay, e.g. a few milliseconds. Nevertheless, due to the limited amount of computational steps required for

the unscented Kalman filter (Section 2.3.4), executing it in real-time could be achieved in principle [135–137]. Thus, a potential option for decreasing the processing time needed for reconstructing poses of a freely-moving animal lies in using an efficient implementation of the unscented Kalman filter. For inferring the underlying pose-encoding variables of the deployed state space model, the filter could then be used instead of the unscented RTS smoother. In this scenario, the required probabilistic hyper-parameters of the state space model could, for instance, be learned beforehand from a previously recorded behavioral sequence of the same animal. Alternatively, the EM algorithm could be executed for a single iteration based on only a few time points in the past, which would allow for repeatedly updating the probabilistic hyper-parameters of the state space model at a relatively high frequency. While only performing a single iteration of the EM algorithm will not lead to the same pose reconstruction quality, it still has the potential to yield considerable improvements, particularly when compared to a scenario where the probabilistic hyper-parameters of the state space model are not updated at all (Figure 2.20).

Lastly, the circumstance that the deployed skeleton model rigidly attaches each modeled surface marker to a single underlying joint represents another technical limitation of the proposed pose reconstruction framework (Section 2.2.2). In fact, the assumption that the motion dynamics of a surface marker are entirely governed by only a single joint simplifies the complex mechanisms, which actually determine how a point on an animal's body surface moves when the underlying bone configuration changes. However, in principle more realistic movement interactions between joints and surface markers could be implemented into the proposed pose reconstruction framework to address this limitation. Particularly, the implemented skeleton model could be extended, such that surface marker movements are influenced by more than a single joint, e.g. via ordinary linear blend skinning [138] or extensions of this method [139].

4.2.2 Analytical limitations

Besides technical limitations of the proposed pose reconstruction framework itself, there also exist analytical limitations with respect to the analyses performed in this thesis. Such an analytical limitation is given by the fact that reconstructed skeletal poses were only validated based on ground truth data of three-dimensional joint positions, obtained via MRI scans (Section 3.1), or independent measurements of two-dimensional paw positions, obtained via an FTIR imaging system (Section 3.2). Since an MRI scan only gives insights into joint locations of an immobile animal in a single static pose and an FTIR imaging system only generates data on paw positions and not the entire skeleton, an exhaustive comparison between reconstructed and ground truth three-dimensional joint positions, obtained for the entire skeleton from animals in motion, is not part of this thesis. In principle, there exist techniques for quantifying the ground truth three-dimensional locations of joints from moving animals by simultaneously using two x-ray emitters [140, 141]. However, unlike MRI scanners, the specialized equipment required for such measurements is not only cost-intensive but also less broadly available, such that respective analyses were not within the scope of this thesis.

Nevertheless, the performed comparisons between reconstructed and MRI-generated ground truth skeletons indicate that three-dimensional joint positions can be reconstructed successfully, if the three-dimensional positions of surface markers are available (Section 3.1). Knowing the precise three-dimensional surface marker positions is equivalent to accurately detecting the two-dimensional surface marker locations in at least two cameras, which allows for recovering their three-dimensional positions, e.g. by using triangulation. Thus, the proposed pose reconstruction framework is certainly capable of inferring accurate three-dimensional joint positions underneath the body surface, when an effective detection system for surface markers is given. However, since detecting two-dimensional surface marker locations is always accompanied by occasional detec-

tion errors, reconstructed three-dimensional joint positions are also expected to be erroneous at times. Nevertheless, extensively high reconstruction errors with respect to the three-dimensional joint positions are still expected to be rare, especially given that the enforced anatomical and temporal constraints have the potential to compensate for detection noise, e.g. in the form of missing measurements (Section 3.2).

Besides, being able to generate smooth and physiologically-feasible skeletal poses is a valuable asset within neuroscientific research, even when modeled and anatomical joints do not coincide perfectly, e.g. because reconstructed joint positions slightly diverge from the actual anatomical joint locations. In such cases the reconstructed joint positions can still be regarded as reasonable approximations of the underlying skeletal movements. As such they can, for instance, be interpreted as biologically-meaningful principle components of a principle component analysis [120, 142, 143]. Following this analogy, the proposed pose reconstruction framework provides an opportunity for quantifying the behavioral spectrum of a freely-behaving animal in a rather low-dimensional space. Consequently, reconstructing smooth and physiologically-feasible skeletal poses by inferring bone rotations via the proposed pose reconstruction framework allows for an unbiased yet interpretable description of behavior, irrespective of occasionally occurring reconstruction errors.

Another limitation of the analyses presented in this thesis is the lack of a broad and direct comparison to other animal pose reconstruction techniques, which detect the locations of anatomical landmarks at an animal's body surface via trained CNNs [8, 69, 76, 77]. However, since the proposed pose reconstruction framework aims at reconstructing joint positions by combining automatically-generated detections of surface marker positions with the implementation of realistic anatomical and temporal constraints, it serves as an extension to the existing CNN-based approaches rather than being a competitor. In fact, the complementary character of the proposed pose reconstruction framework persists for all currently existing surface landmark detection schemes, regardless of whether landmark detections are computed in two- [8, 76, 77] or three-dimensional space [69]. The only adjustment required for the proposed pose estimation framework to be compatible with detected three-dimensional surface marker locations is to skip the last computational step in the emission function of the state space model, in which surface markers are projected from three- to two-dimensional space (Section 2.3.1). Due to this compatibility, the computations performed by the proposed pose reconstruction framework could, in principle, be attached as downstream processing steps to any CNN-based animal pose estimation technique, which allows for detecting anatomical feature locations in two-dimensional images. As long as the upstream CNN yields overall reasonable surface marker detections with acceptable noise levels, the proposed pose reconstruction framework generates anatomically- and physiologically-feasible skeletal poses with smooth motion transitions, while being capable of compensating for sparse detection patterns or even isolated misdetections of surface markers.

4.3 Outlook

The proposed pose reconstruction framework ushers in a suite of new possibilities for quantifying poses and therefore the behavior of freely-moving animals (Section 1.2). As such, it complements recent supervised learning approaches for tracking body surfaces of animals [8, 69, 76, 77] by taking advantage of build-in mechanistic knowledge of the physical world in the form of anatomical and temporal constraints.

With respect to future short- to mid-term developments in the field of animal pose estimation it can be expected that future pose reconstruction frameworks will increasingly aim at capitalizing on respective constraints by implementing them directly into fully end-to-end trainable CNNs. For instance, adding computations for constraining poses via a realistic skeleton model to the last lay-

ers of a CNN would force the network to learn biologically-meaningful bone rotations and allow for deriving accurate anatomical feature positions on an animal's body surface. Such a synergistic approach would offer the potential to obtain skeletal poses fully end-to-end as well as to improve the overall pose estimation accuracy, since the underlying skeleton model would effectively prevent inferring physiologically-infeasible poses. Furthermore, given that the proposed pose estimation framework can be considered as a general example for how black-box deep learning frameworks can profit from build-in mechanistic world knowledge, it can be expected that incorporating respective constraints into neural networks trained for other tasks than animal pose estimation would also improve their overall capabilities. Consequently, more and more relying on mechanistic world knowledge within deep learning frameworks has the potential to not only have implications for the field of neuroscience but also many other scientific areas, where the deployment of deep neural networks is starting to increase.

In a next step, more long-term developments in the field of animal pose estimation could aim at expanding the underlying skeleton model to account for realistic muscles, such that inferring anatomical forces, which act on skeletal bones, would become feasible. When finally combined with tools for measuring neural activity (Section 1.1.1), applying such global approaches for animal pose estimation could help to incrementally bridge the knowledge gap between the neural computations conducted by the brain and the downstream muscle contractions, which ultimately dictate bone movements and therefore behavior itself. Furthermore, approaches for tracking skeletal poses and forces could enable detailed and seamless quantification of animal behavior in real-time on a broad scale.

Hence, the extent of future methods and related scientific studies can be expected to continuously increase and span behavior analyses of various animal species in environments mimicking their natural habitats, until behavior and neural activity of multiple freely-moving animals in the wild can be quantified in a simultaneous yet accurate manner. While these developments are of primary interest within the field of neuroscience, the required competences to achieve this massive undertaking are highly interdisciplinary and cover many branches of science, particularly biology, physics, computer science and mathematics. Thus, the perpetual endeavor for improving measurement techniques might also offer the opportunity to bridge conceptual, methodological and foremost ideological gaps between different scientific fields, providing a chance for creating a more unified and therefore efficient scientific community, which is nevertheless still maintaining its ability for critical thinking and scientific debate. Therefore, the depicted developments might usher in a new era of fully quantifiable animal behavior with a yet unpredictable potential for future scientific discoveries.

Bibliography

[1] D. Anderson and P. Perona, "Toward a science of computational ethology," *Neuron*, vol. 84, pp. 18–31, Oct. 2014.

[2] J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel, "Neuroscience needs behavior: Correcting a reductionist bias," *Neuron*, vol. 93, pp. 480–490, Feb. 2017.

[3] S. R. Datta, D. J. Anderson, K. Branson, P. Perona, and A. Leifer, "Computational neuroethology: A call to action," *Neuron*, vol. 104, no. 1, pp. 11–24, 2019.

[4] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. J. J. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, and U. Brose, "Automated image-based tracking and its application in ecology," *Trends in Ecology & Evolution*, vol. 29, pp. 417–428, July 2014.

[5] A. E. X. Brown and B. de Bivort, "Ethology as a physical science," *Nature Physics*, vol. 14, pp. 653–657, July 2018.

[6] N. Seethapathi, S. Wang, R. Saluja, G. Blohm, and K. P. Kording, "Movement science needs different pose tracking algorithms," *CoRR*, vol. abs/1907.10226, 2019.

[7] M. W. Mathis and A. Mathis, "Deep learning tools for the measurement of animal behavior in neuroscience," *Neurobiology of Behavior*, vol. 60, pp. 1–11, Feb. 2020.

[8] T. D. Pereira, J. W. Shaevitz, and M. Murthy, "Quantifying behavior to understand the brain," *Nature Neuroscience*, vol. 23, pp. 1537–1549, Dec. 2020.

[9] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat," *Brain Research*, vol. 34, no. 1, pp. 171–175, 1971.

[10] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, pp. 801–806, Aug. 2005.

[11] D. J. Wallace and J. N. D. Kerr, "Circuit interrogation in freely moving animals," *Nature Methods*, vol. 16, pp. 9–11, Jan. 2019.

[12] S. B. Kodandaramaiah, G. T. Franzesi, B. Y. Chow, E. S. Boyden, and C. R. Forest, "Automated whole-cell patch-clamp electrophysiology of neurons in vivo," *Nature Methods*, vol. 9, pp. 585–587, June 2012.

[13] J. P. Seymour, F. Wu, K. D. Wise, and E. Yoon, "State-of-the-art mems and microsystem tools for brain research," *Microsystems & Nanoengineering*, vol. 3, p. 16066, Jan. 2017.

[14] T. W. Margrie, M. Brecht, and B. Sakmann, “In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain,” *Pflügers Archiv*, vol. 444, pp. 491–498, July 2002.

[15] M. E. J. Obien, K. Deligkaris, T. Bullmann, D. J. Bakkum, and U. Frey, “Revealing neuronal function through microelectrode array recordings,” *Frontiers in Neuroscience*, vol. 8, 2015.

[16] G. J. Broussard, R. Liang, and L. Tian, “Monitoring activity in neural circuits with genetically encoded indicators,” *Frontiers in molecular neuroscience*, vol. 7, p. 97, 2014.

[17] Z. Wei, B.-J. Lin, T.-W. Chen, K. Daie, K. Svoboda, and S. Druckmann, “A comparison of neuronal population dynamics measured with calcium imaging and electrophysiology,” *PLoS computational biology*, vol. 16, no. 9, p. e1008198, 2020.

[18] A. Badura, X. R. Sun, A. Giovannucci, L. A. Lynch, and S. S. H. Wang, “Fast calcium sensor proteins for monitoring neural activity,” *Neurophotonics*, vol. 1, no. 2, p. 025008, 2014.

[19] J. Nakai, M. Ohkura, and K. Imoto, “A high signal-to-noise ca2+ probe composed of a single green fluorescent protein,” *Nature Biotechnology*, vol. 19, pp. 137–141, Feb. 2001.

[20] B. F. Fosque, S. Yi, D. Hod, Y. Chao-Tsung, O. Tomoko, R. Tadross Michael, P. Ronak, Z. Marta, S. Kim Douglas, B. Ahrens Misha, J. Vivek, L. Looger Loren, and R. Schreiter Eric, “Labeling of active neural circuits in vivo with designed calcium integrators,” *Science*, vol. 347, pp. 755–760, Feb. 2015.

[21] R. Kerr, V. Lev-Ram, G. Baird, P. Vincent, R. Y. Tsien, and W. R. Schafer, “Optical imaging of calcium transients in neurons and pharyngeal muscle of c. elegans,” *Neuron*, vol. 26, no. 3, pp. 583–594, 2000.

[22] T. Schrödel, R. Prevedel, K. Aumayr, M. Zimmer, and A. Vaziri, “Brain-wide 3d imaging of neuronal activity in caenorhabditis elegans with sculpted light,” *Nature Methods*, vol. 10, pp. 1013–1020, Oct. 2013.

[23] N. G. Horton, K. Wang, D. Kobat, C. G. Clark, F. W. Wise, C. B. Schaffer, and C. Xu, “In vivo three-photon microscopy of subcortical structures within an intact mouse brain,” *Nature photonics*, vol. 7, no. 3, pp. 205–209, 2013.

[24] W. Denk, “Two-photon excitation in functional biological imaging,” *Journal of Biomedical Optics*, vol. 1, pp. 296–304, July 1996.

[25] D. R. Miller, J. W. Jarrett, A. M. Hassan, and A. K. Dunn, “Deep tissue imaging with multiphoton fluorescence microscopy,” *Current opinion in biomedical engineering*, vol. 4, pp. 32–39, 2017.

[26] F. Helmchen, M. S. Fee, D. W. Tank, and W. Denk, “A miniature head-mounted two-photon microscope: High-resolution brain imaging in freely moving animals,” *Neuron*, vol. 31, pp. 903–912, Sept. 2001.

[27] J. Sawinski, D. J. Wallace, D. S. Greenberg, S. Grossmann, W. Denk, and J. N. D. Kerr, “Visually evoked activity in cortical cells imaged in freely moving animals,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 46, pp. 19557–19562, 2009.

[28] A. Klioutchnikov, D. J. Wallace, M. H. Frosz, R. Zeltner, J. Sawinski, V. Pawlak, K.-M. Voit, P. S. J. Russell, and J. N. D. Kerr, “Three-photon head-mounted microscope for imaging deep cortical layers in freely moving rats,” *Nature Methods*, vol. 17, pp. 509–513, May 2020.

[29] M. O. Pasquet, M. Tihy, A. Gourgeon, M. N. Pompili, B. P. Godsil, C. Léna, and G. P. Dugué, "Wireless inertial measurement of head kinematics in freely-moving rats," *Scientific Reports*, vol. 6, p. 35689, Oct. 2016.

[30] A. M. Michaiel, E. T. T. Abe, and C. M. Niell, "Dynamics of gaze control during prey capture in freely moving mice," *eLife*, vol. 9, p. e57458, July 2020.

[31] A. Weissbrod, A. Shapiro, G. Vasserman, L. Edry, M. Dayan, A. Yitzhaky, L. Hertzberg, O. Feinerman, and T. Kimchi, "Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment," *Nature Communications*, vol. 4, p. 2018, June 2013.

[32] F. de Chaumont, E. Ey, N. Torquet, T. Lagache, S. Dallongeville, A. Imbert, T. Legou, A.-M. Le Sourd, P. Faure, T. Bourgeron, and J.-C. Olivo-Marin, "Real-time analysis of the behaviour of groups of mice via a depth-sensing camera and machine learning," *Nature Biomedical Engineering*, vol. 3, pp. 930–942, Nov. 2019.

[33] F. de Chaumont, R. D.-S. Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J.-C. Olivo-Marin, "Computerized video analysis of social interactions in mice," *Nature Methods*, 2012.

[34] T. M. Hoogland, J. R. D. Gruijl, L. Witter, C. B. Canto, and C. I. D. Zeeuw, "Role of synchronous activation of cerebellar purkinje cell ensembles in multi-joint movement control," *Current Biology*, 2015.

[35] A. S. Machado, D. M. Darmohray, J. Fayad, H. G. Marques, and M. R. Carey, "A quantitative framework for whole-body coordination reveals specific deficits in freely walking ataxic mice," *eLife*, 2015.

[36] J. Matsumoto, S. Urakawa, Y. Takamura, R. Malcher-Lopes, E. Hori, C. Tomaz, T. Ono, and H. Nishijo, "A 3D-video-based computerized analysis of social and sexual interactions in rats," *PLOS ONE*, 2013.

[37] A. Nanjappa, L. Cheng, W. Gao, C. Xu, A. Claridge-Chang, and Z. Bichler, "Mouse pose estimation from depth images," *CoRR*, vol. abs/1511.07611, 2015.

[38] T. Nakamura, J. Matsumoto, H. Nishimaru, R. V. Bretas, Y. Takamura, E. Hori, T. Ono, and H. Nishijo, "A markerless 3D computerized motion capture system incorporating a skeleton model for monkeys," *PLOS ONE*, 2016.

[39] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3d human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, pp. 1–20, 2016.

[40] J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao, "Deep 3d human pose estimation: A review," *Computer Vision and Image Understanding*, vol. 210, p. 103225, Sept. 2021.

[41] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[42] J. D. Marshall, T. Li, J. H. Wu, and T. W. Dunn, "Leaving flatland: Advances in 3d behavioral measurement," 2021.

[43] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. C. Nabbe, I. A. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social interaction capture," *CoRR*, vol. abs/1612.03153, 2016.

[44] T. McLaughlin and S. S. Sumida, "The morphology of digital creatures," in *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, (New York, NY, USA), p. 1–es, Association for Computing Machinery, 2007.

[45] E. Muybridge, "Animal locomotion: An electro-photographic investigation of consecutive phases of animal movements," 1887.

[46] K. Wei and K. P. Kording, "Behavioral tracking gets real," *Nature Neuroscience*, vol. 21, pp. 1146–1147, Sept. 2018.

[47] A. Mathis, S. Schneider, J. Lauer, and M. W. Mathis, "A primer on motion capture with deep learning: Principles, pitfalls, and perspectives," *Neuron*, vol. 108, no. 1, pp. 44–65, 2020.

[48] R. Poppe, "Vision-based human motion analysis: An overview," *Special Issue on Vision for Human-Computer Interaction*, vol. 108, pp. 4–18, Oct. 2007.

[49] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. C-22, no. 1, pp. 67–92, 1973.

[50] T. Serre, "Deep learning: The good, the bad, and the ugly," *Annu. Rev. Vis. Sci.*, vol. 5, pp. 399–426, Sept. 2019.

[51] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, pp. 55–79, Jan. 2005.

[52] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1014–1021, 2009.

[53] M. Eichner, V. Ferrari, and S. Zurich, "Better appearance models for pictorial structures.," in *Bmvc*, vol. 2, p. 5, Citeseer, 2009.

[54] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

[55] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, 2012.

[57] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," *CoRR*, vol. abs/1511.06645, 2015.

[58] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," *CoRR*, vol. abs/1605.03170, 2016.

[59] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” 2017.

[60] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand keypoint detection in single images using multiview bootstrapping,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4645–4653, 2017.

[61] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “Openpose: Realtime multi-person 2d pose estimation using part affinity fields,” 2019.

[62] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, “Scape: Shape completion and animation of people,” *ACM Trans. Graph.*, vol. 24, p. 408–416, jul 2005.

[63] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: A skinned multi-person linear model,” *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, pp. 248:1–248:16, Oct. 2015.

[64] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, “Keep it smpl: Automatic estimation of 3d human pose and shape from a single image,” in *Computer Vision - ECCV 2016*, (Cham), pp. 561–578, Springer International Publishing, 2016.

[65] L. Sigal, A. Balan, and M. J. Black, “HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion,” *International Journal of Computer Vision*, vol. 87, pp. 4–27, Mar. 2010.

[66] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: New benchmark and state of the art analysis,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[67] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[68] P. C. Bala, B. R. Eisenreich, S. B. M. Yoo, B. Y. Hayden, H. S. Park, and J. Zimmermann, “Automated markerless pose estimation in freely moving macaques with openmonkeystudio,” *Nature Communications*, vol. 11, p. 4560, Sept. 2020.

[69] T. W. Dunn, J. D. Marshall, K. S. Severson, D. E. Aldarondo, D. G. C. Hildebrand, S. N. Chettih, W. L. Wang, A. J. Gellis, D. E. Carlson, D. Aronov, W. A. Freiwald, F. Wang, and B. P. Ölveczky, “Geometric deep learning enables 3d kinematic profiling across species and environments,” *Nature Methods*, vol. 18, pp. 564–573, May 2021.

[70] D. Joska, L. Clark, N. Muramatsu, R. Jericevich, F. Nicolls, A. Mathis, M. W. Mathis, and A. Patel, “Acinoset: A 3d pose estimation dataset and baseline models for cheetahs in the wild,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13901–13908, 2021.

[71] J. D. Marshall, U. Klibaite, A. Gellis, D. E. Aldarondo, B. Olveczky, and T. W. Dunn, “The PAIR-r24m dataset for multi-animal 3d pose estimation,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[72] S. Kearney, W. Li, M. Parsons, K. I. Kim, and D. Cosker, “Rgbd-dog: Predicting canine pose from rgbd sensors,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[73] L. A. Bolaños, D. Xiao, N. L. Ford, J. M. LeDue, P. K. Gupta, C. Doebeli, H. Hu, H. Rhodin, and T. H. Murphy, "A three-dimensional virtual mouse generates synthetic training data for behavioral analysis," *Nature Methods*, vol. 18, pp. 378–381, Apr. 2021.

[74] J. Deane, S. Kearney, K. I. Kim, and D. Cosker, "Dynadog+t: A parametric animal model for synthetic canine image generation," 2021.

[75] A. Mathis, T. Biasi, S. Schneider, M. Yüksekgönül, B. Rogers, M. Bethge, and M. W. Mathis, "Pretraining boosts out-of-domain robustness for pose estimation," 2020.

[76] A. Mathis, P. Mamidanna, K. M. Cury, T. Ab, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, 2018.

[77] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe, and I. D. Couzin, "Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning," *eLife*, vol. 8, p. e47994, Oct. 2019.

[78] T. D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz, "Fast animal pose estimation using deep neural networks," *Nature Methods*, vol. 16, pp. 117–125, Jan. 2019.

[79] M. Bartul, A. Dunn Benjamin, T. Tuce, B. V. P. T. N. C. Srikanth, and R. Whitlock Jonathan, "Efficient cortical coding of 3d posture in freely behaving rats," *Science*, vol. 362, pp. 584–589, Nov. 2018.

[80] J. D. Marshall, D. E. Aldarondo, T. W. Dunn, W. L. Wang, G. J. Berman, and B. P. Ölveczky, "Continuous whole-body 3d kinematic recordings across the rodent behavioral repertoire," *Neuron*, vol. 109, pp. 420–437.e8, Feb. 2021.

[81] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, "Using deeplabcut for 3d markerless pose estimation across species and behaviors," *Nature Protocols*, vol. 14, pp. 2152–2176, July 2019.

[82] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: A toolkit for robust markerless 3d pose estimation," *Cell Reports*, vol. 36, p. 109730, Sept. 2021.

[83] X. Liu, S.-y. Yu, N. A. Flierman, S. Loyola, M. Kamermans, T. M. Hoogland, and C. I. De Zeeuw, "Optiflex: Multi-frame animal pose estimation combining deep learning with optical flow," *Frontiers in Cellular Neuroscience*, vol. 15, 2021.

[84] C. Zimmermann, A. Schneider, M. Alyahyay, T. Brox, and I. Diester, "Freipose: A deep learning framework for precise animal motion capture in 3d spaces," *bioRxiv*, p. 2020.02.27.967620, Jan. 2020.

[85] S. Zuffi, A. Kanazawa, D. Jacobs, and M. J. Black, "3D menagerie: Modeling the 3D shape and pose of animals," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, pp. 5524–5532, IEEE, July 2017.

[86] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla, "Creatures great and smal: Recovering the shape and motion of animals from video," in *Computer Vision - ACCV 2018*, (Cham), pp. 3–19, Springer International Publishing, 2019.

[87] S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black, “Three-D safari: Learning to estimate zebra pose, shape, and texture from images ”in the wild”,” in *International Conference on Computer Vision*, pp. 5358–5367, IEEE, Oct. 2019.

[88] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla, “Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop,” in *ECCV*, 2020.

[89] C. Li, N. Ghorbani, S. Broomé, M. Rashid, M. J. Black, E. Hernlund, H. Kjellström, and S. Zuffi, “hsmal: Detailed horse shape and pose reconstruction for motion pattern recognition,” 2021.

[90] A. Gosztolai, S. Günel, V. Lobato-Ríos, M. Pietro Abrate, D. Morales, H. Rhodin, P. Fua, and P. Ramdya, “Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals,” *Nature Methods*, vol. 18, pp. 975–981, Aug. 2021.

[91] G. A. Kane, G. Lopes, J. L. Saunders, A. Mathis, and M. W. Mathis, “Real-time, low-latency closed-loop feedback using markerless posture tracking,” *eLife*, vol. 9, p. e61909, Dec. 2020.

[92] I. Sarkar, I. Maji, C. Omprakash, S. Stober, S. Mikulovic, and P. Bauer, “Evaluation of deep lift pose models for 3d rodent pose estimation based on geometrically triangulated data,” 2021.

[93] S. Günel, H. Rhodin, D. Morales, J. Campagnolo, P. Ramdya, and P. Fua, “Deepfly3d, a deep learning-based approach for 3d limb and appendage tracking in tethered, adult drosophila,” *eLife*, vol. 8, p. e48571, Oct. 2019.

[94] L. Zhang, T. Dunn, J. Marshall, B. Olveczky, and S. Linderman, “Animal pose estimation from video data with a hierarchical von mises-fisher-gaussian model,” in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* (A. Banerjee and K. Fukumizu, eds.), vol. 130 of *Proceedings of Machine Learning Research*, pp. 2800–2808, PMLR, 13–15 Apr 2021.

[95] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. USA: CRC Press, Inc., 1st ed., 1994.

[96] G. Pons-Moll and B. Rosenhahn, “Ball joints for marker-less human motion capture,” in *IEEE Workshop on Applications of Computer Vision (WACV)*, Dec. 2009.

[97] L. Euler, “De motu corporum circa punctum fixum mobilium,” *Opera postuma*, vol. 2, pp. 43–62, 17XX. 1862.

[98] H. Cheng and K. C. Gupta, “An Historical Note on Finite Rotations,” *Journal of Applied Mechanics*, vol. 56, pp. 139–145, 03 1989.

[99] F. S. Grassia, “Practical parameterization of rotations using the exponential map,” *J. Graph. Tools*, vol. 3, p. 29–48, mar 1998.

[100] O. Rodrigues, “Des lois géométriques qui régissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire,” *J. Math. Pures Appl*, vol. 5, no. 380-400, p. 5, 1840.

[101] W. R. Hamilton, “On quaternions; or on a new system of imaginaries in algebra,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, no. 163, pp. 10–13, 1844.

[102] Intel, *The OpenCV Reference Manual*. Intel, 4.5.2 ed., 2021.

[103] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[104] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018.

[105] M. A. Branch, T. F. Coleman, and Y. Li, “A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems,” *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 1–23, 1999.

[106] E. Jones, T. Oliphant, P. Peterson, *et al.*, “SciPy: Open source scientific tools for Python,” 2001.

[107] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: Elsevier, 1976.

[108] R. Brockett, A. Stokes, and F. Park, “A geometrical formulation of the dynamical equations describing kinematic chains,” in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 637–641 vol.2, 1993.

[109] J. Chen, S. Nie, and Q. Ji, “Data-free prior model for upper body pose estimation and tracking,” *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4627–4639, 2013.

[110] P. M. Prodinger, P. Foehr, D. Bürklein, O. Bissinger, H. Pilge, K. Kreutzer, R. von Eisenhart-Rothe, and T. Tischer, “Whole bone testing in small animals: systematic characterization of the mechanical properties of different rodent bones available for rat fracture models,” *European Journal of Medical Research*, vol. 23, 2018.

[111] C. D. Newton and D. Nunamaker, *Textbook of small animal orthopaedics*. 1985.

[112] I. Akhter and M. J. Black, “Pose-conditioned joint angle limits for 3d human pose reconstruction,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 1446–1455, IEEE Computer Society, 2015.

[113] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2e ed., 2006.

[114] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, 2019.

[115] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal of Scientific Computing*, vol. 16, pp. 1190–1208, Sept. 1995.

[116] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[117] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, and D. Andina, “Deep learning for computer vision: A brief review,” *Intell. Neuroscience*, vol. 2018, jan 2018.

[118] Z. Zhao, P. Zheng, S. Xu, and X. Wu, “Object detection with deep learning: A review,” *CoRR*, vol. abs/1807.05511, 2018.

[119] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

[120] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 ed., 2006.

[121] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[122] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *Proceedings of 1995 American Control Conference - ACC’95*, vol. 3, pp. 1628–1632 vol.3, 1995.

[123] S. Särkkä, *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks, Cambridge University Press, 2013.

[124] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, USA: Cambridge University Press, second ed., 1992.

[125] J. Kokkala, A. Solin, and S. Särkkä, “Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems,” 2015.

[126] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[127] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. Springer, 4th ed., 2017.

[128] M. Šimandl and J. Duník, “Design of derivative-free smoothers and predictors,” (Newcastle), pp. 1–6, IFAC, 2006.

[129] S. Särkkä, “Unscented rauch–tung–striebel smoother,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.

[130] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[131] R. L. Maynard and N. Downes, *Anatomy and histology of the laboratory rat in toxicology and biomedical research. ID - 20193201650*. London: Academic Press, 2019.

[132] E. J. Ambrose, “A surface contact microscope for the study of cell movements,” *Nature*, vol. 178, pp. 1194–1194, Nov. 1956.

[133] C. S. Mendes, I. Bartos, Z. Márka, T. Akay, S. Márka, and R. S. Mann, “Quantification of gait parameters in freely walking rodents,” *BMC Biology*, vol. 13, p. 50, July 2015.

[134] B. Fornberg, “Generation of finite difference formulas on arbitrarily spaced grids,” *Mathematics of Computation*, vol. 51, no. 184, pp. 699–699, 1988.

[135] V. M. Fico, C. P. Arribas, A. R. Soaje, M. A. M. Prats, S. R. Utrera, A. L. R. Vazquez, and L. M. P. Casquet, “Implementing the unscented kalman filter on an embedded system: A lesson learnt,” in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 2010–2014, 2015.

[136] T. Cantelobre, C. Chahbazian, A. Croux, and S. Bonnabel, “A real-time unscented kalman filter on manifolds for challenging auv navigation,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2309–2316, 2020.

[137] M. Impraimakis and A. W. Smyth, “An unscented kalman filter method for real time input-parameter-state estimation,” *Mechanical Systems and Signal Processing*, vol. 162, p. 108026, Jan. 2022.

[138] J. P. Lewis, M. Cordner, and N. Fong, “Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (USA), p. 165–172, ACM Press/Addison-Wesley Publishing Co., 2000.

[139] X. C. Wang and C. Phillips, “Multi-weight enveloping: Least-squares approximation techniques for skin animation,” in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’02, (New York, NY, USA), p. 129–138, Association for Computing Machinery, 2002.

[140] E. L. Brainerd, D. B. Baier, S. M. Gatesy, T. L. Hedrick, K. A. Metzger, S. L. Gilbert, and J. J. Crisco, “X-ray reconstruction of moving morphology (xromm): precision, accuracy and applications in comparative biomechanics research,” *J. Exp. Zool.*, vol. 313A, pp. 262–279, June 2010.

[141] D. D. Moore, J. D. Walker, J. N. MacLean, and N. G. Hatsopoulos, “Validating marker-less pose estimation with 3d x-ray radiography,” *bioRxiv*, p. 2021.06.15.448541, Jan. 2021.

[142] 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.

[143] H. Hotelling, “Analysis of a complex of statistical variables into principal components.,” *Journal of Educational Psychology*, vol. 24, pp. 498–520, 1933.

[144] K. B. Petersen and M. S. Pedersen, “The matrix cookbook,” nov 2012. Version 20121115.

Appendix

A.1 Evaluating expectation values of log-transformed normal distributions

Given a d-dimensional normal distribution $\mathrm{p_{norm}}$ with expectation value $\mu_y \in \mathbb{R}^d$ and covariance matrix $V_y \in \mathbb{R}^{d\times d}$, evaluating it for a normally distributed random variable $y \sim \mathcal{N}(m, \Sigma)$ gives the following expression:

$$\mathrm{p_{norm}}\left(y|\mu_y, V_y\right) = (2\pi)^{-\frac{d}{2}} \det\left(V_y\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y-\mu_y)^T {V_y}^{-1}\left(y-\mu_y\right)\right), \tag{A.1}$$

where $\det\left(V_y\right) \in \mathbb{R}$ denotes the determinant of matrix V_y. Applying a logarithmic transformation to Equation A.1 yields

$$\ln \mathrm{p_{norm}}\left(y|\mu_y, V_y\right) = -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\mathrm{tr}\left({V_y}^{-1}\left(y-\mu_y\right)\left(y-\mu_y\right)^T\right), \tag{A.2}$$

where $\mathrm{tr}\left(V_y\right) \in \mathbb{R}$ denotes the trace of matrix V_y. Consequently, noticing that $\mathbb{E}\left[yy^T\right] = \Sigma + mm^T$ [125, 144] allows for calculating the expectation value of Equation A.2 with respect to y:

$$\mathbb{E}\left[\ln \mathrm{p_{norm}}\left(y|\mu_y, V_y\right)\right] = -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\mathrm{tr}\left({V_y}^{-1}\mathbb{E}\left[\left(y-\mu_y\right)\left(y-\mu_y\right)^T\right]\right) \tag{A.3}$$

$$= -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\mathrm{tr}\left({V_y}^{-1}\left(\Sigma + \left(m-\mu_y\right)\left(m-\mu_y\right)^T\right)\right). \tag{A.4}$$

A.2 Derivatives

Given a d-dimensional vector $v \in \mathbb{R}^d$, two symmetric matrices $M \in \mathbb{R}^{d\times d}$ and $C \in \mathbb{R}^{d\times d}$ as well as a scalar $c \in \mathbb{R}$, the following derivatives exist [144]:

$$\frac{\partial}{\partial v}\mathrm{tr}\left(Cvv^T\right) = Cv + C^T v = 2Cv \tag{A.5}$$

$$\frac{\partial}{\partial M}\ln\det\left(cM\right) = M^{-1} \tag{A.6}$$

$$\frac{\partial}{\partial M}\mathrm{tr}\left(M^{-1}C\right) = -\left(M^T\right)^{-1}C^T\left(M^T\right)^{-1} = -M^{-1}CM^{-1}. \tag{A.7}$$

A.3 Statistics

For computing the p-values stated in this thesis it is assumed that the underlying data is uncorrelated. This also applies for data obtained from recorded images, such that potential correlations between images from different but consecutive time points are ignored.

A.4 Quantifying periodic gait cycles

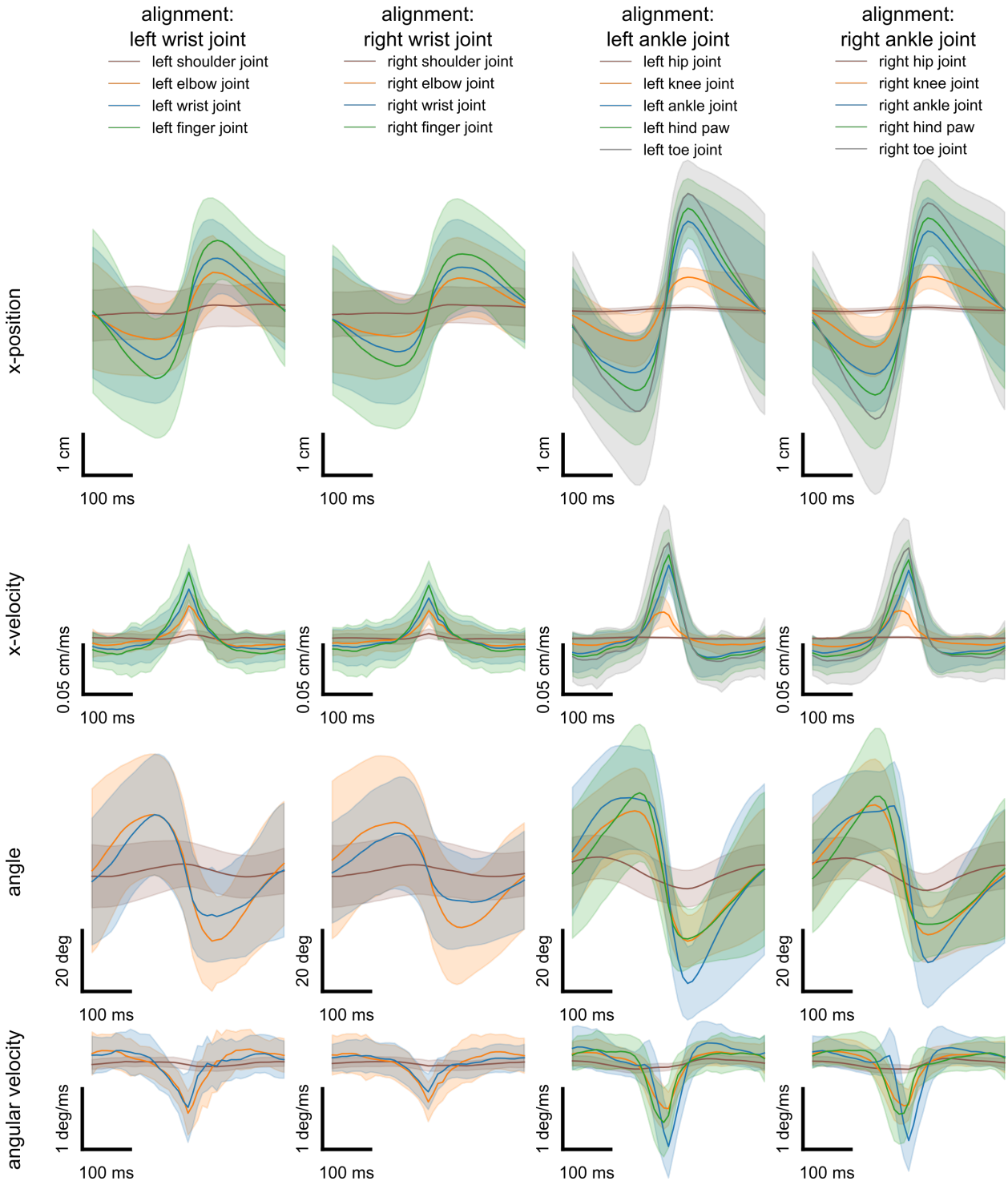

Figure A.1: Same as Figure 3.13 with the exception that traces belong to single limbs.

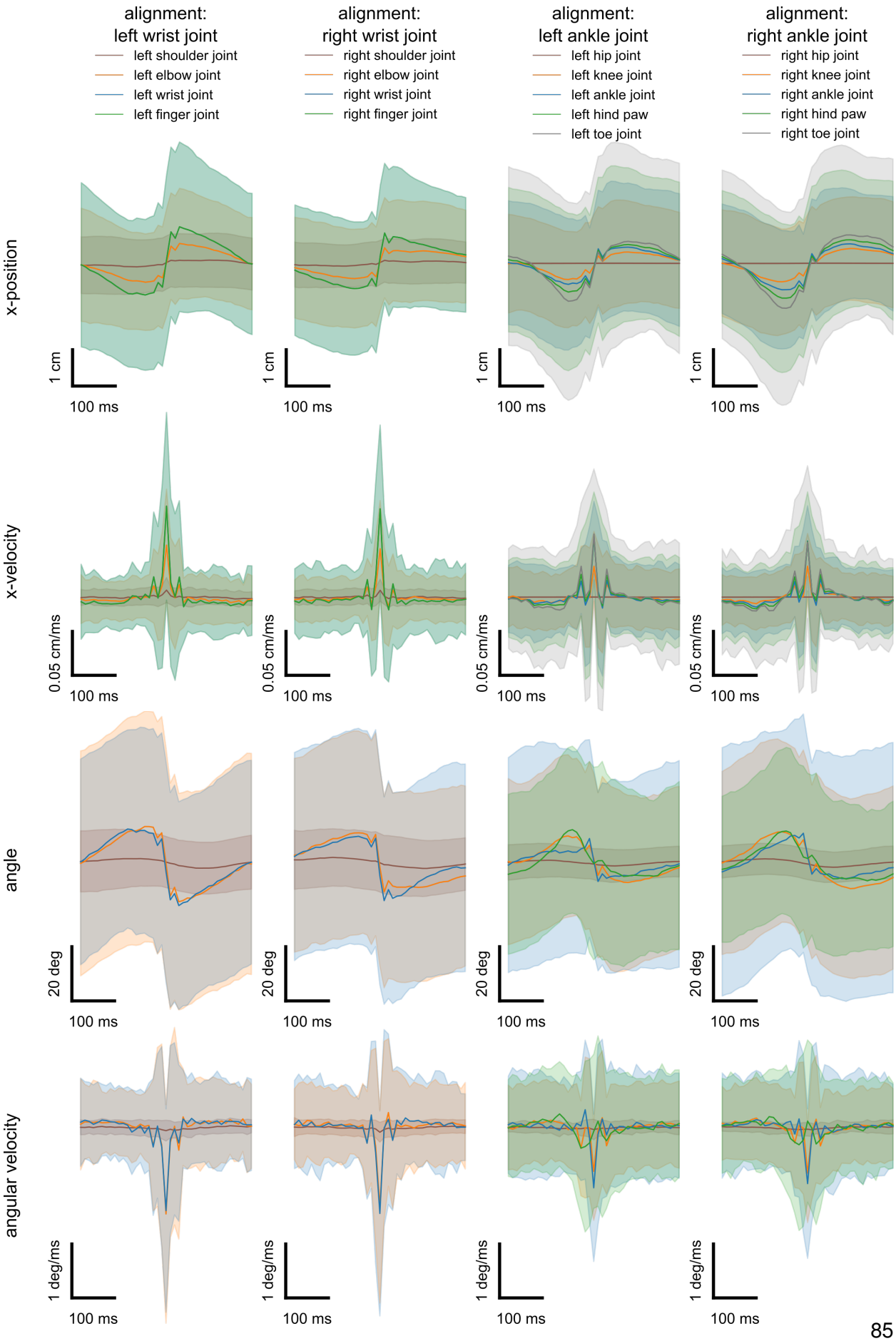

Figure A.2: Same as Figure 3.14 with the exception that traces belong to single limbs.

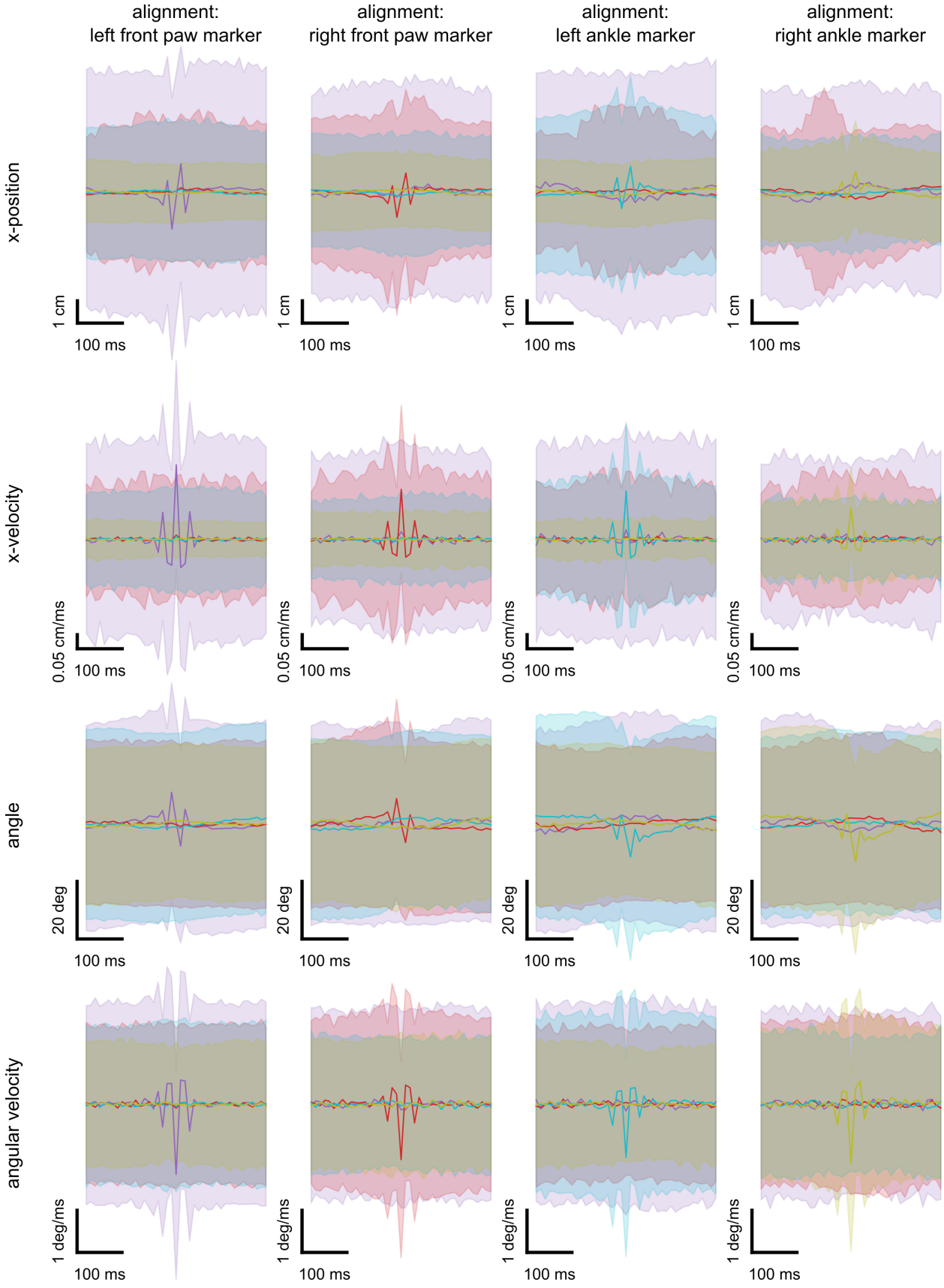

Figure A.3: Same as Figure 3.13, except that triangulated surface marker positions were used for analyzing gait. Individual traces therefore correspond to the left front paw (purple), right front paw (red), left ankle (cyan) and right ankle (yellow) marker.

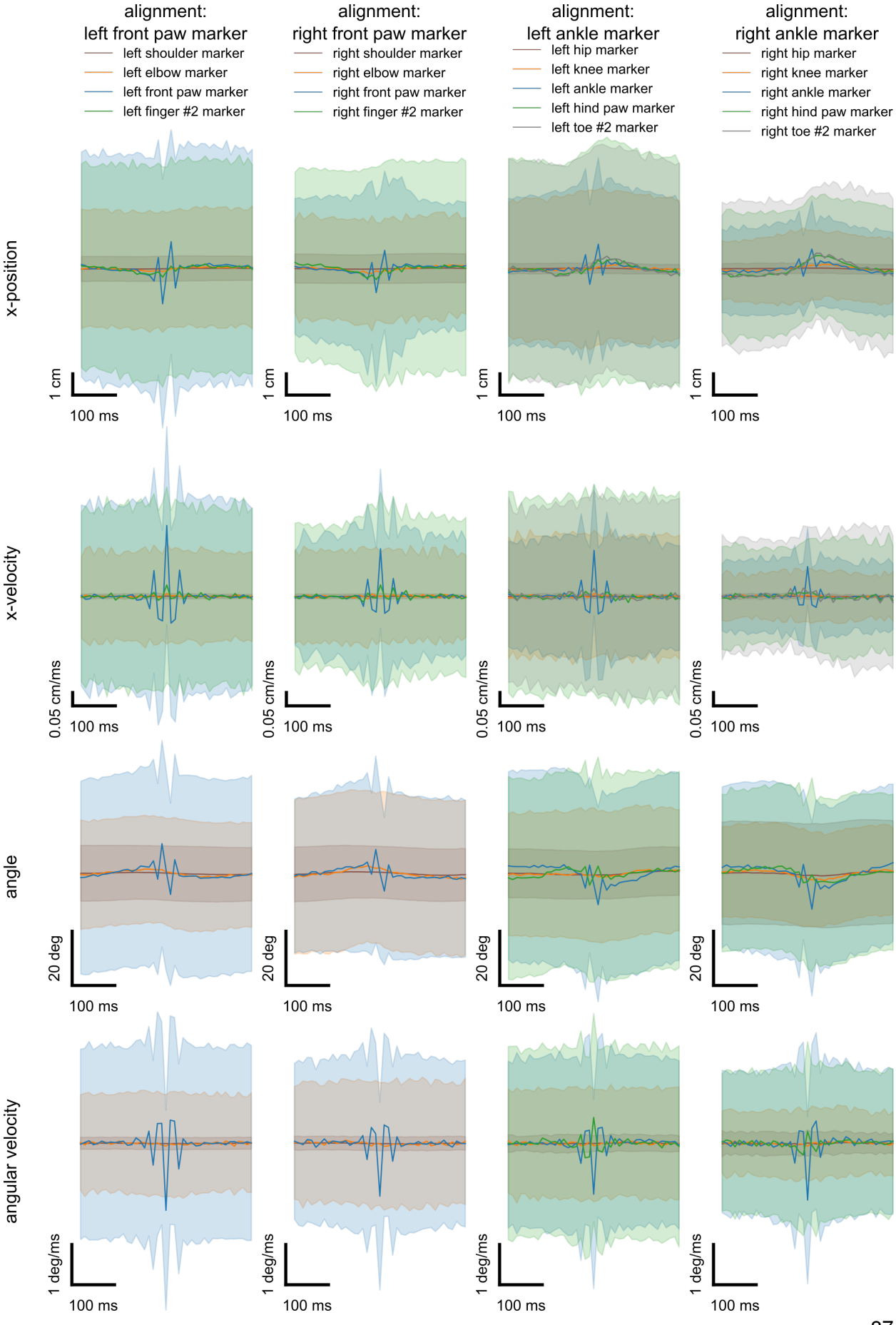

Figure A.4: Same as Figure A.3 with the exception that traces belong to single limbs.