Patterns in Motion - From the Detection of Primitives to Steering Animations

Dissertation

zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Dipl.-Math. Anna Magdalena Vögele

aus Herdecke

Bonn, Februar 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Dekan: Prof. Dr. U.-G. Meißner

1. Referent: Prof. Dr. Reinhard Klein
2. Referent: Prof. Dr. Andreas Weber

Tag der mündlichen Prüfung: 12. September 2016
Erscheinungsjahr: 2016

CONTENTS

ZUSAMMENFASSUNG

In den letzten Jahrzehnten hat sich die Welt der Technologie rapide entwickelt. Beispielhaft für diese Entwicklung ist die wachsende Zahl erschwinglicher Methoden zum Aufzeichnen neuer und immer größerer Datenmengen. Die sich daraus ergebenden Massen multivariater und hochdimensionaler Daten stellen Forschung wie Industrie vor neuartige Probleme.

Diese Arbeit ist der Entwicklung neuer Verfahren zur Verarbeitung multivariater Zeitreihen gewidmet und stellt sich damit einer großen Herausforderung, welche unmittelbar mit dem neuen Feld der sogenannten *Data Science* verbunden ist. In ihr werden ein Reihe von verschiedenen Verfahren zur Verarbeitung multivariater Zeitserien eingeführt. Die verschiedenen Verfahren gehen jeweils auf unterschiedliche Anforderungen und typische Stadien der Datenverarbeitung ein und reichen von Vorverarbeitung bis zur Nachverarbeitung und darüber hinaus zur Wiederverwertung.

Viele der vorgestellten Techniken eignen sich zur Verarbeitung allgemeiner multivariater Zeitreihen. Allerdings wurden hier eine Anzahl verschiedenartiger Aufnahmen von menschlichen und tierischen Subjekte ausgewählt, welche als Vertreter für allgemeine multivariate Zeitreihen gelten können. Zu den unterschiedlichen Modalitäten der Aufnahmen gehören Motion Capture Daten, Beschleunigungen, Gyroskopdaten, Elektromyographie, Tiefenbilder (Kinect) und animierte $3D$ -Meshes.

Es ist das Ziel dieser Arbeit, am Beispiel der multivariaten Bewegungsdaten ein tieferes Verständnis für den Umgang mit multivariaten Zeitreihen zu vermitteln. Um jedoch einen Überblick über die Materie zu wahren, folgt sie jedoch einer grundlegenden und allgemeinen Pipeline. Diese Pipeline wurde als Leitfaden für die Verarbeitung von Zeitreihen entwickelt und ist der erste Beitrag dieser Arbeit. Jeder weitere Teil der Arbeit behandelt eine von drei größeren Stationen in der Pipeline, welche sich unter unter die Themen Segmentierung, Analyse und Synthese eingliedern lassen. Beispiele verschiedener Datenmodalitäten und Anforderungen an ihre Verarbeitung erläutern die jeweiligen Verfahren.

Ein wichtiger Beitrag dieser Arbeit ist ein neuartiges Verfahren zur zeitlichen Segmentierung von Bewegungsdaten. Dieses basiert auf der Idee der Selbst-Ähnlichkeit von Bewegungsdaten und ist in der Lage, verschiedenste Bewegungsdaten voll-automatisch in unterschiedliche Aktivitäten und Bewegungs-Primitive zu zerlegen.

Die Beispiele für die Analyse multivariater Zeitreihen spiegeln die Rolle der Datenanalyse in verschiedenen interdisziplinären Zusammenhänge besonders wider und illustrieren auch die Vielfalt der Anforderungen, die sich in interdisziplinären Kontexten auftun.

Schließlich wird das Problem der Synthese multivariater Zeitreihen unter Verwendung eines graph-basierten und eines Steering Beispiels diskutiert. Synthese ist insofern ein wichtiger Schritt in der Datenverarbeitung, da sie es erlaubt, auf kontrollierte Art neue Daten aus vorhandenen zu erzeugen. Dies macht die Nutzung bestehender Datensätze und den Zugang zu dichteren Datenmodellen möglich, wodurch Alternativen zur ansonsten zeitaufwendigen manuellen Verarbeitung aufgezeigt werden.

Abstract

In recent decades, the world of technology has developed rapidly. Illustrative of this trend is the growing number of affordable methods for recording new and bigger data sets. The resulting masses of multivariate and high-dimensional data represent a new challenge for research and industry.

This thesis is dedicated to the development of novel methods for processing multivariate time series data, thus meeting this *Data Science* related challenge. This is done by introducing a range of different methods designed to deal with time series data. The variety of methods reflects the different requirements and the typical stage of data processing ranging from pre-processing to post-processing and data recycling.

Many of the techniques introduced work in a general setting. However, various types of motion recordings of human and animal subjects were chosen as representatives of multi-variate time series. The different data modalities include Motion Capture data, accelerations, gyroscopes, electromyography, depth data (Kinect) and animated $3D$-meshes.

It is the goal of this thesis to provide a deeper understanding of working with multi-variate time series by taking the example of multi-variate motion data. However, in order to maintain an overview of the matter, the thesis follows a basic general pipeline. This pipeline was developed as a guideline for time series processing and is the first contribution of this work. Each part of the thesis represents one important stage of this pipeline which can be summarized under the topics segmentation, analysis and synthesis. Specific examples of different data modalities, processing requirements and methods to meet those are discussed in the chapters of the respective parts.

One important contribution of this thesis is a novel method for temporal segmentation of motion data. It is based on the idea of self-similarities within motion data and is capable of unsupervised segmentation of range of motion data into distinct activities and motion primitives.

The examples concerned with the analysis of multi-variate time series reflect the role of data analysis in different inter-disciplinary contexts and also the variety of requirements that comes with collaboration with other sciences. These requirements are directly connected to current challenges in data science.

Finally, the problem of synthesis of multi-variate time series is discussed using a graph-based example and examples related to rigging or steering of meshes. Synthesis is an important stage in data processing because it creates new data from existing ones in a controlled way. This makes exploiting existing data sets and and access of more condensed data possible, thus providing feasible alternatives to otherwise time-consuming manual processing.

ACKNOWLEDGEMENTS

First of all, this thesis is not a chronological essay of the research done in the past years. At the first attempts to data-driven motion synthesis, I found that I knew too little about the intrinsic properties of motion. When consequently trying to study and advance tools and methods for analysis of motion data, I found that a satisfactory basic representation of the material was missing. In that sense, discovering the prominent role of segmentation in all data processing was pretty much a „journey back to basics“.

In a way, the same holds true for the personal journey that writing this thesis meant to me. When I started to work in computer graphics, it was Prof. Dr. Reinhard Klein who offered my a job with his group. Thank you for supporting my decision to explore a field I was completely unfamiliar with and for giving me the motivation to pull through.

When I found that my heart was in data-driven computer animation, it was Dr. Björn Krüger who gave me the opportunity to work on the GeMMQuad project (funded by the German Research Foundation (DFG) under research grants KR 4309/2-1). Dear Björn, thank you so much for having me on your team, for believing in my work and for always having my back.

I would also like to thank Dr. Rebeka R. Zsoldos and Prof. Dr. Theresia Licka for the most interesting time I had on the GeMMQuad project and for all the invaluable lessons on working with people, horses and other friendly creatures.

I would like to thank my co-authors and also the anonymous reviewers of all our papers. Your input has made me strive to become a better writer, scientist and colleague. I am particularly indebted to Dr. Jürgen Bernard, Nils Wilhelm and Dr. Qaiser Riaz for all the work and encouragement.

When times are tough, a sense of humor is vital. I would like to thank Thomas Kuhlmann for pointing this out to me whenever necessary. And I thank my ’partners in crime’, Stefan Hartmann, Michael Weinmann and Heinz-Christian Steinhausen for reading my novellas and stage plays and cheering me up with jokes and nonsense. Especially, I would like to thank Dr. Raoul Wessel: You have been a great colleague and a true friend when I needed one most and your passion for both science and human motion never ceases to amaze and inspire me.

I would like to thank Dr. Katharina Heine, Pauline Degenhardt, Stephanie

Köhler, Anne-Sophie Matheron and Tatjana Krym for being my friends.

Felix Hoffmann, you have been a great partner in so many ways. Thank you for all the support, for always being by my side and for making things work. With you I can do anything.

I would like to express my deepest gratitude to my parents: You may not realize it but you have the greatest part in this. This work would not have been possible without your education, love and support. Thank you for everything you have committed.

CHAPTER 1

MOTIVATION

The whole problem with the world is that fools and fanatics are always so certain of themselves, and wiser people so full of doubts.

Bertrand Russell

During the last decades, the world of technology has rapidly developed. One example of such developments is the growing number of devices for recording new and more data at increasing speed. Multiple opportunities of recording multivariate time series have arisen in the professional as well as the consumer sector. In particular, with this multitude of tools available, capturing human/robot/animal motion has become easy. At the same time, techniques for motion processing have become extremely important. Many techniques for pre-processing (e. g. pre-processing of motion capture data, Bruderlin and Williams [BW95], segmentation of motion capture data, Barbič et al. [BP$^+$04]), analysis (Aggarwal and Qai [AC99], Poppe [Pop07]) and further processing (e. g. synthesis (Krüger [Krü12], Min and Chai [MC12] or retrieval, Müller and Röder [MR06]) of motion data have developed.

The goal of this thesis is to give a better understanding of the tools and methods needed to investigate and exploit patterns in motion data. This is a very complex and wide subject as motion data depend on a range of conditions. To begin with, subjects that cause motion can vary (e. g. animals, humans, robots) and the types of motion they display can be largely different (e. g. full-body vs, partial). At the same time, there are many different modalities of recording motion which are defined by the choice of sensors and their combination, capture setups and calibration. Finally, the representations of motions depends not only on the choice of the above-mentioned but allows for the definition of features and methods used to process and the acquired data.

Given such a number of degrees of freedom, it is not possible to do an exhaus-

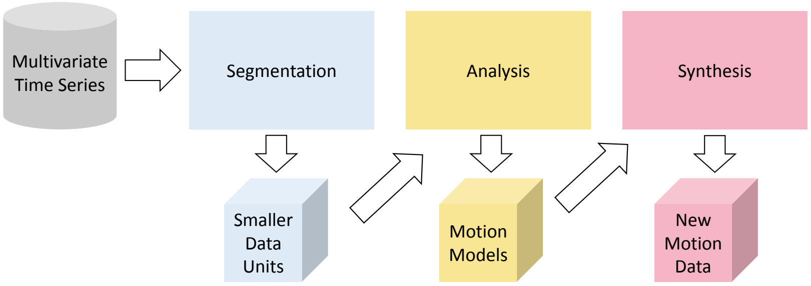


Figure 1.1: Overview of pipeline. Relationship between three subjects which are also reflected by the three parts of this thesis.

tive survey on all possible variations. Instead, a selection of different kinds of data modalities, different kinds of moving subjects and different kinds of motion will be given as examples throughout this thesis. The stages of data processing will be introduced following a pipeline (see Figure 1.1). This exemplary pipeline illustrates how multivariate time series can be processed starting at the data base level and proceeding through different stages of processing on synthesis. This provides an idea how the complex task of data processing can be broken down into smaller tasks. There are three more general parts in this thesis representing three stages in the pipeline: segmentation (Part I), analysis (Part II) and synthesis (Part III). The chapters of each part introduce specific tasks representing challenges associated with the corresponding stage.

Part I

Segmentation

CHAPTER 2

SEGMENTATION OF MULTIVARIATE TIME SERIES

Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things.

Isaac Newton

The term *Motion Capture* or *MoCap* which was once associated with producing special effects for films and video games, is used today in diverse contexts ranging from health care to consumer electronics. It refers to a number of techniques that allow the recording of data representing any aspect of human, animal or robot motion. Examples of systems and devices performing this task are: optical motion capture (with or without markers), video cameras, accelerometers, gyroscopes, EMG sensors etc.

Nowadays, it is easy as never before for consumers to monitor any activity with devices like the fitbit or kinect. Everyone has a lot of options from health care to entertainment. The ever-increasing simplicity to capture data by different sensor modalities, along with the sheer amount of existing recorded data creates a demand for motion analysis methods that are computationally efficient, yet with minimal human input.

Dividing streams of motion data into perceptually meaningful segments is a precursor to almost all analysis and synthesis methods. For example, creating a statistical motion model calls for data already preprocessed into well-defined activity segments. Further segmentation of these activities into individual cycles is greatly beneficial for action recognition, especially when repetitions should be counted, or for compressing motion data. However, the quantity of captured data does not always allow for time-consuming manual segmentation. As such, unsupervised segmentation and learning of motion primitives is a topic of interest that has been addressed in both the computer vision [RA00, ZMI06, VMP03, LF04, ZlTH08, ZlTH13, HLDlT11, AK14] and the computer graph-

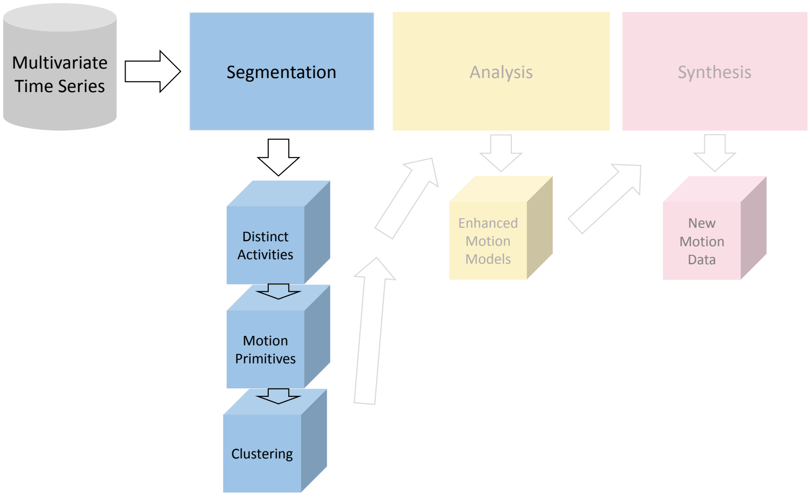


Figure 2.1: Segmentation of multi-variate time series.

ics [BP$^+$04, MRC05, LM06, VKK14, HCHMT15] communities.

We propose a segmentation method which automatically identifies **distinct actions** within motion sequences and further decompose such actions into atomic **motion primitives**, all in an unsupervised fashion. For example, in a sequence of a person who first walks and then breaks into a run, we can separate walking from running, as well as the individual steps of the walk and run. We identify both the actions and the motion primitives by exploiting the self-similarities that exist in motion sequences.

The proposed segmentation techniques can be found in [VKK14, KVW$^+$15]. The former is about segmenting optical motion capture data, the latter generalized to arbitrary multivariate motion capture data. The technique is given as an efficiently solvable graph problem. To further improve computational efficiency, we employ a Neighborhood Graph [KTWZ10] which was originally designed as a fast substitute for dynamic time warping. In summary, our main contributions are:

- A novel technique for feature bundling.
- Segmentation of activities allowing distinguish between periodic structures and non-periodic transitions.
- Subdivision of activities into self-similar units, i. e. one step in a walking sequence, while efficiently detecting the correct number of repetitions of each primitive.

- Detection and exploitation of motion symmetries.
- Solving these tasks efficiently.
- Robustness and applicability to motion data from varying sensor modalities.

First, we propose a novel feature bundling technique for pre-processing motion features. The feature bundling allows for compact model representations of the motions, as well as robustness to noise, to accommodate modalities such as marker-less motion capture or accelerometers. Second, we introduce a notion of motion symmetry, and exploit this as a means of refining primitive detection. Considering symmetry often leads to primitives with more physical meaning—for instance walking cycles can be split into left and right steps.

Third, we propose a clustering method for the detected motion segments also based on the self similarities which needs no assumption on the number of clusters. We show the applicability of our method on a wide variety of motion datasets, ranging from markered and markerless motion capture to accelerometer and surface electromyography (EMG) recordings.

Defining the segments based on self-similarity gives our method several advantages over previous segmentation methods. First, it allows us to distinguish not only the distinct action segments, but also the transition segments between actions. Explicit treatment of transitions has so far not been addressed in previous works on unsupervised segmentation [ZlTH08, ZlTH13, MC12]. However, it has a direct impact on synthesis methods, such as motion graphs [KGP02, MC12]; not handling the transitions forces synthesized sequences to include additional primitives coming from possibly unrelated transitions, although this is neither convenient nor intrinsically motivated.

The second advantage of our method is that resulting motion primitives are highly consistent, with similar start and end points from sequence to sequence. Previous approaches often yield primitives which are phase-shifted from one another [ZlTH13]. Any statistical model built on unaligned primitives will be noisier and less representative of the actual motion. We note that our motion primitives are not limited to being full motion cycles i.e. those starting and ending in the same body pose, but can also be either parts of a cycle or entirely non-cyclic in nature.

Our approach concentrates on data sequences representing biped as well as quadruped motion data which contain repetitive patterns in the sense that similar features reoccur in a cyclic sense or other repetitive fashion. Note that this explicitly includes primitives which are not full motion cycles but rather parts of cycles or even of entirely non-cyclic nature.

Our approach works well for human motion sequences which contain repetitive patterns in the sense that similar poses reoccur in a cyclic sense or other repetitive fashion. In our systematic examination of all data given in the CMU [CMU13] and HDM05 [MRC+07] data bases we observed that, by taking into account all poses of sequences of length more than 8 seconds which reoccur - after some intermission - within a time slot not wider than 10 seconds, we reach 81.89% of all motion sequences in these data base. The examples in the MSRC-12 Kinect Gesture Data Set [FMKN12] are gestures which are not as highly repetitive and also contain a lot more noise. Nevertheless, we found the approach also works for this type of data.

2.1 Related Work

Temporal segmentation is related to a number of different fields such as data mining [KCHP01, Fea06], audio and speech processing [ODK96, MGW09, MG12, PM14], and behavioural pattern recognition [XM07]. In statistical terms, the segmentation problem can be posed as a change point detection task [Fea06] which has been extended to a multi-dimensional setting [XM07] based on the established Bayesian techniques [WSJF09]. From a more general signal processing point of view, kernel-based methods for change-point analysis [HMB09, DDD05], Hidden Markov Models as means for training and recognition [ODK96], and audio thumbnailing combined with enhanced similarity matrices [MGW09, MG12], have been introduced. There have been a number of attempts to solve the nontrivial problem of automatic segmentation of human motion data, which we outline below.

Pose Clustering

One strategy for segmentation is based on clustering the poses present in a time series [BCvdPP08, LMGCG12, BWK+13]. Beaudoin et al. [BCvdPP08] proposed to extract motion motifs as building blocks of graphs. Gall et al. [LMGCG12] create temporally meaningful pose clusters associated with unlabeled actions. Bernard et al. [BWK+13] proposed an exploratory search and analysis system called *MotionExplorer* based on similarity features. Depending on the aggregation level, shorter or longer motion segments were found which may be associated with isolated human actions. However, Bernard's main focus was on a new visual representation of motion data rather than analysis tasks as discussed here.

Examplars and Template Models

An alternative approach for segmentation is to apply example segments or precomputed templates and match them to test sequences. For example, Lv et al. [LN06] uses a combination of HMMs and AdaBoost to learn discriminative feature templates from labeled segments to perform action recognition and segmentation. Müller et al. [MRC05, MR06, MBS09] developed a framework based on geometric features to learn templates for solving and accelerating solutions to matching problems such as annotation and retrieval [MR06, MBS09]. Adaptive segmentation [MRC05] is a fundamental result from using geometric feature vector sequences to compare motion capture data streams at the segment level rather than frame level. Another set of approaches [LS13, SZB15] present related ideas on learning intrinsic regularities for segmentation and demonstrate that motion capture data can be segmented by using only a limited set of example motions, even when belonging to different action types than that of the training data. Template approaches work well if the templates are available; our work is targeted at cases in which the exemplars or templates are not known beforehand.

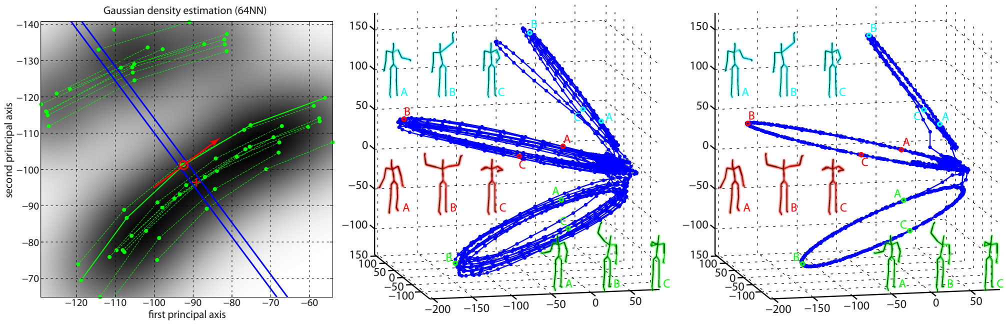
Motion Synthesis

Segmentation has also been addressed in conjunction with motion synthesis, such as motion concatenation [KGP02] and motion parameterization [KG04]. In motion concatenation, a *motion graph* is constructed from clips of motion capture data; new sequences are then sythesized by motion extraction on this graph. In motion parameterization, motion elements are retrieved from large datasets, based on similarity to a query motion, and then blended together according to user constraints. The novel distance relation used in this work [KG04] has become the standard for finding similar motion clips at interactive speeds.

Later works [HG07, SH07, MC12] combine both these ideas to accomplish motion synthesis techniques for high quality interactive applications. Min and Chai's Motion Graphs++ [MC12] is an advanced combination which effectively enables a variety of applications such as motion segmentation, recognition and online synthesis. All these approaches have a need for meaningful motion primitives that can be clustered to build statistical motion models or at least to allow for interpolation. Typically these are found by manual selection of some example primitives and a retrieval component to search for further exemplars in a database.

Unsupervised Motion Segmentation

The methods most similar to ours are those which segment motions in an unsupervised way. Partitioning motion sequences into behavior segments by a PCA-based method was proposed by Barbič et al. [BP$^+$04]. This segmentation focuses on detecting behavior segments and is similar in spirit to the first step of our approach for isolating distinct actions. In [BP$^+$04], the quality of local PCA models is tracked temporally; new activities are defined when the old PCA model cannot



(a)

(b)

Figure 2.2: (a) 2D example for **density estimation**: Original data points (red circle), 64 **nearest neighbors** (green dots connected in time), the **direction of movement** (red arrow), the 1D **subspace** used for optimization (blue lines) and the resulting position (red cross). (b) left-hand side: 3D **PCA projection** of the feature sets of a MoCap sequence (CMU 86_11). The three sections of loops correspond to repetitions of differing arm rotation movements as indicated by representative body poses. (b) right-hand side. **Bundled feature points** for the same motion capture sequence. Figure is best viewed in colour.

capture the data variance and a new PCA model is required. This approach is neither able to separate activities that fit into one local model, nor is able to detect individual representations.

The groundbreaking work of Zhou et al. [ZlTH08, ZlTH13] uses (hierarchically) aligned cluster analysis (H)ACA to temporally cluster poses into motion primitives which are then assigned to different action classes. These approaches use kernel based projections and a time alignment to compare motion primitives of varying length. An initial segmentation of uniform length is refined by the clustering approach though the final resulting segments do not vary much from the initial length. One of the major differences between the work of Zhou et al. and ours is that the transitions between distinct actions are not considered. Transition frames are assigned either to the previous or the following action segment, thereby reducing the consistency of the primitives.

Finally, Gong et al. [GMZ14] present an online approach based on *kernelized temporal cuts* which incorporate Hilbert space embedding of distributions when extending change-point detection methods. This work is not directly comparable to ours as it is an online approach.

2.2 Feature Bundling

Semantically or visually similar motions, even when represented in dedicated feature spaces, may still differ notably due to variations in the performance of each action or cycles within the same action. Inspired by the idea of edge bundling used in visualization [Hol06, BSL$^+$14], where similar edges of a graph are visualized together for a better overview, we propose a bundling of similar features. The goal of feature bundling is to topologically align disjoint motions; motions of same action class, but exhibiting differences in the feature space as a result of performance variations can be bundled together, while motions belonging to different classes or showing entirely different styles should be kept apart. Note that even though we name our technique feature bundling, we have completely different objectives as well as different methodological approaches than *bundle adjustments* used in 3D reconstruction algorithms.

In our bundling technique we use a density estimation based optimization method which adjusts each point orthogonally to the direction of its trajectory in the feature space, effectively projecting the features onto a smoother manifold (see Figure 2.2 for an overview). Vejdemo-Johansson et al. [VJPSK14] consider a related idea by computing a typical motion cycle of a set of similar periodic motions.

For each frame i of an input sequence of length N, a feature vector F_i where $i \in [1 \dots N]$ is computed; the specific feature depends on the sensor modality (more details in Section 2.6). The goal is to compute a new feature vector $\hat{F}_i$ representative of F_i, but closer to other features at the same stage of the same motion cycle. Specifically,

1. For each F_i, find k **nearest neighbors** within all other features $F_j,\ j \in [1 \dots N] \setminus i$.

2. Compute **direction of movement** d_i for F_i and build a $D{-}1$ dimensional **subspace** $\mathcal{S}_i$ orthogonal to d_i.

3. **Optimize** $\hat{F}_i$ using a **kernel density estimate** of F_i in $\mathcal{S}_i$ based on the k nearest neighbors.

4. Backproject $\hat{F}_i$ to the original feature space for the final representation.

For the **k-nn search** we use a kd-tree as per [KTWZ10] to find similar frames within the motion sequence. We do not use a fixed search radius since the distance between sample points may vary to a great extent. Instead, a fixed number of k nearest neighbors ensures that the model reflects the local density of samples.

The **direction of movement** d_i for frame i is given by the numerically centered five-point derivative at this frame. Constructing an orthogonal **subspace** prevents the overall data structure from collapsing. The basis of this subspace is computed via QR decomposition of a $D{\times}D$ matrix, where the first column vector is set to the direction of movement, while all other entries are filled with random numbers.

The local kernel **density estimation** is based on the k nearest neighbors and characterizes the data distribution of F_i with kernel function K_H, i.e. a symmetric multivariate density with bandwidth matrix H:

$$K_{H^i}(x) = |H^i|^{-\frac{1}{2}} K({H^i}^{-\frac{1}{2}} x). \tag{2.1}$$

Since our data is assumed to be Gaussian, we use a general approximation of the bandwidth matrix which minimizes the MISE (refer to Scott [Sco92], Chapter 6, Scott's rule) by

$$H_{jj} = \sigma_j k^{\frac{-1}{d+4}}, \tag{2.2}$$

where $j = 1, \dots, d$ and σ_j is the standard deviation of the jth variate. We use the multivariate Gaussian kernel:

$$K(x, \mu, \sigma) = e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}, \tag{2.3}$$

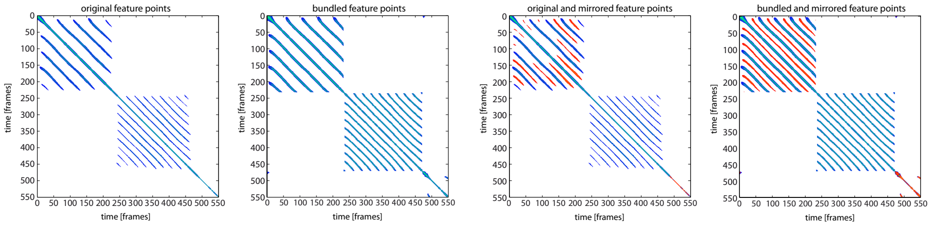
with $x = (x_1, \dots, x_d)$, $\mu = (\mu_1, \dots, \mu_d)$ the vector of empirical mean values, and Σ the sample covariance matrix.

Note that the kernel density estimation is compatible with a preceding dimensionality reduction step. In fact, it is possible to reliably estimate the kernel density function without increasing the sample size k [Sco92]. Therefore, we apply a PCA to each set of samples in advance to reduce the dimension while keeping 97.5% of the sample set's variance. In principle, any of the commonly used dimensionality reduction techniques is applicable; we prefer PCA for its simplicity and speed, though other approaches have been explored extensively in the past [JM04, BP$^+$04, GFA07].

The **optimization** searches for the new position $\hat{F}_i$ and is posed as an energy minimization problem. The offset O_i from F_i is optimized with respect to the density estimation, resulting in the final feature position $\hat{F}_i$:

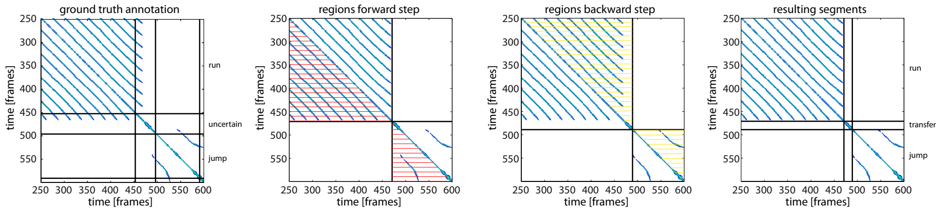
$$\hat{F}_i = \underset{O_i}{\operatorname{argmin}}\, K_{H^i}(F + O_i). \tag{2.4}$$

An example of the feature bundling is given in Figure 2.2 a). A 3D projection of input and output feature points of a motion sequence are given in Figure 2.2 b) and c), respectively. A comparison between a sparse self similarity matrix (SSSM) based on the original and the bundled features is given in Figure 2.3 a) and b) respectively.



(a) (b) (c) (d)

Figure 2.3: Comparison of SSSMs of two different situations: original and bundled features. Part (a) shows the SSSM of original features as they were extracted from the input motion. Part (b) shows the same SSSMs bur of the bundled features. Images (c) and (d) include the mirrored features also (highlighted red). Note that the SSSM based on the original mirrored features, the structure in the SSSM (c) is less consistent than in SSSM of the bundled representation (d).



(a) (b) (c) (d)

Figure 2.4: Region growing for activity separation: (a) Sparse self similarity matrix with ground truth annotations; the label 'uncertain' indicates an area of inconclusive user annotations. (b) Same matrix without main diagonal; also results for forward step of region growing. (c) Results backward step of region growing. (d) Final outcome of segmentation. Note that in the actual computation of the region growing steps (b) and (c) the main diagonal is removed. It is displayed in this representation for visualization purposes though.

2.3 Segmentation Technique

The input to our method is a multi-dimensional time series acquired by recording a motion trial. First, the local neighbors of each frame in the motion sequence are found as a preliminary step. The sequence is then partitioned into distinct temporally coherent action segments; a subsequent step investigates the structure of these actions to find recurring patterns, i.e. shorter *motion primitives* potentially enclosed as part of the action.

2.3.1 Local Neighbors

A motion sequence M is given as a collection of m subsequent (possibly high-dimensional) data points $p_1, \ldots, p_m$, each of which is represented by a feature vector $F = (f_1, \ldots, f_N)$ of dimension N capturing the information over time. The features are modality specific and stacked in the time dimension, yielding a vector representation $[p_{i-\phi_1}, p_i, p_{i+\phi_2}]'$ of features over a window $w = [i-\phi_1, i, i+\phi_2]$ in a higher dimension.

Within the feature space, we define a search radius r to search for the neighbors. Given no prior knowledge of the input data, we introduce a **generalized search radius** R which is independent of the feature set time window size w and input data dimensionality N. R is defined as the search radius for a window size of $|w| = 1$ and dimensionality of $N = 1$; the search radius is then rescaled as $r = R\sqrt{|w| \cdot N}$.

We construct a **kd-tree** from all feature points F_i in the input stream and then search for the points located within the radius r based on the Euclidean distance d_{ij} between the features F_i of p_i and F_j of p_j. As a result of this search, we obtain a set S_i of neighbors for each data point p_i. The neighbors are specified as pairs $(j \in [1 : M], d_{ij})$ of an index j to a frame in the input motion and the distance d_{ij} between the query point and neighbor j.

Connection between Neighbors and SSSM

The sets of neighbors of a motion trial can be converted into a **sparse self similarity matrix (SSSM)**. Self-similarity matrices are commonly used in human motion analysis for a range of tasks ranging from retrieval [KG04] to multi-view action recognition [JDLP11]. A SSSM, as shown in Figure 2.3 a), is generated by initializing an empty $M \times M$ matrix $\mathcal{M}$. For each frame $i \in [1 : M]$ we set the entries $\mathcal{M}_{i,j} \forall k \in S_i$ to the values of d_{ij} which are stored in S_i. An illustration of this connection is also given in Figure 2.5 a) and b). Note that we use the matrices only for visualization in this work; for efficiency purposes, computations are performed directly on the sets of neighbors or derived data structures.

2.3.2 Segmentation into Distinct Activities

Figure 2.4 shows an example SSSM with two cyclic activities, running and jumping, separated by an 'uncertain' period of inconclusive user annotations. Note that the cyclic activities are characterized by structured diagonal blocks. We separate the activities by searching for these characteristic blocks, using **region growing** to determine the blocks' borders.

A connected region starts as a seed in the upper left corner of the neighborhood representation matrix $\mathcal{M}_{1,1}$. Contents of the lower triangular matrix below the main diagonal are then probed using scan lines. The triangular region is gradually extended to adjacent rows as long as the number of nearest neighbors in the updated region increases. If no new neighbors are found between frame i and $i + w$ in the larger region, the current region is considered complete. The parameter w is introduced to handle noisy data and is set to $w = 8$ in all our experiments. With such a stop criterion, neighbors from the main diagonal of the SSSM cannot be considered — otherwise these elements would also be counted in the region-growing and result in a large region covering the entire SSSM. We remove all entries of the main diagonal in the proximity corresponding to one second, based on observations that cyclic behavior in motion data does not occur at higher speeds. A new region is then started from the upper left entry of the remaining matrix $\mathcal{M}_{i,i}$, with scan lines probing the content of the lower triangular matrix between $\mathcal{M}_{i,i}$, $\mathcal{M}_{i,j}$ and $\mathcal{M}_{j,j}$, where j is the current frame.

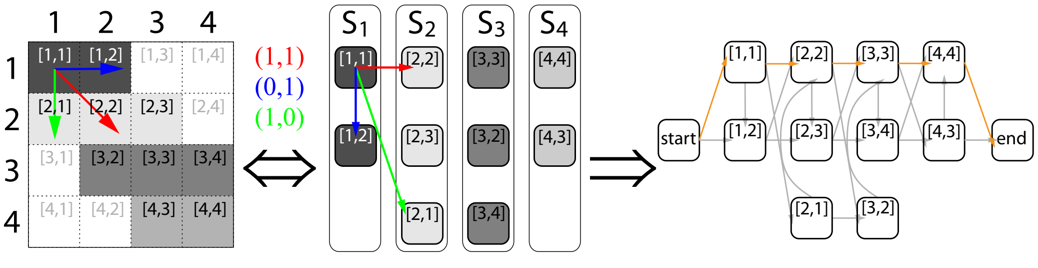
The region growing is performed once as a **forward step**, seeding the first region at frame $\mathcal{M}_{1,1}$ of the matrix, to identify the end of repetitive patterns (see Fig. 2.4b) and once as a **backward step**, seeding at the last frame $\mathcal{M}_{n,n}$ of the input sequence first to identify the start of the actions (Fig. 2.4c). The lower right corners of the forward region growing correspond to end frames of an action, while the upper left corners of the backwards step correspond to the start frames of an action. Areas in between are considered transitions between the repetitive parts.

For efficiency, we work on the sets S_i of neighbors, counting the number of entries in the neighborhoods between the seed frame and the current scan line index. Because we work with a symmetric matrix, this is equivalent to scanning triangular parts of the sparse similarity matrix.

Compared to the region growing approach of Vögele et al. [VKK14], where the neighbors were counted in a quadratic region, the method presented here is more computationally efficient. For each scan line, only one set of nearest neighbours needs to be considered, while for the quadratic region in [VKK14], all preceding neighbour sets are reconsidered for each growing step. The runtime complexity is therefore reduced from $O\left(k\frac{n(n-1)}{2}\right)$ to $O(kn)$ in our approach, where k is the maximum number of nearest neighbors and n is the number of frames of the motion trial—in the worst case the first region grows over the whole SSSM.

2.3.3 Subdividing Actions into Motion Primitives

Once the input sequence is segmented into distinct actions, we can search for motion primitives within each action. Consider a single action, such as walking in



(a) (b) (c)

Figure 2.5: Toy example illustrating the relationship between SSSM and neighborhood graph G_M. (a): SSSM of four consecutive points with allowed steps indicated by arrows. Red arrow: step $(1,1)$, blue arrow: step $(0,1)$, green arrow: step $(1,0)$. (b): Neighborhoods of each of the four points, e. g. S_i is the neighborhood of the point i. (c): Resulting neighborhood graph.

Figure 2.6; we want to find the reoccurring units, i.e. steps of which the activity consists. Such units are responsible for the minor diagonals in the SSSM of the specific activity, with start and end frames of each unit corresponding to the start and end position of a diagonal. To find such frames we use the minimal cost warping path in the SSSM. The start and end positions of this path are considered as representative positions for the diagonal. Rather than searching for the diagonals' starts and ends in the SSSM, we use an alternative neighborhood graph representation and simplify the problem to finding the shortest paths. We note that if no reoccuring primitives are found, the action segment is considered itself a single primitive. In the following, we describe an alternative way of describing the sparse self similarity matrix. With the help of this representation we can give more insight on how to find the minimal cost warping paths.

Alternative Representation by a Neighborhood Graph

All of the neighbors p_j within one specific activity stored in the set of neighborhoods $S = \{S_i,\ i \in 1,\dots,n\}$ can be considered as nodes of a graph G_{act}. The criteria for connecting the nodes in this graph is based on accessibility between the points and can be characterized by the concept of dynamic time warping. Therefore, we will briefly recall this concept to clarify how to access one data point in the neighborhood graph from another. Consider two points $p_j \in S_i$ and $p_{j^*} \in S_{i^*}$. A valid time warping step to access the point p_{j^*} from p_j is defined as a pair $(a,b) \in \{(1,1),(0,1),(1,0)\}$ such that $p_{j^*} = p_{j+b}$ and $S_{i^*} = S_{i+a}$. In particular, the point p_{j^*} is always an entry further below in the neighborhood list S_{i^*} than p_j is in the list S_i, while S_{i^*} is either identical to S_i or lies to its right hand side $(S_{i^*} = S_{i+1})$. Figure 2.5 shows a toy example to illustrate a possible scenario.

Sparse self-similarity matrices are never actually **symmetrical** because the data point similarities which constitute the matrix block representation are restricted to a maximum number of admissible neighbors. Therefore it is possible, a neighbor p_l is taken into account for point p_u, which, in turn, does not appear to be in the neighborhood of p_l because the number k cuts the list of p_l's neighbors before p_u is found. Nevertheless, those parts of the matrix where a minor diagonal has a fairly symmetrical match 'mirrored' by the main diagonal arise from more perceptually coherent matches in the nearest neighbor search. In short, preserving **symmetry constraints** up to a point leads to more stable results.

An important property of the graph G_act which we exploit lies in the following observation: each diagonal of the matrix $\mathcal{M}_\text{act}$ reflecting local similarities is one **connected component** of G_act. For the next steps, it is useful to work only with neighbors belonging to the same connected component cc for a given frame, with the resulting restricted graph denoted as G_{cc} (Refer to Figure 2.6 for a visualization of connected components). Note that there is never a need for an exhaustive search since every minor diagonal of the sparse self similarity matrix corresponds to a connect component in G_act.

Computation of Warping Paths

Dynamic time warping, to calculate an optimal match between two given time series A and B with certain restrictions, creates a path between these sequences. The sequences are matched non-linearly in the time dimension to optimize for a similarity measure. Technically, a warping path $\mathcal{P}_{A,B}$ of length λ between two such sequences is given as a pair of vectors (v_A, v_B) where $v_A = (a_1, \dots, a_\lambda)$ with $a_i \in A$ meeting constraints such as $a_i \leq \nu a_{i+1}$ for all indices and $v_B = (b_1, \dots, b_\lambda)$ with $b_i \in B$ meeting $b_i \leq \nu b_{i+1}$. In our experiments, we use $\nu = 2$. The constraints on v_A and v_B could be seen as upper and lower limit of the paths **slope**.

The sets of neighbors S_i are suitable to replace conventional dynamic time warping based on the neighborhood graph described above. (for more details on how this can be used instead of time-warping, see Krüger et al. [KTWZ10] using the example of motion capture time series).

Since each diagonal in the SSSM translates to one connected component in the graph, searching for an optimal warping path reduces to finding a shortest path through the connected component G_{cc}. To this end, we add one additional start and one end node to the graph. The start node connects to all nodes corresponding to the first set of neighbors in the component, while the end node is connected to all nodes that correspond to the last set of neighbors. Now, the warping path can be found efficiently by searching the shortest path from the start to the end node. The costs of a path is the accumulated distance of the included neighbors.

We further limit the set of warping paths per activity based on their length and slope. First, paths covering less than five frames of the motion trial are discarded; such paths are found for very short but similar segments existing between longer primitives. Although these segments may be semantically meaningful, we ignore these to prevent extremely short primitives. Secondly, paths with average gradients less than 0.5 and larger than 2 are also discarded. Such paths represent mapping between motions whose speeds vary by a factor greater than two. We want to avoid such cases, e.g. when a longer standing sequence is mapped to a few poses in the middle of a walking cycle.

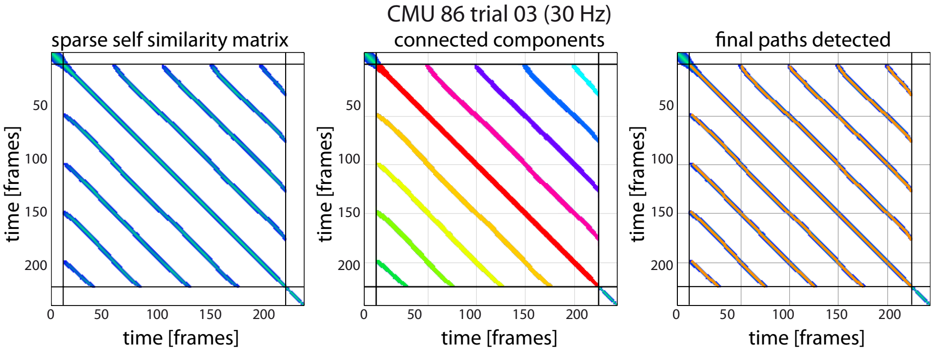
For each valid warping path $\mathcal{P}_i$ we have a pair (a_i, b_i) representing the starting position within the SSSM. These positions correspond to the bordering frames between motion primitives. We only check if any candidates are closer than 5 frames. If this is the case we only consider the one where the corresponding warping path had smaller costs.

In the given example in (Figure 2.6) the geometric shape of the minor diagonals in the matrix encode important information on the relationship between the match candidates they depict. For instance, two parallel stretches indicate there will be an warping path between the associated subsequences with a slope near to 1, i. e. the distortion of the time dimension brought about by matching them will be small, in effect, pointing to a near-linear motion match.

There are two objectives of **path projection coverage**. Valid path candidates should not be shorter than 5 frames to avoid minor structures that actually represent noise. Also, when projected to either the rows $i(m_1), \dots, i(m_l)$ or the columns $j(m_1), \dots, j(m_l)$ in the matrix index notation, the length of the respective projection of the path should not drop below the path length l by more than 50%. Note that this corresponds to admitting only a mean slope of between $\frac{1}{2}$ and 2. Detecting the motion primitives can also be combined with the symmetry detection step.

Complexity Analysis

The critical computation step for detecting primitives is building the graph representation from the sets of nearest neighbors. Creating this graph requires checking all possible connections of each neighborhood entry in $\mathcal{N}$ to other entries by a number of s possible steps. For each of the neighborhoods of each activity there is a maximum of k entries. Since the number of edges is limited by $O(ksn)$, the search for connected components is limited to the same complexity, the overall run time complexity is also $O(ksn)$.



(a) (b) (c)

Figure 2.6: Illustration of connected components in G_{act}. (a) SSSM corresponding to walking. (b) Same matrix with its connected components color coded. (c) Optimal warping paths highlighted by orange lines.

2.4 Symmetry of Motion Data

We aim to exploit specific types of intrinsic symmetry information found in motion data. In general, it makes sense to distinguish three situations:

(S1) Synchronous activity in two laterally mirrored body parts in the sense that the features describing such body parts are mirror-symmetric. One example of a synchronous activity is a jumping jack.

(S2) Phase-shifted synchronous activity between two laterally mirrored body parts in the sense that the characterizing features have a phase-shifted symmetry. One example of a phase shifted symmetrical activity is walking.

(S3) Asynchronous or asymmetric activity between body parts.

In the following, we will be particularly interested in the phase-shifted symmetry (S2).

2.4.1 Mirrored Motions

Motion data contains several intrinsic symmetries which can be exploited during analysis. We focus on mirrored motions and begin by defining the plane of symmetry. Let $X = \{x_1, \ldots, x_J\} \in \mathbb{R}^{3\times J}$ be a geometric model of a moving subject, i.e. an ordered set of joints which are connected by edges according to kinematic properties of the model. A motion of X is a multi-dimensional trajectory $\mathbb{X}$ of X over time. Let P_X

be a plane spanned by two perpendicular vectors which connect joints or linear combinations of joints.

For human models, the plane of symmetry is the saggital plane, i. e. the vertical plane which passes from anterior to posterior, dividing the body into left and right. This plane is spanned by a vector oriented vertically, i. e. connecting the center of body mass with the top of the head, and the vector indicating the exact front, i.e. the normal in the sternum. A motion is symmetric with respect to this mirror plane if, for a set of descriptive features $\mathcal{F}$, at least one pair of features $f_i, f_j \in \mathcal{F}$ can be switched without imposing a (significant) change on $\mathbb{X}$.

The concept of the mirror plane can be transferred from humans to other models; all vertebrates are bilaterally symmetrical with two pairs of appendages such as limbs, fins or wings and the saggital plane is also a mirror plane. Many invertebrates also show this feature. The mirror plane for the symmetry of vertebrates can also be assumed the saggital plane. In order to describe the mirror plane for the symmetry of vertebrates, one spanning vector may be assumed to be in line with the vertebral column of the model and and the other can be defined the normal in the model's sternum.

2.4.2 Detecting Mirrored Motions

As a natural consequence of the saggital plane being the mirror plane, the prominent symmetric features f_i, f_j are the ones describing movements of extremities. For example, locomotion activities such as walking and running have a phase-shifted symmetry. Primitives of walking and running actions, according to Section are double steps which may start with either foot. However, if one were to mirror the action along the saggital plane and merge the cuts separating the primitives from the original and the mirrored action, then the separation will extend to finding single steps rather than the double steps.

The comparison of original motions and its counterparts mirrored at the saggital plane leads to different variations of symmetry. The symmetry of an action segment X_A, based on its mirrored version X'_A mirrored at the saggital plane, may be characterized as follows:

1. X_A is highly symmetric if its primitive segmentation is exactly the same as that of X'_A in the sense of item S1 on the list 2.4.

2. X_A is exclusively phase-shift symmetric if its primitive segmentation has no cuts in common with X'_A.

3. X_A is asymmetric if the primitive segmentation of X'_A returns no cuts at all.

In particular, their primitives without including the findings of inspecting the mirrored version of them, are double steps which may start with either foot. When merging the cuts separating primitives in both and $X_{A_{mirrored}}$, the separation extends to finding single steps rather than double steps. Apart from a refined segmentation, this also results in insight on the presence of each type of symmetries in the list (S1)-(S3).

2.4.3 Symmetry Detection on Mirrored Motions

Identifying occurrences of symmetry by investigation of motion representations can help distinguish changes in movements which indicate motion primitives. This concept will be exploited in Section 2.3.3. In order to do so, the concept of mirrored motions will be used.

The idea is detect the presence and whereabouts of motion primitives in an activity X_A. Independently of that, we also detect motion primitives in a mirrored version of the motion. When both findings are merged, this helps refine the separation of primitives.

Naturally, a mixture of two or more situations is possible when an action contains different types of motion primitives (see Appendix 2.4.4 for an overview). Therefore, it makes sense to treat each square region of the SSSM (identified by the activity separation) representing individual motion activities separately, as there may be some activities which have symmetric counterparts and some which do not.

Note that it is possible to define a Boolean operation on the similarity matrices associated with motions and their mirrored versions. This can be defined as the operation which joins entries of both matrices. Technically, this means detecting paths in the junction of the original similarity matrix $\mathcal{S}(X)$ with the similarity matrix $\mathcal{S}(X_{mirrored})$ of the mirrored motion. Though this eventually amounts to the same thing, we treat the original input motion and its mirrored version independently for efficiency reasons as was described above.

One example of exploiting phase-shifted symmetry is distinguishing single steps in walking.

2.4.4 Symmetry Types

In the following, we want to revisit the different types of symmetry defined in Section 2.4. Schematic representations of self-similarity matrices for each of the possible cases, discussed in Section 2.4 are given in Table 2.2. For symmetrical motions, there is no difference in the diagonal structure in the SSSM from the original and mirrored features. This leads to no additional cuts when symmetry is exploited for the segmentation. Phase-shifted symmetry can occur with and without speed variation. Without any speed variations, a more diverse structure in the self-similarity matrices occurs in both the original and mirrored features.

For the original features, more block-shaped parts may appear on each diagonal, while in the mirrored setting, all diagonals aggregate to a more curvy pattern. In the asynchronous case, the mirrored self-similarity matrix shows no diagonal structures at all, adding to no additional cuts. When symmetries are mixed, the SSSMs are characterized locally according to the corresponding symmetry type.

2.5 Clustering of Motion Primitives

By clustering the motion primitives, it is possible to find the frequency with which the same primitives occur and also have an indication of the semantic and temporal relationships between different primitives. This is of great interest for motion synthesis, using motion graphs [MC12, HTK$^+$15], or for motion analysis, in terms of action recognition. We propose a unique clustering algorithm in which we do not have to provide the number of clusters in advance.

A cluster graph $\mathbf{G}_{\mathbb{M}}$ is used to store the similarity information between the motion primitives. In this cluster graph, each primitive is represented as a node. Consider now a sparse self similarity matrix associated with a motion M; the motion primitives m_q are represented by squares on the main diagonal. The goal is now to search for valid warping paths between each pair of motion primitives m_i and m_j. To this end, we can build a neighborhood graph (see Section 2.3.3) including the neighbors in the rectangle region that is spanned when comparing m_i and m_j. The shortest path from the top entries to the bottom entries in this submatrix is found if it exists. If this shortest path satisfies a minimum length requirement and has a slope inside the range of valid slopes (see Section 2.3.3) we add an edge between the corresponding nodes in $\mathbf{G}_{\mathbb{M}}$. Note that this this a different neighborhood graph where each node is not now a frame but a primitive. After the algorithm has gathered all similarity information, a search for the strongly connected components is performed on $\mathbf{G}_{\mathbb{M}}$. Each strongly connected component represents a cluster of motion primitives.

The algorithm presented above is a modified version of the algorithm of Vögele et al. [VKK14]. Both approaches perform equally well for clustering though the current approach is more efficient, since only small neighborhood graphs between each pair of primitives are constructed, while in the previous work one large neighborhood graph was constructed over all neighbors of the trial and then was cut into smaller parts when needed.

Table 2.1: Schematic self similarity matrices for various types of symmetries.

Original

	A	B	C	D	A	B	C	D
A	0	∞	∞	∞	0	∞	∞	∞
B	∞	0	∞	∞	∞	0	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
D	∞	∞	∞	0	∞	∞	∞	0
A	0	∞	∞	∞	0	∞	∞	∞
B	∞	0	∞	∞	∞	0	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
D	∞	∞	∞	0	∞	∞	∞	0

Mirrored

	A	B	C	D	A	B	C	D
C	∞	∞	0	∞	∞	∞	0	∞
D	∞	∞	∞	0	∞	∞	∞	0
A	0	∞	∞	∞	0	∞	∞	∞
B	∞	0	∞	∞	∞	0	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
D	∞	∞	∞	0	∞	∞	∞	0
A	0	∞	∞	∞	0	∞	∞	∞
B	∞	0	∞	∞	∞	0	∞	∞

Original

	A	A	B	B	C	D	A	A	B	B	C	D
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0

Mirrored

	A	A	B	B	C	D	A	A	B	B	C	D
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
C	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0
D	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	0
A	0	0	∞	∞	∞	∞	0	0	∞	∞	∞	∞
B	∞	∞	0	0	∞	∞	∞	∞	0	0	∞	∞

Two cases of phase-shifted symmetries. The first matrix shows that the original and mirrored features are disjoint, i.e. the diagonal structures in the SSSM are at different locations. The second is a similar situation where the original motion shows some speed variation, leading to a more diverse structure in both matrices. The mirrored features match the originals in some sense but the variation is seen in the slopes of the diagonals.

Original

	E	F	G	H	E	F	G	H
E	0	∞	∞	∞	0	∞	∞	∞
F	∞	0	∞	∞	∞	0	∞	∞
G	∞	∞	0	∞	∞	∞	0	∞
H	∞	∞	∞	0	∞	∞	∞	0
E	0	∞	∞	∞	0	∞	∞	∞
F	∞	0	∞	∞	∞	0	∞	∞
G	∞	∞	0	∞	∞	∞	0	∞
H	∞	∞	∞	0	∞	∞	∞	0

Mirrored

	E	F	G	H	E	F	G	H
E	0	∞	∞	∞	0	∞	∞	∞
F	∞	0	∞	∞	∞	0	∞	∞
G	∞	∞	0	∞	∞	∞	0	∞
H	∞	∞	∞	0	∞	∞	∞	0
E	0	∞	∞	∞	0	∞	∞	∞
F	∞	0	∞	∞	∞	0	∞	∞
G	∞	∞	0	∞	∞	∞	0	∞
H	∞	∞	∞	0	∞	∞	∞	0

Example of a symmetric motion: The diagonal structures in the SSSM are at exactly the same locations, leading to no additional cuts.

2.6 Experiments

We now report on a series on experiments to show the effectiveness of our approach. First, we compare our results with previous methods on a set of motion

Table 2.2: Schematic self similarity matrices for various types of symmetries.

Original

	I	J	K	L	I	J	K	L
I	0	∞	∞	∞	0	∞	∞	∞
J	∞	0	∞	∞	∞	0	∞	∞
K	∞	∞	0	∞	∞	∞	0	∞
L	∞	∞	∞	0	∞	∞	∞	0
I	0	∞	∞	∞	0	∞	∞	∞
J	∞	0	∞	∞	∞	0	∞	∞
K	∞	∞	0	∞	∞	∞	0	∞
L	∞	∞	∞	0	∞	∞	∞	0

Mirrored

	I	J	K	L	I	J	K	L
M	∞	∞	∞	∞	∞	∞	∞	∞
N	∞	∞	∞	∞	∞	∞	∞	∞
O	∞	∞	∞	∞	∞	∞	∞	∞
P	∞	∞	∞	∞	∞	∞	∞	∞
M	∞	∞	∞	∞	∞	∞	∞	∞
N	∞	∞	∞	∞	∞	∞	∞	∞
O	∞	∞	∞	∞	∞	∞	∞	∞
P	∞	∞	∞	∞	∞	∞	∞	∞

Example of an asynchronous motion: The diagonal structures in the original SSSM have no matches in the mirrored version, again leading to no additional cuts.

Original

	A	J	C	L	A	J	C	L
A	0	∞	∞	∞	0	∞	∞	∞
J	∞	0	∞	∞	∞	0	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
L	∞	∞	∞	0	∞	∞	∞	0
A	0	∞	∞	∞	0	∞	∞	∞
J	∞	0	∞	∞	∞	0	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
L	∞	∞	∞	0	∞	∞	∞	0

Mirrored

	A	J	C	L	A	J	C	L
C	∞	∞	0	∞	∞	∞	0	∞
N	∞	∞	∞	∞	∞	∞	∞	∞
A	0	∞	∞	∞	0	∞	∞	∞
P	∞	∞	∞	∞	∞	∞	∞	∞
C	∞	∞	0	∞	∞	∞	0	∞
N	∞	∞	∞	∞	∞	∞	∞	∞
A	0	∞	∞	∞	0	∞	∞	∞
P	∞	∞	∞	∞	∞	∞	∞	∞

Example of a motion with different symmetries, with local matches for the original diagonals in the mirrored version. This means that the motion has phase-shifted symmetrical phases interrupted by other phases which are asynchronous.

capture data. Second, we show that meaningful results are obtained when using different sensor modalities such as accelerometers and EMG sensors. Finally, we apply our approach to Kinect skeleton data and show that our motion primitives are meaningful and consistent with of human-annotated key frames.

2.6.1 Segmenting Markered Motion Capture Data

We apply our segmentation method to the the motion sequences of subject 86 of the CMU database [CMU13], as was done by Zhou et al [ZlTH08, ZlTH13]. In their works, Zhou et al. compare the results of their segmentation with the results of the PCA-based method of Barbič et al. [BP$^+$04]. The PCA-based method is designed to separate activities. This naturally results in a coarser representation of the motion than the segmentation of Zhou et al. which separates motion primitives and clusters those into activity groups. The activity clusters are observed to coincide with the segmentation results of Barbič et al. We compare our segmentation results to the methods of Zhou et al., as well as to our previous work [VKK14], thus there is an implicit comparison to the PCA-based method as well. The details can be found in Figure 2.7. We show a number of improvements in comparison to [BP$^+$04, ZlTH13], the most notable being the ability to segment fine-structured motion primitives in nearly all cases. First, we are able to distinguish different styles of executing a task. For example, in the case of wiping a window/black board (CMU, trial 12 of subject 86), Zhou et al. [ZlTH13] group all primitives together as the same type (dark gray blocks), while we are able to distinguish between back and forth wiping motions versus circular wiping movements (various shades of gray).

Secondly, we are able to distinguish symmetric movements and separate primitives accordingly. Examples include rotation of the body in trial 7 (variations of blue and green blocks), dribbling the ball (right vs. left hand) in trial 14 (dark green, light green and red, orange blocks). Other approaches are unable to distinguish between a step forward with the left versus right foot, nor handling of the ball with the left versus right hand.

In this case though, left and right was clearly distinguished in the segmentation step and by the clustering (two different shades of green). This phenomenon can be observed for several other instances of walking, jumping and other symmetrical actions.

Note that the underlying motions are symmetrical in the sense of item 2 on the list in 2.4.3.

Accuracy Comparison

Our method produces the same action classes as [VKK14] and nearly the same classes as [ZlTH08, ZlTH13]. We use the same methods as [VKK14] to evaluate the segmentation accuracy on a frame-wise level, using both a strict and a more tolerant evaluation, and present the results in Figure 2.8. For a motion primitive s, the strict method checks whether all of s's frames belong to the same action class as the other primitives found from the same segment; the tolerant method eases the constraint to both the same action class as well as transition/uncertainty segments. Due to the finer division of primitives found by our method, there are different clusters representing the same motion. For example, walking consist of alternating left and right steps. For consistent evaluation with previous methods, we have assigned such symmetrical counterparts to the same class, i.e. the 'left step' cluster and the 'right step' cluster are both assigned to the walk action. We achieve significantly higher accuracy values for both types of evaluation, with an average of 90% for our method, 88% for the method of Vögele et al., and 79% for (H)ACA for the strict evaluation and 99% in comparison to 97% for Vögele et al., 92% for HACA and 91% for ACA for the tolerant evaluation. Vögele et al. discuss the application of the segmentation technique to the label transfer problem. Based on these values we anticipate that applying feature bundling would yield similar results.

Intra-Cluster Variance for Motion Primitives

Dynamic time warping is an established distance measure for temporal sequences and accumulates the local (frame-wise) distances from one segment to the warped version of the other. For a given cluster C a cumulative distance measure D, tallied over all pairs of segments s_i and s_j contained in the cluster can be defined as:

$$D = \sum_{i=1, j \neq i}^{\|C\|} \left(\frac{\mathrm{DTW}_\alpha(s_i, s_j)}{\|s_i\|} \right) \tag{2.5}$$

where $\|s_i\|$ is the length of segment s_i and DTW_α is the DTW distance of point clouds as defined by Kovar et al. [KGP02]. Note that D is particularly sensitive to outliers and will detect scattered or inconsistent clusters.

By making finer distinctions between motion primitives, we achieve lower intra-cluster variances for the clusters. Figures 2.9 a) and b) show variances of each motion class. Figure 2.9 a) compares the clustered results by our method, the method of Vögele et al., the HACA [ZlTH13] and the ACA [ZlTH08] method. The respective means are given by the blue color bars for our method, red for Vögele et al., (dark) light green color bars for (H)ACA. The variances are shown as gray glyphs. Figure 2.9 b) shows the same variance values for the case where mirrored features are included. The classes show similarly low variance. The given example is one case where additional motion classes are introduced by exploiting symmetry of motion: there are two classes of ‘steps’ and also two classes of ‘kicks’. The distinctions are caused by different types of symmetry. While walking is phase-shifted, kicking included one part which was symmetrical (more static) and one which was asynchronous (where the leg was up).

Our clusters group together a variety of motion primitives without transitions and primitives from other actions. In particular, our primitives reflect exactly the number of repetitions within actions. For illustration, there are five repetitions of ’rotate arms’ in one sequence (see Figure 2.7, Subject 86 trial 03, frames 1600-1800) and we segment this into exactly five primitives. Zhou et al.’s methods [ZlTH08, ZlTH13] does not account for the inherent repetition and segments the action into three primitives, thereby yielding much higher DTW distances between these primitives.

Over all clusters of all trials from actor 86 we obtain a average cluster variance of 1.18 (min: 0.52, max: 2.66, std: 0.42) for mirrored and bundled features, 1.17 (min: 0.58, max: 2.58, std: 0.37) for bundled features, 1.69 (min: 0.69, max:

3.45, std: 0.54) for the original features. Compared to HACA 2.51 (min: 0.48, max: 8.78, std: 1.53) and ACA: 2.56 (min: 0.86, max: 9.42, std: 1.50).

Parameter Evaluation

The most important parameters for our segmentation method are the search radius (Section 2.3.2) and the temporal window for feature stacking (Section 2.3.1). We found that the method is insensitive to either parameter and show the segmentation accuracy in Figure 2.10 for various parameter settings for the strict and tolerant evaluations for both original and bundled features. All plots show that the accuracies are high for nearly all possible combinations of parameters.

Our region growing in the activity separation step stops if no new neighbours were found in a window of w frames. We tested our approach with various window sizes and computed the strict accuracy measure for evaluation. Figure 2.11 a) shows the accuracy results. The method is very robust against this parameter, the accuracy only drops when the window size is chosen extremely large (128 frames).

For segmenting the motion primitives, there are the additional parameters of the allowed minimal and maximal value of the warping path slope ν (Section 2.3.3). Following conventions in the literature [KG04, KTWZ10] we set the slope to be within the window $\frac{1}{\nu}$ and ν. In all our experiments we set this parameter to be $\nu = 2$. We evaluated this parameter by computing the average number of motion primitives found per activity on the motion capture dataset. The results are shown in Figure 2.11 b). If ν is smaller than 2 the number of primitives drastically decreases, while larger values for k don't increase the number of motion primitives. We choose the value of two to permit extreme temporal deformations between motion segments. Although, our experiments indicate that such warps did not occur in the CMU dataset.

Details on Timings

Figure 2.12 shows a timing breakdown for the CMU examples. The first segmentation step (part c)) including feature bundling (part a) and knn search (part b) activity detection are, in practice, approximately linear in the number of frames (Figure 2.12 a) - c)). In theory, the worst case complexity for region growing is $O(kn)$, i.e. when the first region grows from the first to the last frame of the input motion sequence. This case was not observed in practice. The shortest path searches for primitive detection (part d)) is quadratic with respect to the number of frames per activity. Finally, the clustering step (part e) also computes in linear time (in the number of motion primitives).

On an Intel Core i7 4930K at 3.40GHz we were able to segment and cluster each motion sequence (up to 3000 frames in length) in less than 15 seconds using our single threaded Matlab implementation.

2.6.2 Combined Motion Sensors

To demonstrate our algorithm's effectiveness on different sensor modalities, we recorded a set of electromyography (EMG) and acceleration motion data using a Delsys Trigno wireless acquisition system. EMG recordings show electrical activity produced by skeletal muscles and are commonly analyzed in biomechanics and neurology. Acceleration recordings show the local accelerations due to changes of the sensors velocity and are commonly analyzed in biomechanics and sport sciences. Nearly all 'wearables' use accelerometer readings to analyse user activity. Typically, analysis of EMG and accelerometer signals is done on sequences already segmented into motion cycles; the segmentation is almost always done manually, and frequently relies on other readings such as motion capture or video data. Our automatic segmentation technique can significantly improve the work flow in domains using EMG and accelerometer recordings.

We take recordings from one trial subject who was asked to re-enact the sequences of subject 86 in the CMU database. Results of two repetitions of trials 3 and 12 are compared to the CMU originals (see Figure 2.13). The raw EMG readings consist of data streams of 16 sensors, each documenting the activation of large skeletal muscle groups including the larger flexors and extensors of the human body (refer to Appendix 2.7 for documentation).

All EMG recordings were pre-processed in a standard fashion: the signals were rectified, re-sampled from 2000 Hz to 30 Hz and smoothed by a 20 Hz low pass filter (see [CRA$^+$13] for a review on EMG data processing). Acceleration recordings were re-sampled from 120 Hz to 30 Hz as well and filtered by a binomial filter over a window of 16 frames.

The segements and primitives shown in Figure 2.13 show that our approach works on EMG and acceleration data as successfully as clean motion capture data, despite the former two being much noisier input sources. The EMG segments are very similar to the acceleration segments representing the same motion; in most actions, the same number of primitives were found across the two modalities. We hypothesize that the slight differences in timing are due to inherent differences in the signals being captured by the two modalities.

Clustering of the motion primitives is comparable between the two modalities for most activities. The largest differences are in 'wiping a window' (gray blocks in trial 12). Here we were unable to distinguish between back and forth wiping versus circular wiping in the EMG, while the accelerations gave clear motion primitive clusters. The EMG data does not reflect this difference since the muscle activation on the main arm extensors and flexors do not change as clearly as the accelerometer readings.

2.6.3 Kinect Action Data

Processing noisier markerless motion capture systems can be a challenge, but our method is able to deal with such data in a reliable way when the feature bundling step is included. We segment the MSRC-12 Kinect Gesture Data Set from Fothergill et al. [FMKN12], consisting of a number of action sequences which were originally recorded for action recognition. The data set consists of 594 sequences in total from 17 actors. The trials range from 1000-2000 frames recorded at 30 frames per second. The data are available as 35 joint angles of 3 scalars over a length of n frames. Examples of different segmentation results can be found in Figure 2.15.

Impact of Feature Bundling

We show the original input data streams, as well as our segmentation results, color-coded according to our clustering, both with and without feature bundling. The given key points [NS12] annotate the actual gestures at a specified key frame and are indicated by red lines in all subplots. Feature bundling allows us to segment a regular pattern of primitives that coincide with the time series (see Figure 2.15(a) and (b)). Figure 2.15(b) shows some more variation in the lengths of the primitives. This originates from a break in the input motion (first half of the trial, darkblue bar) and also some speed variation (later half of the trial) which can clearly be seen in the plot of the data stream. Figure 2.15(c) is particularly interesting because the primitive have a much finer structure. The motion primitives occur as ones recurrent groups of smaller parts (light green/green) alternating with a longer primitive (blue) and are well aligned with the keypoint annotations.

Without feature bundling, the segmented motion primitives are far less regular. In Figure 2.15(a), three repetitions were broken up into two individual parts (yellow and turquoise). In Figure 2.15(b), the longer breaks (brown) between the repetitions were found but not all repetitions (dark blue) could be cleanly separated. Finally, in Figure 2.15(c), a similar pattern was found both with and without feature bundling, but without bundling, the shorter primitives are split into smaller irregular parts that are assigned to different clusters.

Consistency Evaluation

We measure how well our identified motion primitives correspond to the given keypoint annotations. To this end, for each of the motion primitives, we measured the difference between its start frame and the key point and scale this value proportionally to the length of the primitive and plot the distribution as well as the standard deviations in Figure 2.14. The histograms shows that the key-point annotation as judged by the human annotator occurs consistently at a relative position of 40-60% in the segmented motion primitive (see Figure 2.14(a)), with a relatively low standard deviation of 15% around the mean values (see Figure 2.14(b)) when feature bundling is enabled. Figure 2.14(c) and (d) show the same values when the segmentation is done without feature bundling; here the mean values are more spread with higher standard deviations.

While the results are very consistent with respect to the given annotations when the bundled features are used, there are still a number of exceptions which contribute to the distribution of deviations. One example is shown in in Figure 2.15 (c)) where a sequence of smaller motion primitives occurs arranged in the same order. According to our goal, to identify repetitive motion primitives, the fine segmentation is the desired result. However, this is non-ideal for the consistency measure, since it considers the relative position within the motion primitive. With very fine primitives, the key points are at the end of the last small cluster (green) or at the beginning of the subsequent larger cluster (blue). Grouping together smaller primitives, in conjunction with re-clustering could alleviate this problem.

2.6.4 Non-cyclic Kinect data

We tested our approach on the more challenging MSR 3D Online Action Dataset [YLY15]. This data set contains noisy, mostly non-cyclic actions such as drinking, eating, using laptop, reading cellphone, etc. recorded with a Kinect device. We apply our method on the subset S4, which contains 36 sequences of long trials (1000-3500 frames recorded at 30 fps) from 11 subjects and is intended for continuous action recognition. The data are available as 20 3D joint positions.An example segmentation result can be found in Figure 2.16 (a) along with the corresponding SSSM (b). Since the annotations are given as intervals in this dataset we use the following measures to evaluate the quality of our results. First we measure the absolute distance in frames from the start and end points of the annotation to the next found cut. This measure alone would favor over-segmentation; thus we compensate with an additional measure of the overlap of an annotation with the largest segment found by our method . Using bundled features we obtain a mean distance between start and end points of the annotations and our primitives of 21.2 frames with a standard deviation of 19.6 frames. The mean overlap is 66.56 percent with a standard deviation of 21.16. If we use original features we obtain 22.5 (mean) 19.7 (std) and 67.2 (mean) 22.3 (std), respectively. Figure 2.16 (c)-(f) shows the histograms for these values. Since these sequences contain relatively static poses, the effect of feature bundling is not as strong as in the more dynamic data sets, since there are less variations in static poses. We were able to determine that many of the annotated actions in this data set can be split into three segments: a dynamic part in the beginning (e.g. raising cup to mouth), a static part in the middle (drinking) and another dynamic part in the end (lowering cup). This corresponds to the rough annotations in this data set where complex actions are annotated as a block, while our approach returns finer motion segments.

2.7 Data Source Documentation

We recorded our EMG and acceleration data using a combined wireless Delsys Trigno system. The sensor setup consisted of 16 sensors placed on larger muscle groups of the trial subject (see Figure 2.17 for a list of locations and a visualization of the hardware attachment). Both sensor attachment and recordings were supervised by an accredited expert.

A selection of trials performed originally by subject 86 of the CMU database was re-enacted by another human subject as true to original as possible. This was done to compare the performance of our segmentation method on other data modalities while maintaining control of the activities represented by the data streams. There are some minor deviations from the original scripts due to different geometry of the location (e.g. in one of our trials, originally 86_12, the subject had to walk some additional steps after sweeping the floor and before reaching the white board). The total length of the trials differs from the original slightly due to similar reasons. Nevertheless, the the sequences contain the same activity classes as the original sequences, and have mostly the same number of repetitions and are therefore suitable for a qualitative comparison.

In our experiments with a Delsys Trigno system, a combined sensor set of wireless electromyographs and accelerometers, we recorded some trials of human daily activities and exercises.

The sensor setup consisted of 16 sensors placed on larger muscle groups of the trial subject. Refer to Figure 2.17 for a list of locations as well as some visualization of how the hardware was attached. Both sensor attachment and recordings were supervised by an accredited expert.

A selection of trials performed originally as part of a motion capture session by subject 86 of the CMU data base was re-enacted by another human subject as true to original as possible. This was done in order to be able to compare the performance of our segmentation method on other data modalities while maintaining control of the activities represented by the data streams. There are some minor deviations from the original scripts due to different geometry of the location (e. g. in one of our trials, originally 86_12, the subject had to walk some additional steps after sweeping the floor and before reaching the white board) Also, the total length of the trials differs from the original slightly due to similar reasons. Nevertheless, the content of the trials falls into the same activity classes as the original actions and mostly has the same number of repetitions.

2.8 Conclusions

We have presented a segmentation method which is able to process a number of data modalities and separate cyclic activities and their transitions. Our approach tries to tackle the segmentation problem on a general level in terms of the choice of crucial parameters, e.g. the search radius and the feature offsets for stacking. The feature bundling is a novel contribution is this area and has proven to be especially helpful for processing noisy data modalities such as EMG, accelerometer and Kinect motion capture. We used a five-point derivation to estimate the direction of movement in the bundling, but when faced with severe noise, one will need more robust methods. This will further reduce variance in the feature space, but there are few implications as long as one does not try to synthesize new sequences from the feature space.

So far, we have only shown our segmentation on sequences taken in fairly constrained settings. We anticipate that it is also applicable to sequences taken "in the wild", given the right features and similarity measures. This remains an open topic for future work. Challenges also remain to be seen once the segmentation is even further generalized, for example to spatial segmentation problems such as mesh animation sequences [SSK05].

Since our method is based on self-similarities, the limits of finding primitives are reached when input sequences contain only non repetitive activities (e.g. one step, jump, turn). However, the assumption that most human activities are of repetitive nature is valid for motion capture data within existing data sets such as CMU [CMU13]. For other sensor modalities we found a similar behavior; if the motion is cyclic, so are the measured local accelerations. For the EMG signals we could not identify any substantial changes between the individual repetitions in the muscle activation patterns. Here changes may occur from fatigue effects in longer motion trials though this was not observed in our experimentation.

Currently, all tested data modalities have been processed individually; a natural next step is to jointly process several data types in a multi-modal setting. Combining the findings on more than one recording of the same motion could help refine segmentation and the resulting analysis. Secondly, exploring smooth embedding approaches [YGGU11] for feature bundling is a promising direction for future work. Even though a PCA works for our examples, small artifacts are nonetheless visible, which may not be the case in other non-linear (but more computationally expensive) embeddings.

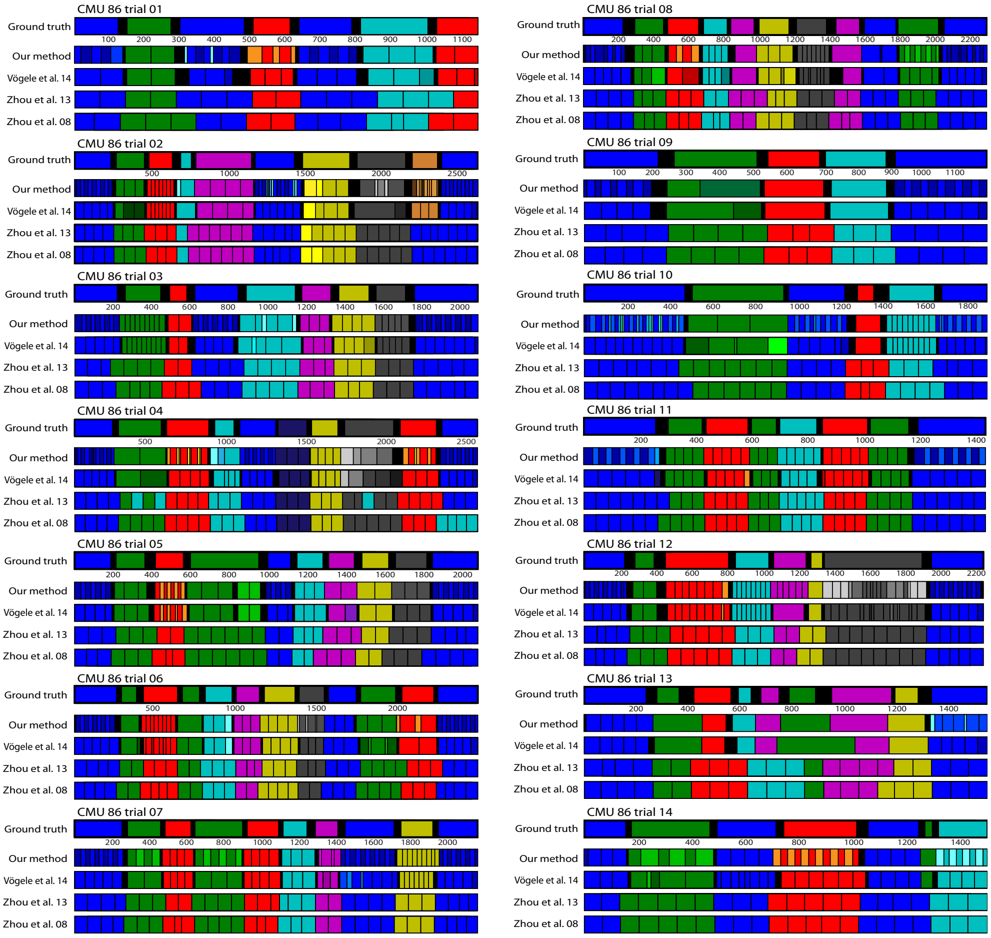


Figure 2.7: Segmentation results for CMU 86 trial 01 to 14. For each of the 14 trials, the first row displays the human ground truth annotations, the second row compares them to our results. The results of the (H)ACA methods (Zhou et al. [ZlTH08, ZlTH13]) and Vögele et al. [VKK14] are given in the two lower rows. Note that there is a variety of units of different sizes indicating that the actual length of motion primitives may vary considerably.

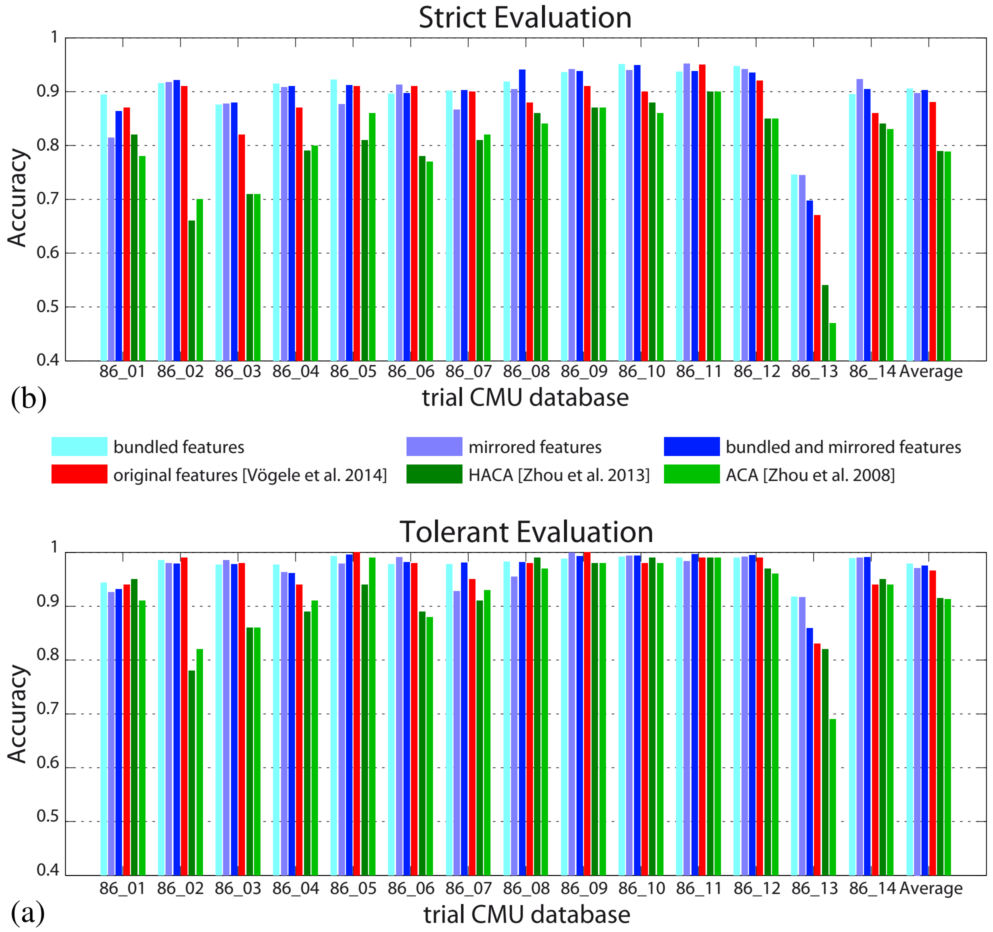


Figure 2.8: Accuracy values of different segmentation methods: Light blue bars refer to our method (bundled features only), purple: our method (mirrored features only), blue: (bundled and mirrored features combined), red: Vögele et al. [VKK14], dark green: Zhou et al.(ACA) [ZlTH08] light green: Zhou et al.(HACA) [ZlTH13] using (a) the strict evaluation, counting strictly the classes all methods detect and (b) the more tolerant evaluation which allows transitions as valid classes. For both evaluations the bundled and mirrored features give higher accuracy values in average compared to the original features and the (H)ACA based approaches. Especially the bundled features have a better effect on the accuracy compared to the mirroring.

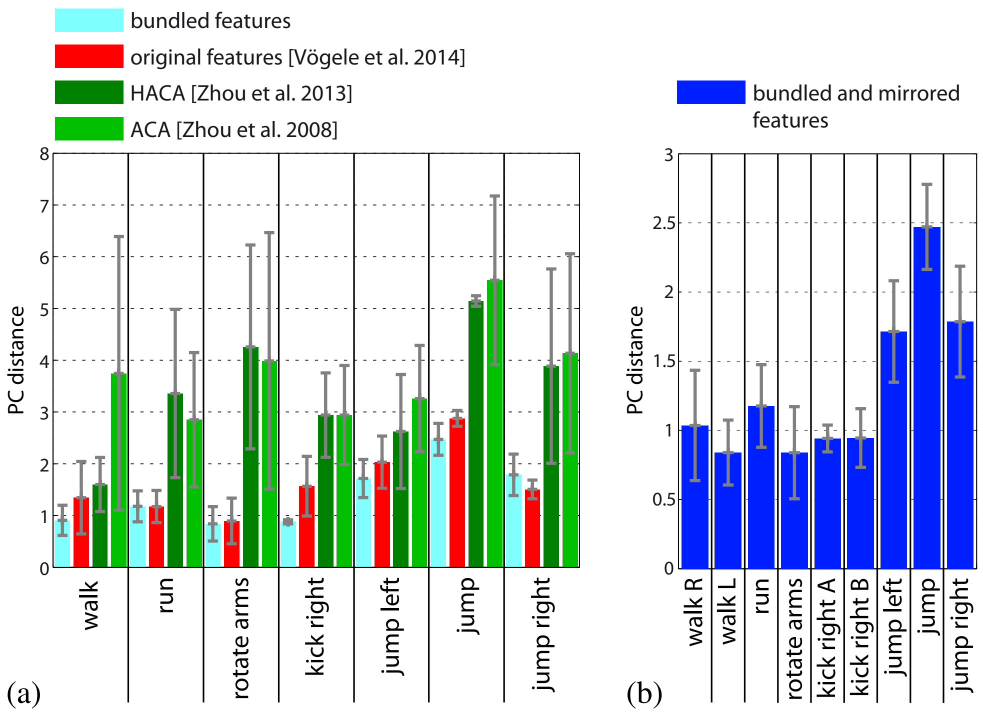


Figure 2.9: (a) Evaluation of clusters produced by our method, the method of Vögele et al., the HACA and the ACA method. The example at hand is trial 3 of CMU subject 86. The respective means are given by the blue color bars for our method, red for Vögele et al., (dark) light green color bars for (H)ACA, with variance shown as error bars. Note that lower distances reflect more consistent clusters. (b) Evaluation of clustered results produced by our method. Here we plot the mean D, with variances shown as error bars as in (a) but for the case where both bundling and symmetry features are included. This is one case where additional motion classes are introduced by exploiting symmetry of motion: there are two classes of steps and also two classes of 'kicks'. The distinctions are caused by different types of symmetry: walking is phase-shifted and kicking included one part which was symmetrical (more static) and one which was asynchronous.

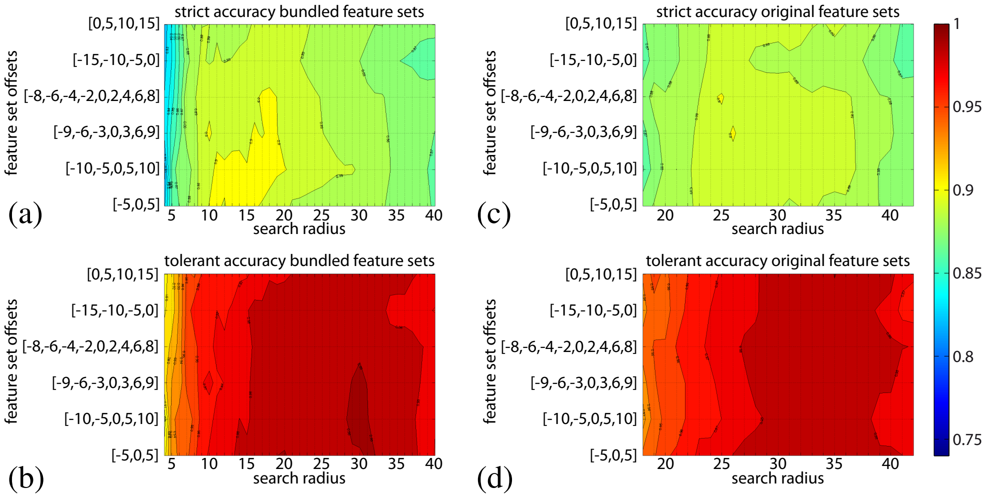


Figure 2.10: Maps displaying the accuracy values for various combinations of the parameters: The generalized radius R is plotted on the horizontal axis, choices of stacking offsets are plotted along the vertical axis. A stacking offset of $[-5, 0, 5]$ is a concatenation of frames at times $i - 5$, i and $i + 5$. Parts (a) and (b) shows the maps based on the bundled features, where (a) is the stricter and (b) is the more tolerant version. Parts (c) and (d) show the same based on the bundled features, where (c) is the stricter and (d) is the more tolerant version.

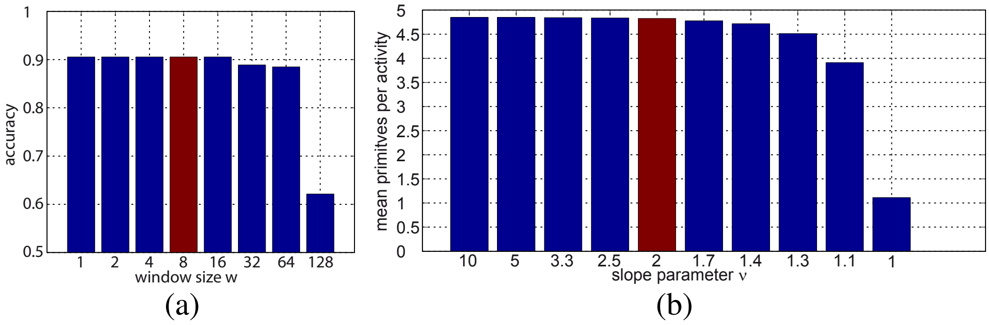
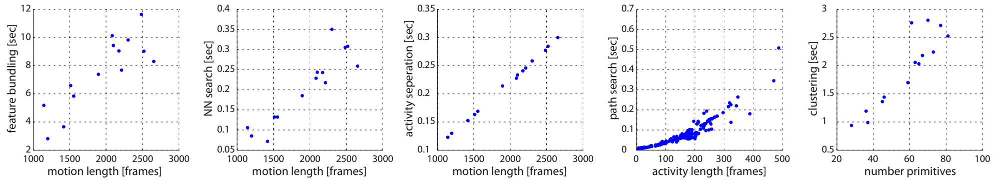


Figure 2.11: (a) Accuracy (strict evaluation) for varying window sizes w of the region growing step. $w = 8$ (red bar) was chosen for our experiments. (b) Average number of motion primitives per activity based on the minimal and maximal slope of the warping paths. $\nu = 2$ (red bar) was chosen for all other experiments in this work.



(a) (b) (c) (d) (e)

Figure 2.12: Scatter Plots for timings of the steps of our method. (a) feature bundling, (b) knn search, (c) region growing (scan lines), (d) path searches, (e) clustering. Note that the complexity of the first three steps ((a)-(c)) is linear in the number of frames, whereas path searches (d) are quadratic in the lengths of the activities. In practice, the clustering (e) also computes linear in the number of motion primitives.

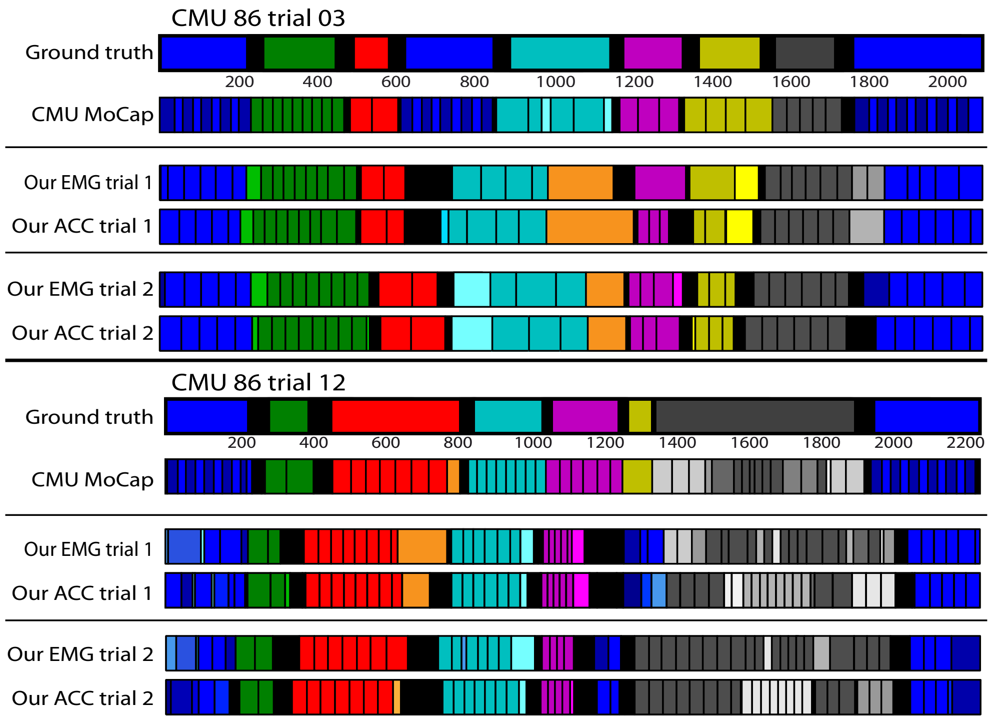


Figure 2.13: Exemplary results of two EMG trials. As can be seen from the key point visualization, this third example is also an interesting situation for the consistency measure.s of subject 86 of the CMU. The color codes correspond to the different motion clusters the primitives were assigned to by our method. The results are very similar between the data sets, even between the original and the re-enacted recordings.

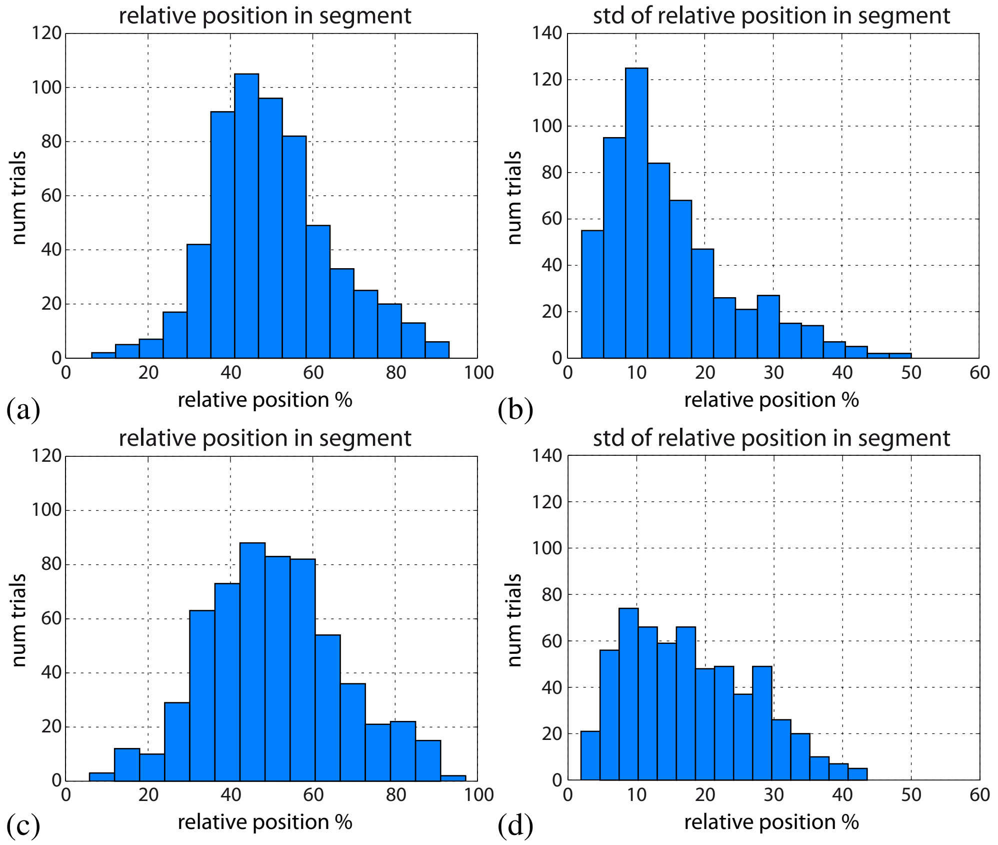


Figure 2.14: The histograms show the average location of given key point annotations (left hand side) in the motion primitive and their standard deviations (right hand side). In the majority of cases, the location of key points is approximately at the center of the corresponding primitive with a deviation of less than 20% if feature bunding is used (a) and (b). If the original features are given as input the average locations are spread with larger standard deviations (c) and (d).

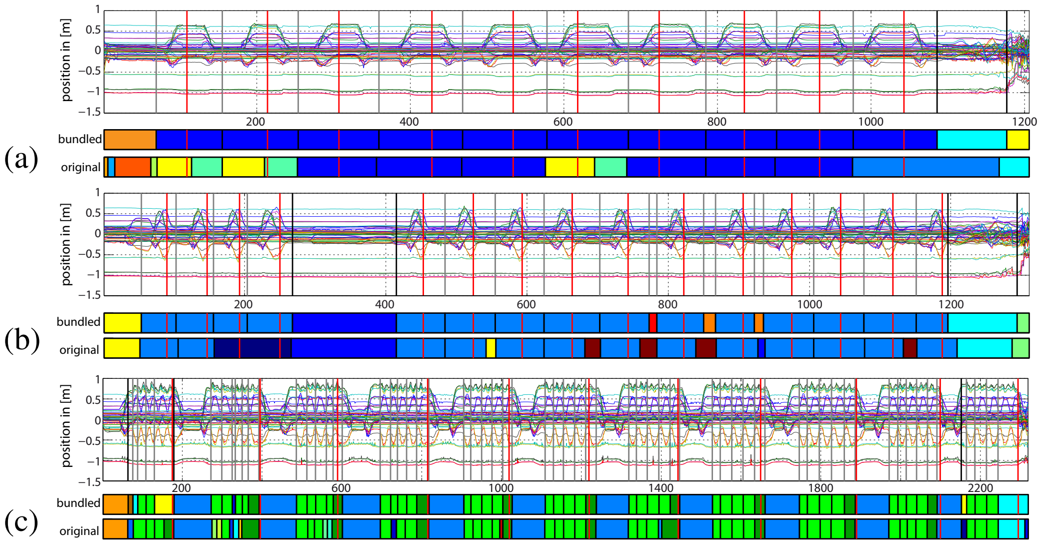


Figure 2.15: Three different results produced on Kinect recordings. The input time series are plotted for an overview of the general structures of the underlying motions, showing motions with (a) regular primitives segments [P1_2_g09_s08], (b) a sequence with a longer break and also some speed variation [P1_2_g05_s08] and (c) many quick repetitions of primitives, i. e. waving both hands (see green bars)[P3_2_g11_s29]. The very fine primitive segmentation is desirable for motion understanding, but may be punished by the consistency measure based on key point locations since the key point annotation now falls at the end or beginning of the fine primitives.

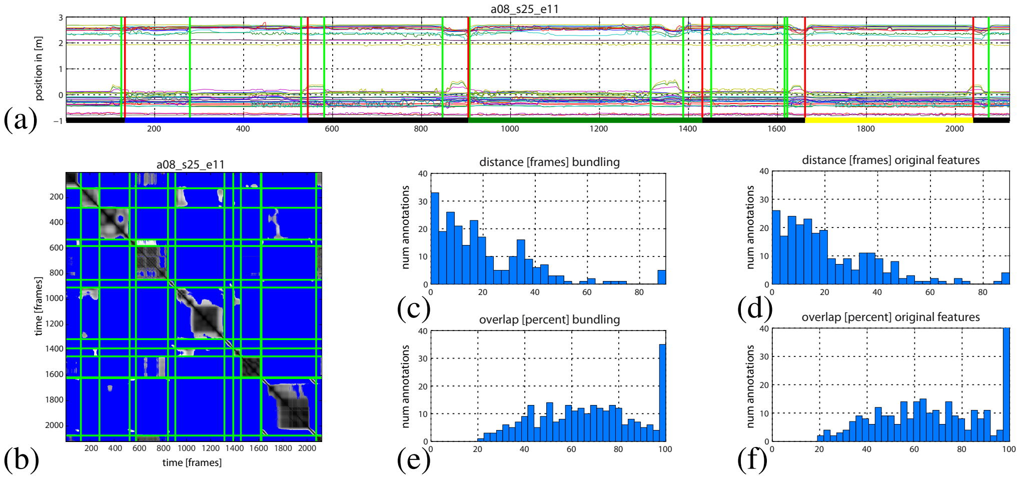
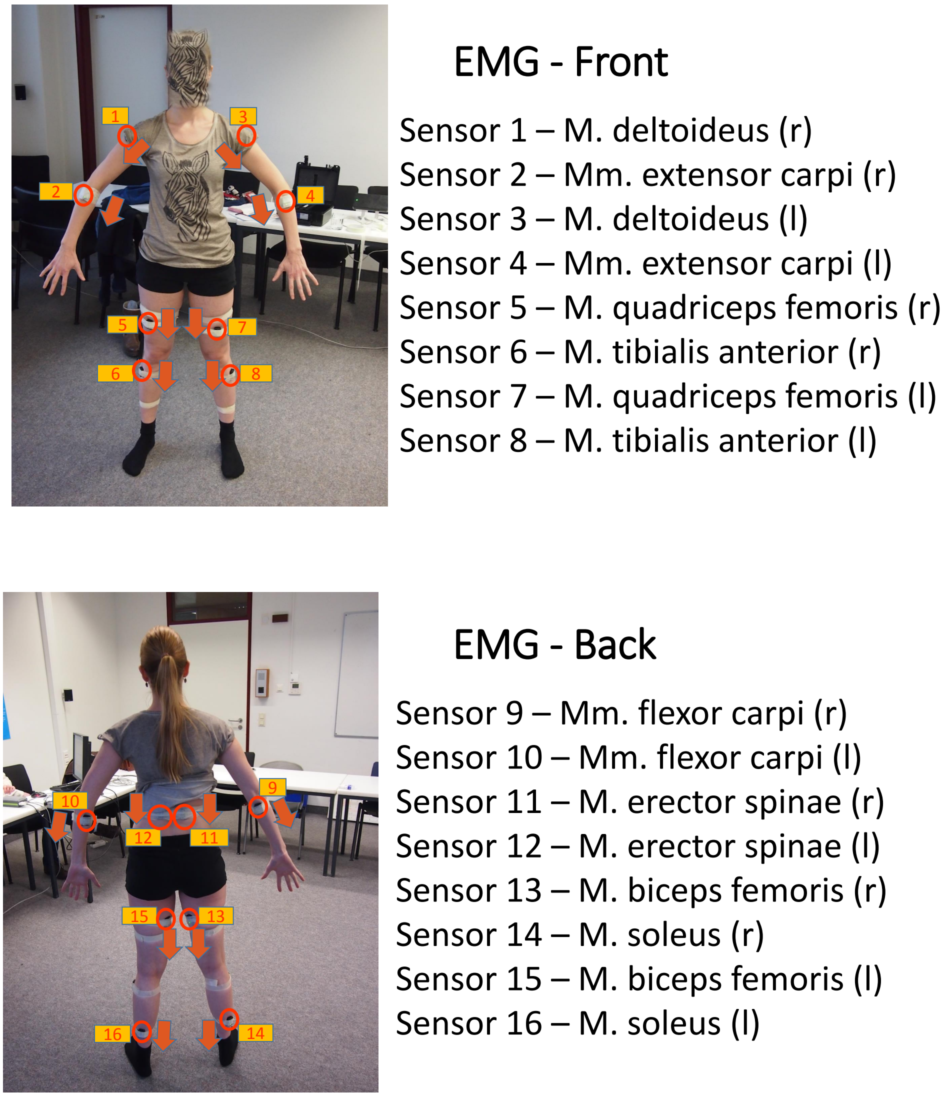


Figure 2.16: Results produced on Kinect recordings from the MSR 3D Action Dataset. Segmentation results (green lines) and ground truth annotations (red lines) for trial a08_s25_e11 (a). The bars at the bottom indicate the ground truth labels: no action (black), drinking (blue), eating (green), reading book (yellow). The corresponding SSSM with segmentation results as green lines (b). Histograms for the distance between annotations and computed cuts on bundled (c) and original features (d). Histograms showing the overlap between annotated segments and primitives identified by our method on bundled (e) and original features (f).



EMG - Front

Sensor 1 – M. deltoideus (r)
Sensor 2 – Mm. extensor carpi (r)
Sensor 3 – M. deltoideus (l)
Sensor 4 – Mm. extensor carpi (l)
Sensor 5 – M. quadriceps femoris (r)
Sensor 6 – M. tibialis anterior (r)
Sensor 7 – M. quadriceps femoris (l)
Sensor 8 – M. tibialis anterior (l)

EMG - Back

Sensor 9 – Mm. flexor carpi (r)
Sensor 10 – Mm. flexor carpi (l)
Sensor 11 – M. erector spinae (r)
Sensor 12 – M. erector spinae (l)
Sensor 13 – M. biceps femoris (r)
Sensor 14 – M. soleus (r)
Sensor 15 – M. biceps femoris (l)
Sensor 16 – M. soleus (l)

Figure 2.17: Documentation of EMG sensor placement on a human subject seen from the front and back, respectively. Orange arrows point to direction of y-axis.

Part II

Analysis

CHAPTER 3

ANALYSIS OF MULTIVARIATE TIME SERIES

Get the habit of analysis - analysis will in time enable synthesis to become your habit of mind.

Frank Lloyd Wright.

Analysis of multivariate and high-dimensional data is a field of statistical and probabilistic mathematics in which techniques are applied to extract and enhance, visualize and investigate, group and sort information captured in complex data structures. This field belongs to a discipline called *data science* which has recently become popular. Data science is based on the recent findings (cf Dhar [Dha13], [Sci]) that over 90 % of data in the world were generated within the preceding two years. Realizing this gave rise to the idea that processing such big data amount, even if only partly, would require a new focus on automatic and versatile data processing methods from different scientific areas.

In this connection, the analysis of multivariate time series can be considered as a particularly challenging area where information given over time leads to even faster growing amounts of data requiring analysis. Examples of such data are motion recordings, audio signals and financial data. The growing amounts of data associated to these areas call for effective analysis methods related to statistics and computer science.

Figure 3.1 shows how analysis of multivariate data fits in the context of our more general pipeline (Chapter 1, Figure 9.2). Many techniques associated with time series analysis can be used independently of the given data type but this thesis will focus on examples of motion data.

Pre-processing of multivariate data sets is crucial, as it helps get first insights into information contained the data. Filtering techniques can extract important information from noisy input. They can also help to protect sensitive data like personal information on test subjects. Some examples of different filtering techniques are introduced in Chapter 4. Chapter 4 is based on two research papers

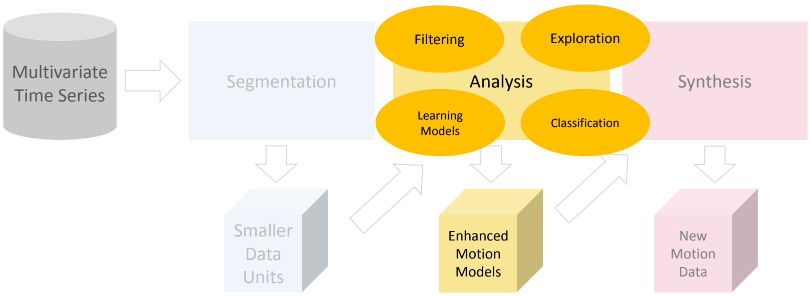


Figure 3.1: Analysis of multivariate time series accommodates a range of topics. Examples of the four areas filtering, exploration, learning models and classification are discussed in Chapters 4 ff.

[KVH+14, KVL+14] published in a collaboration with researchers from the Interdisciplinary Center of Sleep Medicine at the Charité University Hospital in Berlin, Germany. In our joint research, we investigated techniques to monitor human sleep patterns in a non-invasive non-clinical way with the help of depth cameras (Kinect). Filtering techniques played a big part in this for two major purposes. One purpose is to extract information in space and time, the other purpose is image simplification for de-identification of human subjects in videos.

Exploration of multivariate data is another important analytical goal. Applied scientists from a number of fields like biology, sports or medicine benefit from tools for visual exploration and preliminary analysis. Making mathematical methods accessible to them in this way contributes highly towards interdisciplinary scientific work. In practice, basic visualization techniques are extended to high-dimensional temporal information. Extracting features and grouping data according to specific objectives allows insight into data subgroups. Designing and training descriptive models which characterize given information makes linking meta-information to given motion data possible and facilitates different types of querying. As a paramount example, Chapter 5 describes FuryExplorer, a system to explore and analyze equine motion capture data. The chapter is based on a research paper [WVZ+15] published collaboratively with a research group from Fraunhofer Institute for Computer Graphics Research (IGD) in Darmstadt, Germany as well as the University of Darmstadt, Germany and the University of Natural Resources and Life Sciences and the University of Veterinary Medicine in Vienna, Austria. This project was partly supported by Deutsche Forschungsgemeinschaft (DFG) under research grants KR 4309/2-1.

Creating statistical models for the investigation of motion is an important part of multivariate data analysis. Based on simple units of motion (e. g. as discussed in Chapter 2), activity patterns can be grouped into descriptive models. One example is discussed in Chapter 6 based on a research paper [VZKL16] on the analysis of animal muscle activity in the field of biomechanics. The research was part of a collaboration with researchers from the University of Natural Resources and Life Sciences and the University of Veterinary Medicine, both in Vienna, Austria. It was part of the GeMMQuad research project supported by Deutsche Forschungsgemeinschaft (DFG) under research grants KR 4309/2-1.

The concluding example is about classification of human motion. Classification problems can be posed with two different directions in mind. First of all, in order to specify the class of action performed by a subject. The classification of sleeping vs. awake discussed in Chapter 4 is an example of this type. Second, in order to detect information about the human subject performing a pre-defined task. The paper of Riaz et al. [RVKW15] introduces solutions to classification tasks of this type. Chapter 7 gives a brief overview on classification of human soft biometrics.

CHAPTER 4

Detection of Human Sleep Patterns by Analysis of Depth Data

You don't want me to look at you?
Then you look at me.

Kaa, The Djungle Book

Different techniques of filtering data can be used towards a variety of goals ranging from finding relevant data to hiding personal information. The papers [KVH$^+$14, KVL$^+$14] feature both above-mentioned goals. In the following, a brief summary of some applications is given.

In recent years, the market for wearable technology designed to assist consumers in health care and fitness has virtually exploded. Since the so-called 'quantified self movement' has taken off, various companies have launched smaller and more powerful sensors and consumer electronics devices which allow for tracking of fitness, physical activity, health, diet and other personal metrics.

Although sleeping is a significant component of human activity it involves being relatively inactive. Nevertheless, the quality of sleep is a vital factor in the overall quality of human performance during daytime. Usually, the analysis of serious sleep disorders requires invasive lab-based monitoring. However, such measurements may also be partly replaced by actigraphy as a basic means to measure sleep quality. An overview of the consumer electronics devices that analyze frequency of wrist movement over time or similar information is found in [Inc].

While wearable devices like the above-mentioned are popular tools to monitor sports and everyday motions during waking hours, reservations against wearing such devices during sleep are a common phenomenon due to their causing impairments on the quality of sleep. In this work we focus on measuring the sleep phases by means of consumer electronics hardware but without immediate physical contact to the subjects. To demonstrate this technology achieves competitive accuracy, we compare our evaluations of Kinect recordings with the results from a professional sleep laboratory.

The standard approach to studying sleep and the detection of specific sleep disorders is a polysomnographic monitoring of patients in a clinical sleep laboratory. This multi-parametric test results in a comprehensive recording of the biophysiological changes that occur during sleep. A survey on standardized specifications and techniques for characterization of natural sleep is given in the scoring manual of Iber et al. [IAICftAAoSM07]. A revision of existing methods for scoring sleep stages as well as new techniques to gather information from the polysomnographic readings are presented by Silber et al. [SAIB$^+$07] as part of a visual sleep scoring initiative. Standards of practice and clinical diagnostics by means of cardiorespiratory polysomnography are found in the contribution of Penzel [Pen06].

As alternatives to studying sleep in the clinical laboratory alone, there is also a wide range of options to record data with professional equipment but within the the patient's domestic environment. A study to compare home polysomnography with the in-lab approach in the diagnosis of sleeping disorders was done by Porter et al. [PPC$^+$03]. Their evaluation showed that unattended polysomnography was still problematic in a number of cases since diagnosis depends on the quality of data obtained under unattended conditions. However, since there is an ongoing trend towards monitoring sleep in an environment comfortable for patients, the necessity to meet the obstructions of self-monitoring sleep quality is vital. The increasing variety of mobile devices for self-recording personal metrics can help alleviate by offering more consumer friendly options of data acquisition.

We conducted a study in order to prove that it is indeed possible to achieve results comparable to clinical polysomnography by recording depth data of sleeping patients with a Kinect device.

To this end, we recorded the sleep of a patient with a Kinect mounted to a panel under his bedroom ceiling. The analysis of the resulting depth data was started by filtering the data in two ways, thus quantifying the amount of relevant motion in the data. First, we computed visual motion summaries showing the average motion $\mathcal{D}$ for every pixel $p_{i,j}$ over a fixed time period of T frames:

$$\mathcal{D}(p_{i,j}) = \frac{1}{T} \sum_{t=1}^{T-1} |p_{i,j}(t) - p_{i,j}(t+1)| \tag{4.1}$$

The results for every pixel of the input image sequence are color coded, starting from black ($\mathcal{D} = 0$) to white ($\mathcal{D} = \max(\mathcal{D}(p_{i,j}))$ for all indices i, j in the image, i.e. $i = 1, \ldots, n$ and $j = 1, \ldots, m$). This allowed for detection of sleep intervals. Second, we computed the average change $\mathcal{N}$ per frame, i. e. an overview on the spatial distribution of motion in the input image sequence. This allowed motion detection in the spacial domain.

$$\mathcal{N}(t) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |p_{i,j}(t) - p_{i,j}(t+1)| \tag{4.2}$$

This allowed motion detection in the temporal domain.

We used an adaption of Cole and Kripke's algorithm (Kripke et al. [KHG$^+$10], [CKG$^+$92]) to classify sleep intervals and used the results of polysomnographic sleep detection as gold standard for evaluation of the results. Kripke's algorithm (originally used on data recorded by an actimetry sensor) works as follows: Based on the data, actigraphic activity counts are performed once every 30 seconds. The score $x_i \in \{0, 1\}$ given at each time $t = j$, $x_j = 0$ for inactive, $x_j = 1$ for active. The scores per epoch are collected and evaluated by

$$B(t) = \mathrm{s} \sum_{\hat{t}=t_s}^{t_e} b_{\hat{t}} x_{\hat{t}}. \tag{4.3}$$

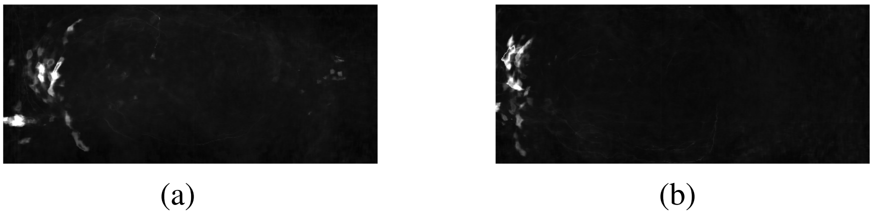
In our case we fixed $t_s = -10$ and $t_e = 3$, s was an overall scaling parameter for normalization and the weights $b_{\hat{t}}$ are given by a Gaussian function. The status of the subject is then classified as 'sleep' (positive) or 'wake' (negative): $D \leq \alpha$ means the subject is awake, $D \geq \alpha$ means he is asleep.

To evaluate our approach we measured two nights in a sleep laboratory environment. To this end, we recorded the subject with our setup and in parallel with the full equipment of the sleep laboratory. Annotations performed by the lab assistants on the basis of their data are used as ground truth reference. Thus, we were able to evaluate the scoring results, x_θ, with respect to the ground truth given by standard polysomnography.

In our first paper on the subject (Krüger et al. [KVH$^+$14]), we showed that the results achieved by this procedure largely matched the results of clinical polysomnography.

Naturally, there is an increasing awareness of privacy issues when personal data are involved. This is important when monitoring people in their own homes, particularly when they are not conscious of their surroundings. In order to tackle the problem of protecting people's privacy we introduced a blurring method to the video and depth data used in [KVH$^+$14]. The evaluations of our follow-up (Krüger et al. [KVL$^+$14]) show that this method obscures the identity of subjects in the data but does not impair our sleep detection method to a significant level.

We use an image simplification technique in order to make persons unrecognizeable while still keeping track of the motion that is performed. To this end we employ a simple PCA based approach. First, we split the image into N quadratic



(a) (b)

Figure 4.1: Visual Motion Summary. (a) first night, (b) second night.

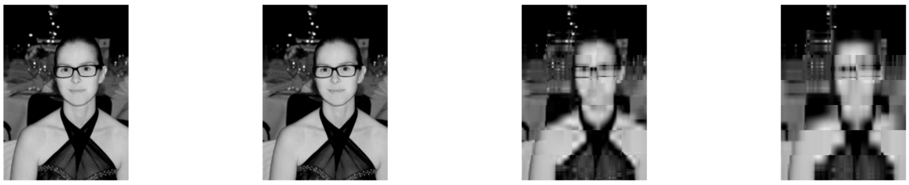


Figure 4.2: Example for simplified images. Original image, 16 PCs, 2 PCs 1 PC.

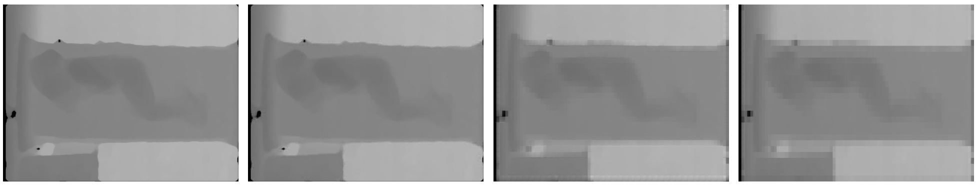


Figure 4.3: Example for simplified depth images. Original data, 16 PCs, 2 PCs, 1 PC.

blocks of size $n \times n$. Second, Each of this blocks is transformed to a vector of size $n^2 \times 1$. Thus, the image is now transformed to an $n^2 \times N$ data matrix. Third, we apply a standard principal component analysis (PCA) to this data. Fourth, we truncate the data in PCA space to the first M principal components (with $M \leq n^2$). Fifth, we obtain our simplified image by transforming the truncated data back to the data space and reshaping the data back to the original pixel positions.

Figure 7.1 shows a sequence results of gray scale images. We show the original (left) image and the modified versions based on 16, 2 and 1 principal components. Similar we show a sequence based on depth images in Figure 4.3.

All in all. our tests achieved fairly good results and proved the effectiveness of unattended sleep detection in the subjects' home by means of a Kinect or similar device. The simplification of images did not result in a significant decrease in accuracy. While our study, above all, serves as prove of concept, future work could focus on a number of convincing tests and the consideration of a reasonably large group of test subjects.

CHAPTER 5

EXPLORATION OF MULTIVARIATE TIME SERIES - EQUINE MOTION

> Horse sense is stable thinking.
>
> Unknown

Since the beginning of the domestication of horses (see e.g., [DG68]), humans have always been interested in equine athletic abilities and have integrated these into work, warfare and culture. In a popular example 130 years ago, Muybridge [Muy78] showed a series of eleven subsequent photographs taken of one gallop motion cycle in an animated loop (see Figure 5.1). Biomechanical analysis has consequently been pursued at an increasing level by modern veterinary medicine, especially since the rise of technical capturing of equine motion. Motion capturing is increasingly gaining importance in a variety of research fields, such as computer animation, sports, and medicine. Similarly, this trend can be expected in the equine community. The focus of interest will shift away from pure observation towards the incorporation of visual-interactive data analysis techniques.

The work at hand is focused on motion capture data recorded in the field of veterinary medicine. 3D positions of markers attached to the animal's body are tracked in an in-lab scenario by multiple cameras with high spatial and temporal accuracy. The resulting data collections are enriched with meta data, stored in various repositories, and may subsequently be analyzed as multivariate time series data. From an information visualization (IV) and visual analytics (VA) perspective, such multivariate time series are considered a challenging data type and the number of approaches is relatively small compared to those for the analysis of uni-variate time series.

From a veterinary perspective, we observe a very similar situation. The analysis of motion capture data is cumbersome. By now, only simple tools exist for the detailed analysis of recorded motion data. Visual-interactive interfaces to support the scientific work flow of trained veterinarians are missing entirely. As a

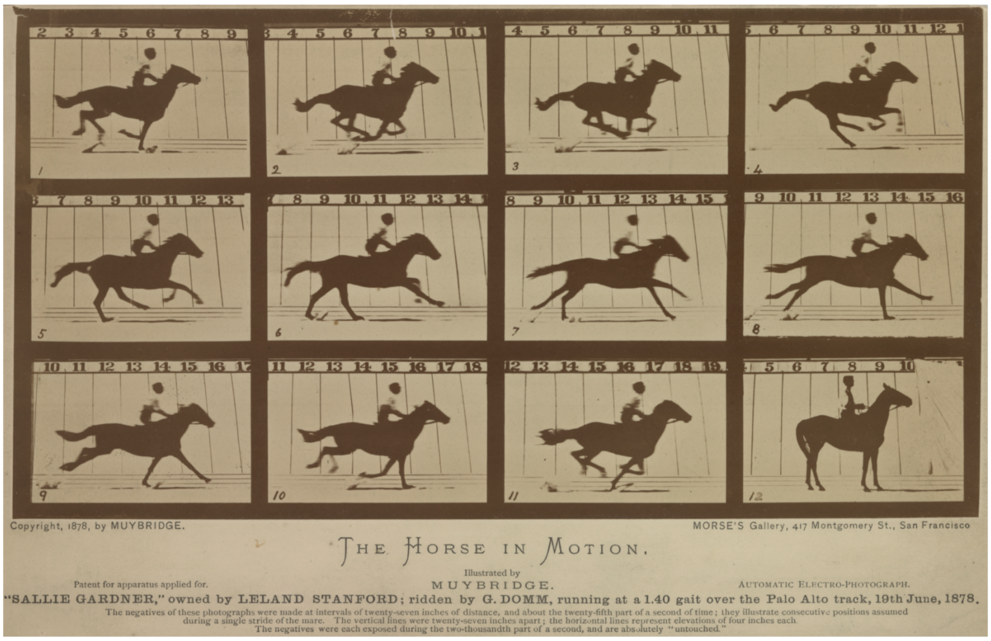

1
2
3
4
5
6
7
8
9
10
11
12
Copyright, 1878, by MUYBRIDGE.
MORSE'S Gallery, 417 Montgomery St., San Francisco
THE HORSE IN MOTION.
Illustrated by
MUYBRIDGE.
Patent for apparatus applied for.
AUTOMATIC ELECTRO-PHOTOGRAPH.
"SALLIE GARDNER," owned by LELAND STANFORD; ridden by G. DOMM, running at a 1.40 gait over the Palo Alto track, 19th June, 1878.
The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each.
The negatives were each exposed during the two-thousandth part of a second, and are absolutely "untouched."

Figure 5.1: Muybridges series of photos taken from a gallop motion cycle [Muy78]

result, typical tasks are currently limited to the analysis of single recordings only, particularly since there is a lack of tools to *compare* multiple measurements at an appropriate rate. Currently, veterinarians need to perform time-consuming and tedious tasks with only low analytical benefit. Therefore, the focus of this work will be on five important challenges:

1. Missing overview capability for large data collections,

2. Querying techniques to retrieve relevant subsets,

3. Missing functionality for drilling down to features of interest and localization of relevant subsets,

4. Visualization of characteristics of both individuals and groups of horses

5. Lack of techniques for relating metadata with poses and motions.

We present the FuryExplorer system, the result of a collaborative project between domain experts in veterinary medicine, information retrieval experts in motion capturing, and data scientists from IV and VA. FuryExplorer is one of the very first VA solutions in the veterinary medicine domain to explore large collections of horse motion capture data. Inspired by Muybridge's visualization, FuryExplorer provides an *overview of horse poses* in a grid layout based on the Self-organizing Map (SOM) clustering algorithm. The complementary means of the SOM display and an additional PCA-based view allows for the comparative *analysis of equine motions*, facilitated by projection-based visual trajectory analysis. Moreover, the overview visualizations of FuryExplorer enable domain experts to *interactively drill-down* to relevant data subsets. As a complement, equine motion (sub-)sequences can be retrieved by *visually querying* single poses *by-example*. *Local aspects* of interest can be analyzed in a list-based sequence view on demand. Here, the temporal progression of a user-definable set of markers can be explored; of a single individual and, moreover, of multiple horses. We show the usability and the usefulness of FuryExplorer in two real-world use cases and in a user study conducted with domain experts of veterinary medicine. The results show that domain experts benefit from improved effectiveness and efficiency. FuryExplorer was developed and evaluated in accordance with domain experts in a design study setup [Mun09]. The work at hand describes the three different phases of the project consisting of a domain characterization phase in Section 5.2, an iterative system development phase in the Sections 5.3 and 5.4, and an evaluation phase containing use cases in Section 5.5, as well as a conducted user study in Section 5.6. We reflect the lessons learned in Section 5.7, Section 5.8 concludes.

5.1 Related Work

5.1.1 Analytical Tools in the Equine Motion Capture Data Domain

Motion capture has become a standard technique in various areas, such as computer animation, sports, and medicine. A variety of systems is on the market, starting from consumer devices, such as the Kinect[1] sensor, up to professional systems offered by Vicon[2] or MotionAnalysis[3]. A comprehensive overview of the available techniques is given by Moeslund et al. [MHK06]. There is an increasing amount of high quality motion capture data and techniques to handle large datasets. Especially, retrieval techniques are gaining importance: Kovar et al. [KG04] introduced a point cloud-based distance measure. Although providing a closed form solution to compute the distance between two poses, this measure is still computationally complex. Krüger et al. [KTWZ10] avoid this problem by considering feature vectors in a so called *pose space* where the Euclidean distance

[1] http://www.xbox.com/en-US/xbox360/accessories/kinect

[2] http://www.vicon.com/

[3] http://www.motionanalysis.com/

is giving results similar to the point cloud distance. In addition, similarity searches can be performed employing a kd-tree index structure which significantly reduces complexity. With this kind of representation, even data-driven online applications (cf Tautges et al. [TZK$^+$11]) are feasible.

Recently motion capturing techniques have been established for kinematics and kinetics in veterinary medicine to facilitate specifically the analysis and comparison of quadruped gaits. Licka et al. [LPZ01] study horses' gaits on a treadmill. They record the positions of markers on the hooves, head, and the skin above the spinous processes of several vertebrae by motion capture in order to examine lateral back movement. More recently, Burn and Brockington [BB01] showed that optical motion capture can be used to measure common patterns of hoof deformation during locomotion in healthy horses. An overview of the kinematic and kinetic analytical techniques introduced to veterinary medicine can be found in the work of Gillette and Angle [GA08].

The authors describe motion capture for quadrupeds as a means of supporting locomotion analysis with a focus on canine gaits. Recently, some of the retrieval techniques have been transferred from human motion to quadruped motions (cf Zsoldos et al. [ZKWL13] and Krüger et al. [KYZW14]. Thus, it is now possible to retrieve similar poses from large data collections of horse motion capture data in real time — a requirement for the presented system. An information visualization approach for the visual analysis of animal behavior was presented by Grundy et al. [GJL$^+$09]. The data of tri-axial accelerometer sensors is visualized as 3D trajectories on a globe metaphor allowing for the analysis of cormorant activity.

5.1.2 Data Aggregation and Projection of Multivariate Time Series Data

We review time series data aggregation and projection techniques which Fury-Explorer utilizes to provide an overview of multivariate time series data sets. For comprehensive surveys on visualization techniques for (multivariate) time-oriented data on a more general level, we refer to the books of Andrienko and Andrienko [AA06] and of Aigner et al. [AMST11].

Multivariate time series data are complex with respect to the number of objects and to the number of time-dependent dimensions [AMST11]. Especially for large data sets an *abstraction* of the data is crucial to improve the scalability and reduce the cognitive load of the analyst. From a technical perspective two strategies are conceivable to downsize the data complexity. *Data aggregation techniques* (cf Elmqvist and Fekete [EF10], McLachlan et al. [MMKN08]) can be applied to reduce the number of data objects and thus, the size of the data set. As a result, the multidimensional data objects are assigned to a feasible number of groups, e.g., by clustering techniques (cf Berkhin [Ber06]). Liao [Lia05] analyzes the way existing clustering techniques have been modified to support time-series data. The calendar view technique presented by van Wijk et al. [VWVS99] can be seen as a best practice example. A disadvantage of data aggregation is that most existing variants apriori do not provide an intuitive order of the resulting groups if no additional layout technique is applied. Alternatively, *dimension reduction techniques* can be used to reduce the number of attributes/dimensions. Techniques exist to project/layout multivariate time-oriented data into 2D, facilitating a topological order for their visual representation (cf Ward and Guo [WG11], Bernard et al. [BWS$^+$12]). However, since most techniques do not reduce the number of data objects, the visual display becomes overplotted if no additional data aggregation technique is applied [EF10]. Different solutions have been presented in the past to overcome respective drawbacks by combining aggregation and projection techniques. Some approaches use (interactive) data aggregation, followed by an additional projection step (cf Spurlock et al. [SCW$^+$10], Bernard et al. [BWK$^+$13]. We neglect this variant since the generalizability may be compromised when other data sets with unknown intrinsic numbers of clusters are to be analyzed. Alternative approaches firstly project data objects into the display space and then aggregate the low-dimensional data representations (cf Steiger et al. [SBM$^+$14]). We also neglect this procedure since we cannot guarantee the validity of such aggregation results for any given (and foreseeable) feature vector configuration.

In the end, the Self-organizing Maps (SOM) algorithm (cf Kohonen et al. [KSH01]) combines both data aggregation and data projection in a single step. The non-linear SOM algorithm highlights the most frequent patterns of a data set (cf Schreck et al. [SBVLK09]). As a complementary means, we incorporate the linear PCA projection algorithm (cf Jolliffe [Jol86]) to a) provide a second view on the data and to b) facilitate the identification of outliers.

5.2 Domain Characterization

We characterize the equine motion analysis domain as part of veterinary medicine on the basis our insights gained in an early accustomization phase. Monitoring and interviewing domain experts carrying out routines specific to their field introduced us to the domain-specific jargon, as well as their analysis goals and the analytical tasks. In turn, we presented VA tools and visual paper designs in order to introduce the domain experts to state of the art in VA. In this section, we characterize what the domain *is*, what the domain *does*, and what the current *challenges* are. We finish with a resulting list of technical *requirements* for FuryExplorer.

5.2.1 The Equine Biomechanics Domain

Biomechanics is the science of movement of the living body, including the interaction of muscles, bones, tendons, and ligaments in order to produce movement. Equine biomechanical analysis as carried out by modern veterinary medicine has rapidly developed based on emerging technologies. Enhanced optical systems and inertial sensors have become available for recording locomotion data. This enabled targeting a number of aspects to describe or detect equine characteristics indicative of good performance. As veterinarians study the effects of medication and athletic exercise on quadrupeds, particularly horses (cf Murray et al. [MZT^{+}02]), questions about the long-term and side effects of these influences arise. Within the domain of interest, horse-rider interaction (cf Weeren and Black [vW05, vWB14]) in equine sports and industry is also an upcoming topic. All such matters converge in gait analysis as an important means of equine muscular-skeletal systems diagnostics (cf Seeherman [See91]). From the rapid development of motion capture, it can be expected that enhanced data analysis methods will have highly increased applicability, efficiency and effectiveness in future. Concurrently with this, the development of IV techniques is crucial for a fast evaluation, interpretation and presentation of the involved data. This particularly facilitates short-term or immediate clinical decision making.

Analytical Goals in the Equine Biomechanics Domain As a first step of the medical analysis process, horses are usually visually inspected by a trained veterinarian. Additional findings of other medical examination can then be interpreted together with these visual impressions. When 3D motion capture data are collected to support this visual biomechanical analysis, the 3D marker data are manually processed and then interpreted by visual inspection. Graphical representations of locomotion data collected of one individual horse often support the quantitative and qualitative analysis. For the evaluation of the result, movements of specific body parts are usually studied in relation to others. Additionally, stride cycles discernible in the data to an expert are compared with the data representations. Since only few of these steps have been automated so far, conducting analyses and evaluations is often painfully time consuming and requires a lot of expertise. There are a number of different focuses of the above-mentioned analyses, we characterize two analytical tasks in detail.

The *comparison of different gaits* is important in order to analyse intrinsic properties of locomotion and the various strains the equine body can be subject to. Specifically, the comparative analysis of the same gait in different individuals or one gait of one individual at different stages of recovery from a medical condition is of particular interest. Ideally, such evaluation includes warping one hoof print pattern onto another, thus quantifying changes and irregularities in the skeletal and locomotor system. Carrying out this kind of evaluation is only possible by manual processing of marker data until now.

For *lameness analysis* a single horse with known orthopaedic pain is observed and measured in order to assess the changes this pain creates in the execution of its movements. This is done by observing the interaction of all body parts during each stride, especially by assessing the degree of elastic energy stored in each leg. The movement of each leg is further evaluated by quantification of stride height, length, and arch of the stride. Lameness of stance phase or swing phase can be detected by these characteristics.

Rendering the horse pain free with pain relief or selective local anaesthesia allows for a comparative evaluation. However, orthopaedic pain cannot be reduced in a single day of pain relief. Of particular interest is monitoring the development of horses' health status (the changes in pose and motion) over a longer period of recovery time (a minimum of two weeks).

In equine lameness evaluation, subjective assessments are still most commonly used as they are the only reliable source of information. The lack of transferability of such assessments makes it difficult to compare evaluations of different veterinarians. Similarly, single evaluators have problems to compare results of longer periods.

Identification of Unsolved Analytical Challenges Current motion capture data visualization tools bring about a number of problems still unsolved.

The tools currently available to the community lack opportunities for interactively browsing data sets. However, an advanced visual overview, enhanced exploration by browsing motion data as well as supplementary support are among the most important priorities indicated by domain experts.

In particular, there is a need for advanced information drill-down capability. Interactive subset selection techniques as well as visual query formulation techniques coupled with retrieval algorithms have not been used in practice so far. Such visual-interactive definitions of interesting sub-sets (cohorts) would allow for the comparison of both poses and motions present in the data sets. The comparison of motions of different individuals would be of particular benefit.

Another concern is the slow pace of current data analyses, especially since it is not yet possible to perform any analysis throughout the process of recording the motion capture data. In order to make data acquisition more efficient, one big challenge is to create immediate visual feedback of all markers involved in a recording session at the same time.

Another challenge is the support of analysis goals of domain experts focusing on specific body parts. Investigations of specific kinematic properties, particularly with respect to a subject's posture of the neck, leg movement, and ground contact of hooves will explain in what ways an individual's performance is affected by observed tendencies or irregularities. Being able to correlate such factors to known metadata would create a whole new basis of insights, thus forming new theories in fields such as genetics and bio-kinematics.

However, this requires a novel system which allows for the definition of markers of interest and more sophisticated techniques to analyze data. In particular, this includes the analysis of reoccurring patterns and also of the relation of different data within user-defined data subgroups. Examples presented in this design study are healthy vs. ill individuals and data of specific body parts, as well as restrictions to selected marker configurations.

Even though some of these requirements may already have been met by existing systems, those currently available are hard to use for novice users. As the domain experts cannot be expected to be trained experts in using algorithmic tools, the existing software is mostly not suitable for them.

5.2.2 Results of the Technical Requirements Analysis

As a result of the domain characterization, we describe requirements for the system from a technical perspective. Core unsolved challenges refer to the ability to visually access multivariate time series data and to make them analyzable for the targeted domain.

We identify the need for an overview of large data sets as well as for an interactive functionality to drill-down the contained information to subsets in higher detail on demand. We aim to incorporate query-by-example functionality into this analytical workflow. Moreover, options to visually compare different horse poses and horse motion sequences at a glance are of particular importance since such analytical tasks are still rarely feasible for the domain experts. We conclude the domain characterization phase with the list of technical requirements in Table 5.1, which we utilized to build FuryExplorer and evaluate the system afterwards.

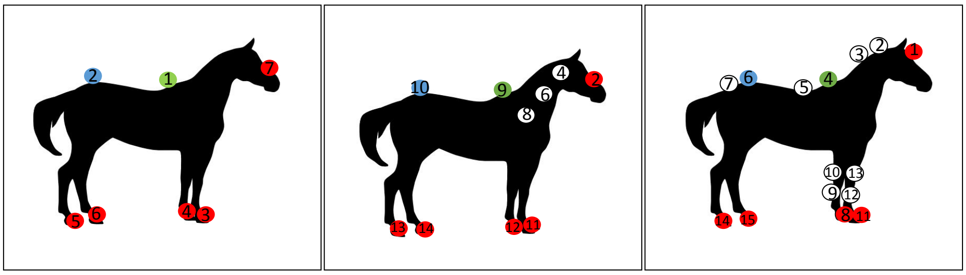
5.3 Data Abstraction and Functional Support

We characterize the data in Section 5.3.1, followed by a description of applied preprocessing routines in Section 5.3.2. The aggregation of poses to provide a content-based overview is described in Section 5.3.3. The retrieval algorithm for motion sequences is presented in Section 5.3.4.

Basis marker setup: Markers that can be matched in all provided motion files

Marker Setup 1: Dense marker configuration around the neck

Marker Setup 2: Dense marker configuration along the spine and fore legs



1: Withers	1: CristaFacialisL (occluded)	9: Withers	1: Forehead	9: FrontAnkleRight
2: Sacrum	2: CristaFacialisR	10: Sacrum	2: C1	10: CarpalRight
3: Right Fore	3: C1L (occluded)	11: Right Fore	3: Th1	11: FrontHoofLeft
4: Left Fore	4: C1R	12: Left Fore	4: Th5	12: FrontAnkleLeft
5: Right Hind	5: C3L (occluded)	13: Right Hind	5: L3	13: CarpalLeft
6: Left Hind	6: C3R	14: Left Hind	6: S1	14: HindHoofRight
7: Forehead	7: C6L (occluded)	15: Forehead (virtual)	7: S2	15: HindHoofLeft
	8: C6R		8: FrontHoofRight	

Figure 5.2: A canonical basis marker setup (left) allows for the comparison of different real-world setups applied in this study.

5.3.1 Data Characterization

The data provided by the domain experts consists of motion capture measurements of several different horses. Each animal is fitted with a set of markers at specific body positions and its motion is recorded by tracking the markers. To this end a optical motion capture system (Motion Analysis Corp.; 10 cameras) is employed. The locations of each marker given as X, Y, and Z coordinates allow for quantitative analyses in the sense that multiple 3D positions of markers can be interpreted as spatial states that develop over time. Technically, a single *pose* of one horse is defined as the 'atomic' data object of the system. A pose $P \in \mathbb{R}^{3n}$ is given by the above-mentioned geometric information of n markers. A motion *sequence* is defined as the development of a pose over time, i. e. a series of temporally consequent poses associated with one individual. The resulting motion capture sequence is a multivariate time series data set recorded with a resolution of 120 Hertz. Due to this high temporal resolution, we consider the data stream to be continuous. Since each motion is recorded within an exercise setup requiring repeated walking/trotting gait, we expect the data to contain highly cyclic behavior. In general, the absolute values of time stamps are not relevant. However, there is one exception, namely if the data are recorded for the documentation of a health recovery process of particular individuals. In this case, the absolute recording time is kept as a source of additional meta data. Additional (static) attributes also contained in the data set are provided as meta data, e. g. on the lameness status or the animal's identification. One difficulty in the characterization of the data is the heterogeneity of marker setups for different motion capture takes. While a small set of (canonical) markers is identical in every take, the domain experts additionally apply markers for testing various specific hypotheses or to perform different analytical tasks regarding local body parts. We identified this as a lack of standardization of marker configurations in the domain. To deal with the discrepancies in existing data, we decided to provide parameterizable parsers to be able to cope with a variety of data. Together with the domain experts, we mapped the marker names of different data sources to a canonical standard, which also defines the applied internal data format for individual markers. By that means, we create a basis for the comparison of data sets of different origin. This provided new possibilities to the domain experts, especially since the comparison of different motion capture measurements $\mathbf{R}_6$ was not considered as technically feasible at first $\mathbf{R}_2$.

5.3.2 Preprocessing of Data

The recording of motion capture data is an efficient process which enables the automatical reconstruction. However, we facilitate additional preprocessing steps to cope with the recommendations of the domain experts and to provide data quality (cf Gschwandtner et al. [GGAM12]). In particular, data cleaning requires some manual inspection as well as semi-automatic labeling of each data point, as tracking of hand-labels over time might fail due to irregularities in the data. Also, fixing problems such as occlusion, noise, or marker loss is mostly necessary requiring manual intervention. Occlusion and marker loss can lead to gaps in the joint trajectories which have to be filled, filtering and smoothing is necessary to deal with noise. When this is done, a mapping between data acquired this way and a basis marker setup is applied in order to get a standardized marker configuration. This allows for the comparison of data from different acquisition setups. The resulting marker setup can be seen in Figure 5.2 (left), along with two denser configurations (middle and right), both examples of typical original data mapped to the basis setup. Transforming the individual marker data into a normalized pose space is well established in the analysis of human (cf Krüger et al. [KTWZ10], Vögele et al. [VKK14]) and animal motions (cf Krüger et al. [KYZW14]). In normalized pose space a comparison of poses is possible on the basis of the Euclidean distance, since absolute position and orientation are neglected. This is why we utilize a pose normalization step which, in our case, is applied directly to the marker data. As a result of the normalization, all positions are given relative to the *Withers* marker (No. 1 of the Basis marker setup, cf. Figure 5.2). Finally, the feature vector (FV) applied in this approach is provided by rotating all marker positions around the y axis such that the *Sacrum* marker (No. 2 of the Basis marker setup) lies in the x-y-plane. In relation to the Withers, the limbs, the head and the neck are most active in equine movement. The middle of the back also has considerable up and down flexion and extension. Therefore, a marker on the withers which lies close to the center of mass in equine animals and represents the relatively rigid thorax is ideal as a system-based reference point.

5.3.3 Data Aggregation and Projection

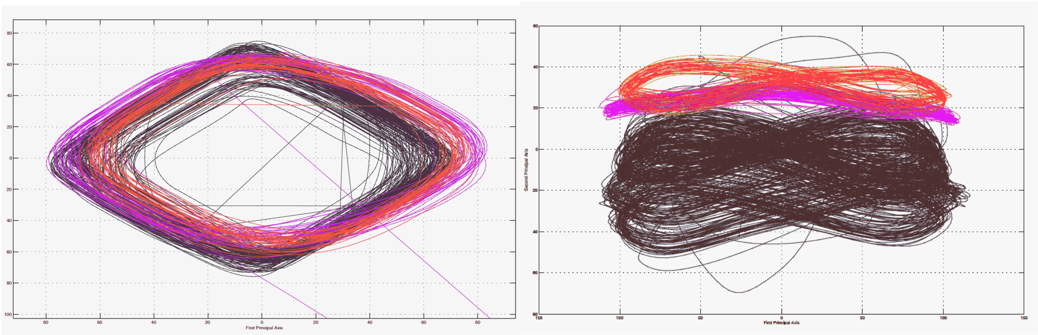
In order to make FuryExplorer scalable for large data sets, we reduce the number of objects by applying an aggregation technique $\mathbf{R}_{11}$. Simultaneously, we provide a topological order of aggregates in 2D to ensure easy visual access to varieties of available equine poses. Contrary to other systems where the distinction was made between a linear and a non-linear projection method, our system provides both projection techniques, each of them presented in a separate view. We employ the Self-organizing Maps [KSH01] algorithm (SOM) as a non-linear projection technique and the PCA [Jol86] as a linear variant (see visual prototypes in Figure 5.3). The SOM provides a data aggregation mechanism which facilitates visual overview by displaying the cluster information of every grid cell with a specific glyph renderer for equine poses $\mathbf{R}_3$ (see Section 5.4.2). Finally, we provide a data structure for the analysis of outlier patterns $\mathbf{R}_{12}$. We indicate FVs (poses) as outliers if the SOM-based vector quantization measure [KSH01] results in values worse than $n\%$ compared with all other FVs, n is a user-steerable parameter.

5.3.4 Similarity Definition and Retrieval

The definition of similarity is an important precondition for processing and comparing data. The FVs applied in FuryExplorer are the geometric positions of the joints, that form the basis marker setup, given in normalized pose space as described in Section 5.3.2. The Euclidean distances between poses are related to the more complex point cloud distance measure but allow for fast similarity searches with spatial data structures like kd-trees. Due to this faster computation, recalculating all distances after adjusting the feature set is possible at interactive frame rates. An efficient retrieval technique $\mathbf{R}_8$ is induced by the kd-tree based index structure: Searching for k nearest neighbors around a specific pose (selectable in the SOM or PCA visualization) is supported by a standard kd-tree implementation.

5.4 Visual Mappings and Interaction Design

In the preceding sections, we characterized the targeted domain and presented respective data models and algorithmic capability. We now show the visual map-



(a)

(b)

Figure 5.3: We decide to incorporate the PCA projection of equine poses. Examples of trot (a) and walk (b). The PCA projection clearly reveals the cyclic behavior of the motions. Moreover the PCA enables domain experts to distinguish between different movement classes. For example, at the lame (pink), healthy (black), and recovering (orange) motion cycles can be compared.

pings and the interaction design applied in FuryExplorer. An overview of the overall system is shown in Figure 5.5. We describe the visual encoding of a single equine pose (the atomic data object), represented as a glyph design in Section 5.4.1. The overview concept is illustrated based on two different data projection techniques in Section 5.4.2. Section 5.4.3 introduces our detail-on-demand concept consisting of a list-based equine motion sequence view. Finally, the interactive features of the system are presented in Section 5.4.4.

5.4.1 A Cluster Glyph for Equine Poses

We provide a glyph design (cf Borgo et al. [BKC^{+}13]) for the visual representation of poses and pose clusters $\mathbf{R}_4$ $\mathbf{R}_9$ (see Figure 5.4). Based on the inquiry of the domain experts, we use the raw data in the original space regardless of transformations or normalizations applied in the FV generation process. To facilitate visual comparability, we synchronize all *Withers* markers (No. 1 of the Basis marker setup, cf. Figure 5.2) and rotate all marker positions around the y-axis. The glyph renderer copes with different marker setups (and joints, respectively) depending on their definition in the FV initialization phase. As an example, the two glyphs on the left-hand side of Figure 5.4 differ in the number of neck markers. The joints between markers indicate a coarse equine skeleton. For the representation of pose clusters the glyph supports bundle rendering. By that means, information like the variance of FVs and the size can be compared visually (see e.g., the third pose in Figure 5.4). Additional information of the pose clusters is shown in the upper corners of the glyph renderer. At the top left, the cluster size is shown, while metadata distributions of clusters are displayed at the top right as a bar chart $\mathbf{R}_{10}$. That way, domain experts are able to relate pose clusters with meta information like the health status of horses or their age, etc. In the *Pose Detail View* (upper left view in Figure 5.5), the glyph renderer is applied in order to show pose details on demand $\mathbf{R}_{9}$. This Pose Detail View also allows for interactive selection of marker sets according to the analytical focus of the task at hand $\mathbf{R}_{5}$. Marker positions are colored with a highly saturated qualitative colormap as provided by the ColorBrewer tool [HB03]. As an example, the fourth pose in Figure 5.4 only contains three explicit hoof marker colors, whereas the other shown equine poses all contain a fourth hoof marker which has been highlighted red. Marker selection events automatically trigger glyph renderers, and other linked views of the system are updated simultaneously. The Pose Detail View also enables domain experts to interactively roate, scale, and translate the displayed pose renderer. An example can be seen in the fourth image of Figure 5.4 where another perspective shown. With this, domain experts are able to study horses' body configurations from any perspective. The colors of joints (between markers) are chosen with respect to a qualitative colormap with low saturation [HB03]. This enables the comparison of horses sharing the same metadata $\mathbf{R}_{10}$ or the comparison between classes of metadata, as seen in the fifth pose in Figure 5.4.

5.4.2 An Overview of Equine Poses and Motion

The central component of FuryExplorer is the *Motion Map*, which offers an overview of the data set to the domain expert. The Motion Map can be seen at the center of the Figures 5.5 and 5.6. The component consists of two main views. On the left-hand side, the SOM projection is shown including cluster glyphs for every SOM cell along with the given cluster information. Domain experts can survey the most frequent poses at a glance $\mathbf{R}_{3}$ and compare pose patterns of interest $\mathbf{R}_{6}$. The poses on the regular grid are organized in a topological order according to the intrinsic properties of the data set. With the SOM-based visual data aggregation concept FuryExplorer is able to cope with large data sets. Thus, the system becomes visually scalable to a certain extent $\mathbf{R}_{11}$. Aside from the analysis of static poses, the view also enables domain experts to analyze motions $\mathbf{R}_{4}$. The simultaneous display of different data granularities (single FVs and FV clusters) is facilitated by applying a micro-macro level analysis (cf Bernard et al. [BvLBS09]. Motions of different horses are shown as colored path lines (trajectories). By that means, domain experts can visually compare different motion behaviors $\mathbf{R}_{6}$. Contrary to most existing tools in the domain, FuryExplorer supports the comparison of different horses in a single view. The trajectories consist of temporally consequent poses of a motion sequence. Single poses are projected to continuous

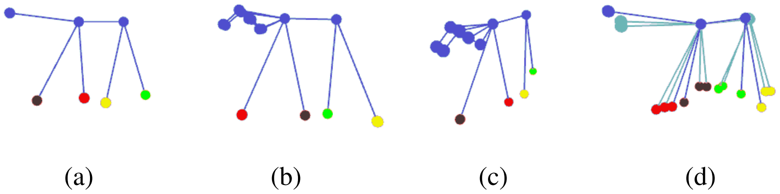

(a) (b) (c) (d)

Figure 5.4: Glyph designs for equine poses. The visual representation provided is able to show single poses and pose bundles, respectively. Selectable marker colors are sensitive to a global color linking concept triggered by user interaction. The glyph is rendered in 3D and thus applicable for affine view transformations based on user interactions.

display coordinates b y utilizing the discrete SOM grid as points of a mesh for interpolation (cf Bernard et al. [BVLBS11]). The results of the PCA projection are shown on the right-hand side of the Motion Map. Again, the trajectories of complete motions are present. By that means, FuryExplorer provides two complementary views to show the pose development within a motion based on different projection techniques. The SOM combines the analysis of most frequent static poses with the analysis of motions. The PCA projection facilitates the analysis of motion as well as the indication of outlier poses. The outlier poses are visually highlighted with circular outlines, as shown in Figure 5.6. A vertical slider between the SOM and the PCA enables domain experts to steer the number n of shown outlier poses. A simple left click on single poses or pose clusters shows respective data in the Pose Detail View. By that means frequent patterns (SOM) but also rather unexpected poses (PCA) $\mathbf{R}_{12}$ can be explored in detail.

5.4.3 Sequence Detail List View

We provide a list-based view to show (sub-)sets of motion sequences in more detail $\mathbf{R}_{9}$ (at the bottom of Figures 5.5 and 5.6). This *Sequence Detail List View* supports information drill-down tasks increasing the granularity of motion sequences and enables their comparison $\mathbf{R}_{6}$. Different interactive means to generate such sets of subsequences are available (see Section 5.4.4). Every element of the list shows the temporal development of marker positions of a single motion. The Pose Detail View provides the domain experts with an option to chose a set of color-coded markers. Based on an inquiry of the domain experts, FuryExplorer offers two different views of the marker development over time: Per default, marker progressions are shown from a lateral perspective. Technically, this means the y-axis of every marker is shown as a linechart over time. But the perspective can

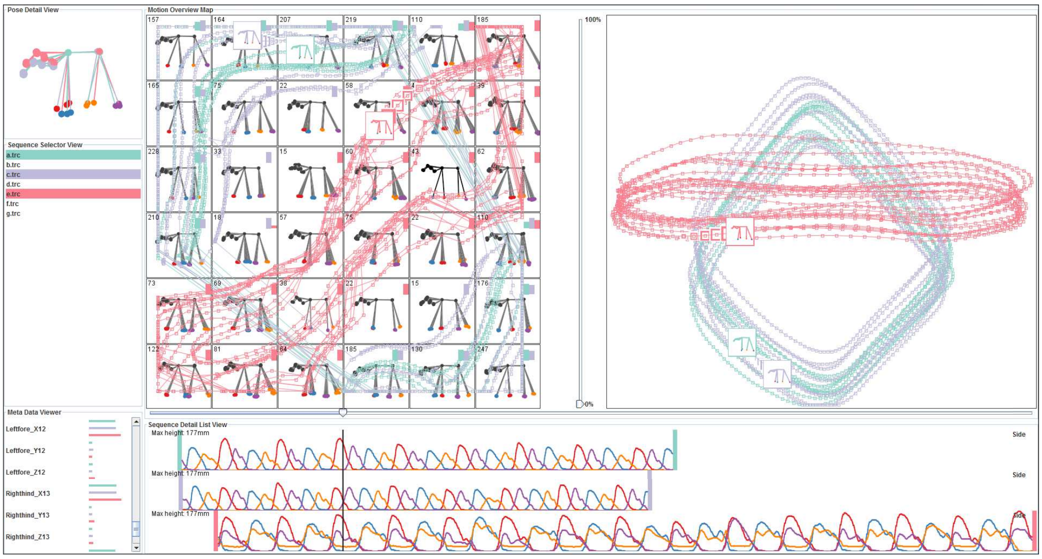


Figure 5.5: The system for use case 5.5.1. Here, the domain experts compare different gaits (walking and trotting). Differences in the two gaits can be identified at three different levels of granularity: a) in the *movement* trajectories, b) in the *pose* distributions, and c) *marker* motions.

also be changed to a virtual camera position directly above the horse. This facilitates tracing the z-axis of the markers over time. Another inquiry of the domain experts concerned the comparability of different equine motion sequences. This is why provide two different adjustment modes to make sequences comparable. Firstly, domain experts are enabled to define temporal offsets for every sequence to synchronize the temporal domain. Secondly, domain experts can re-scale single sequences in order to compare different speeds in terms of their cyclic behavior, as can be seen in Figure 5.5.

5.4.4 Interaction Design

We briefly describe the most relevant interactive features of FuryExplorer with respect to the list of requirements. We use color-coding to emphasize linking between different views to connect marker information and selections of pose/gait groupings. The application of the qualitative colormap described earlier enables looking up respective elements in multiple views. On the left-hand side of the system view window, a list of available horses is shown. We shall refer to this list as the *Sequence Selection View*. Any sequence selection event triggers all views to adjust the color-coding of all relevant information. For example in the Motion Map, the colors of each trajectory are updated for all selected sequences. The (multi-)selection of equine sequences is one of the information drill-down functionalities provided by FuryExplorer $\mathbf{R}_6$. In particular, a selection-change event triggers the Sequence Detail List View to refresh the list of visible elements $\mathbf{R}_9$. In order to explore hidden relations between the data content (poses/gaits) and the metadata (like horses' names, health/lameness status, or variation of hoof motion, cf Bernard et al. [BSW+14, BRS+12]), the *Meta Data Viewer* lists available metadata attributes, as can be seen on the right-hand side of Figure 5.4.

In this example, variation within the movement of the feet of four different horses is illustrated by colored bar charts. The green candidate shows the highest variations. The selection of attributes from the list of available metadata classifies corresponding pose/gait visualizations by assigning colors. This enables the analysis of relations between e.g., the health status of horses with their gait performance. In addition to the selection options, FuryExplorer provides query-by-example functionality $\mathbf{R}_7$. Whenever a domain expert right-clicks a pose, the system offers nearest-neighbor retrieval based on the algorithm described in Section 5.3.4. Again, the Sequence Detail List View shows the retrieved results. Finally, we provide an interactive control to enable navigation through the temporal domain of equine motion sequences. By that means, details of selected sequences can be analyzed on-demand. Inspired by visual representations introduced as part of related approaches (cf Robertson et al. [RFF+08], Bernard et al. [BWS+12], we present a slider control which facilitates sequence animation. By dragging the slider, the two projection views show the positions as they develop over the course of time. This is done with respect to the pattern generated by all trajectories. We call this the 'Rollercoaster Animation' (see e.g., Figure 5.5). The current slider position is also indicated by a vertical bar in the Sequence Detail List View. Thus, domain experts can visually compare different sequence animations and respective marker properties easily. The Rollercoaster Animation also allows for the analysis of outlier poses. Outlier poses are highlighted by a gray shadow in the Sequence Detail List View. They can be examined more closely by dragging the slider, as it was illustrated in Figure 5.6.

5.5 Use Cases

We aim to show the applicability of FuryExplorer. Together with the domain experts we have worked out two analytical workflows based on important analysis tasks, as described in the domain characterization section.

5.5.1 Analysis of Different Gaits

In the first use case, we perform a comparison task between different gaits. In Figure 5.5, different input sequences to the system are shown. Three of the seven motion sequences were selected in the Sequence Selector and color-coded emerald green, light purple and pale red. It was interesting for the domain experts to see the different trajectory patterns of the three color-coded motions in a comparative manner. In the PCA-based projection to the right of the Motion View, it can be seen that the green and the light purple motion sequence are predominantly similar. Both sequences follow a cyclic four-step pattern, their trajectories are diamond-shaped. Contrarily, the red trajectory shows a cyclic two-step motion which mainly evolves around the horizontal axis. It was easy for the domain experts to distinguish between these two patterns. Moreover, the domain experts were able to interpret the four-step motion pattern as a typical *walk*, whereas the two-step motion pattern corresponds to a *trot*.

In order to take a closer look at the poses presented in the SOM-based projection, the four hoof markers were selected. This is done in the Pose Detail View. The set of markers is coded in blue, orange, red, and purple, respectively. On the left-hand side of the SOM, the cells show equine poses where the red marker is located in front of the blue marker. On the right-hand side of the SOM, the blue marker is positioned in front of the red marker. There is a recurring pattern in the vertical distribution of horses' body configurations on the SOM grid. Here, the back hooves of the horses can be identified. While the upper half of the poses show the orange marker in front of the purple marker, the hoof configuration is reverse at the bottom of the SOM. Thus, the SOM distributes the available body configurations into four different quadrants. Again, the red two-step motion can be identified (as a diagonal). This brings the insight that there is a classification of the gait with respect to the pose distributions by the arrangement in the SOM. The trajectories of the other four-step motions create a more circular SOM pattern.

In order to investigate the hoof markers of the three sequences in the Sequence Detail List View, we shift and scale the three sequences interactively to increase comparability. It can be seen that the two upper sequences have similar hoof motion, whereas the pale red sequence at the bottom is quite different. One of the main differences between the two types of gaits is that, in the red sequence, one pair of markers (red and purple) show similar behavior, while the remaining two markers (blue and orange) are similarly as well, whereas in the other two sequences, each hoof motion has a different pattern. The domain experts recognize this as the precise distinction between the already identified walking and a trotting gaits. In the trot, the left fore and the right hind (and the remaining pair of hoofs, respectively) follow the same temporal motion cycle. However, in the walking gait, every hoof has its own timing. The Sequence Detail List View also enables

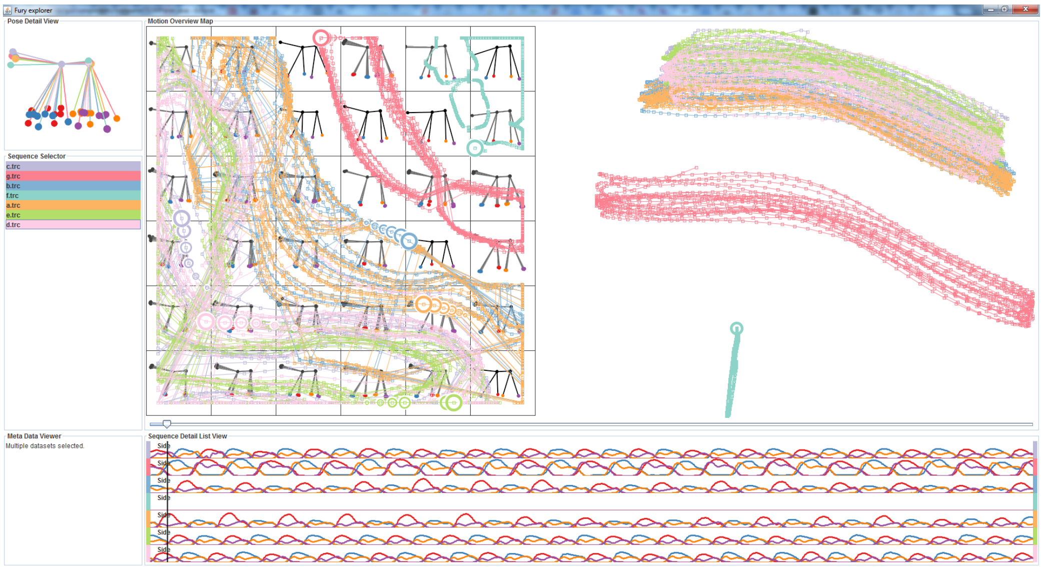


Figure 5.6: The system of use case 5.5.2. Domain experts analyze equine motions distinguished by the meta data attribute *health status*. The trotting lame horse (red) performs significantly different compared to the trotting healthy (green, orange, pink) or standing healthy (green) horses.

the domain experts to analyze micro variations in identical sequence cycles. The animation slider (Figure 5.5 in the Sequence Detail View) was placed at a position where the red hoof marker has reached the highest peak in all three sequences. While focusing on the red sequence, the domain experts identify an unexpected ‘double swing’ when the markers decrease from the highest peaks until the hoof is placed to the ground. Finally, we compare the three gait sequences in the Meta Data Viewer at the bottom left. We see that the two-step gait (pale red) has the highest variance compared to the two four-step gaits. These findings were another interesting novelty to the domain experts.

5.5.2 Lameness

In the second use case, we reorganize the analytical reasoning workflow in the following sense. Before, the use case had started with the interpretation of poses and motion (the content of the motion capture data). In this case, we start with one meta data attribute, the *health status*, in order to look up *lame* and *healthy* horses. We aim to relate the according poses and sequences provided by the system to respective meta data tags. We start with the selection of the meta data attribute *health status* in the Meta Data Viewer (see Figure 5.6 at the lower left). This results in a partition of all available sequences into two different color-coded classes.

Healthy horses are coded emerald green and lame horses are colored purple according to the indication in the Meta Data Viewer. The interesting question was to what extent the content-based views would discriminate horse motions with respect to the selected metadata class label. In the Sequence Selector View, it can be determined that three of the four horses under examination are healthy, while one horse is lame. Both in the PCA and in the SOM, we identify a separation between three different trajectory areas without overlap. The trajectory of the lame horse (purple) is located in between two healthy trajectory areas. We investigate the poses provided in the SOM display in order to investigate this finding. The domain experts identify that, in the upper right area of the SOM, an equine pose is shown with its head down, which is unique among all pose patterns visualized by the SOM. Based on that finding, the domain experts were able to relate the different map locations to different gaits: the trajectory located top right of the SOM is associated with a stance, whereas the motions shown at the center of the SOM are trotting gaits. Thus, we are able to distinguish between a healthy horse standing, a lame horse trotting, and two healthy horses also showing trotting gaits. Although it would need a greater number of equine motion data for generalization, the domain experts are impressed by the capability of the system allowing for a rough visual classification of different gait and lameness conditions.

To be able to compare all four sequences, we shift and scale the them in the Sequence Detail View for synchronization. This enables the application of the Rollercoaster Animation. The poses of all four motions at the current slider position can be compared in the Pose Detail View (upper right-hand side) by rotating the pose bundle glyph. Again, we identify that one of the four shown poses has a lowered head. It can also be observed that the lame horse (bottom of Sequence Detail List View) has a different marker motion cycle. Interestingly, the markers of the lame horse show higher variance throughout the motion. In order to prove for this hypothesis, we employ the Meta Data Viewer to isolate those green bar charts (right-hand side of Figure 5.4) showing the highest variance. In fact, this isolates exactly the data from the lame horse. This does not coincide with the expectations of the domain experts and offers an interesting new perspective for more analyses.

Finally, we would also like to detect possible outlier patterns. Dragging the vertical slider between the SOM and the PCA and results in the display of pose outliers in the SOM and the PCA, respectively. Such outliers are indicated by circle icons at the positions of the outlier poses in the motion trajectories. The pose outliers can be enhanced for closer analysis by using the Rollercoaster animation slider. It can be seen that many outliers appear at similar locations within the circular path of all trajectories. Together with the domain experts we hypothesize that the occurrence of outliers is an indication of different speed conditions in the gait performances. The higher the speed, the lower the number of similar poses within the respective vector space which increases the probability of detecting outliers in these regions. This hypothesis is supported by the fact that the number of outliers is considerably higher for trot sequences of healthy individuals where faster motion dynamics can be expected. At any rate, this is a good starting point to introduce further improvement to the system in the sense of supporting additional analysis tasks, which are beyond the requirements for this work though. This proposes an interesting research question to be discussed in future work.

5.6 User Study

We conducted a user study to demonstrate the usability of the visual design and the interactive capability of FuryExplorer. In particular, we would like to ascertain if the system supports the domain experts in their research. The insight-based summative user study was conducted with 8 domain experts of veterinary medicine at their lab. All participants were female, seven of them between the age of 25 and 34, one participant was between 45 and 54 years old. The educational level of five participants was a Master‘s degree, while three held a PhD or higher. All members of the subject group stated that they actively work with horses and are also able equestrians. Moreover, all participants work in equine research and are familiar with all particular animals that are present in the data. Six of eight respondents stated to be familiar with equine motion analysis.

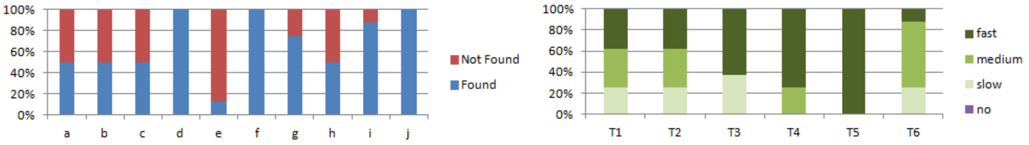
Test Environment In order to attest the success of our design, we created different tasks for the participants to solve. All of them were based on the requirements of the domain characterization phase (see Section 5.2.2). A complete list is found in Table 5.2. Tasks $\mathbf{T}_1$ - $\mathbf{T}_3$ were based on a first use case where the focus was on recognizing different gaits (cf. our first use case introduced in Section 5.5). Tasks $\mathbf{T}_4$ - $\mathbf{T}_6$ relate to use case 2 (Section 5.5.2). Tasks $\mathbf{T}_1$ and $\mathbf{T}_4$ are very similar, which allows to evaluate the degree of familiarization of the participants with the system in the course of the user study.

Test Execution In the beginning of the study, we introduced the participants to the overall goal of the FuryExplorer project, and to the analytical tasks it is designed to support them with. We asked the participants about their expertise in veterinary medicine and their specific research interests. In the first phase, we wanted to attest the usability of the system. To that end, we asked the participants to do a field test with FuryExplorer without prior introduction. An observation of this familiarization phase established the ‘intuitiveness’ of FuryExplorer.

We assessed the usability by observing the exploitation of the following ten features: a) if the glyph was rotated, b) if the glyph was zoomed, c) if the glyph was translated, d) if a single selection was made in the Sequence Detail List View, e) if multiple selections were made in the Sequence Detail List View, f) if the list of meta data was scrolled in the Meta Data Viewer, g) if the Rollercoaster animation slider was applied, h) if the topology of equine poses was identified in the SOM, i) if path trajectories were identified in the SOM or the PCA, and j) if the marker of the current animation pose was identified in the projections. As a next step, we gave a short introduction of FuryExplorer, in case the participants overlooked specific features of the system. Subsequently, the participants were able to attempt the six pre-defined tasks. We observed the task-completion time and aggregated the quantitative results to four qualitative intervals ranging from ‘fast’ (within few seconds), to ‘medium’ (lasting about 20 seconds’), to ‘slow’ (with a duration up to one minute), to ‘no’ (no success). After the participants executed the tasks, we concluded each study with an interview regarding the system, the performed tasks, and other possibly relevant issues.

Results of the Usability Observation Phase Figure 5.7 (a) shows to what extent the ten visual-interactive capabilities of FuryExplorer (a-j) were identified and successfully applied by the participants without prior introduction or assistance. Half of all participants figured out the interactive glyph transformations (a-c). The selection of single sequences (d) was identified by all participants. The multi-selection of sequences (e) got the weakest ‘intuitiveness score’, which would have been accessed by pressing the CTRL key in combination with the primary mouse button. Here, a tooltip functionality might have helped discover this interactive functionality without supervision. Scrolling the list of meta data (f) was tested by all participants. The Rollercoaster slider (g) was identified by roughly 80% of the group. As known from related work, the identification of the potentially beneficial topology of the SOM (h) only partially succeeded. Making projections more self-exploratory in general is a challenge of future research. However, the identification of trajectories in the projections (i) worked almost perfectly. Finally, the selection of the current animated pose in the projections (j) was identified by all participants. We conclude the usability section with the lesson learned that rather complex interactive means need to be brought to attention more explicitly. However, the majority of the participants was able to use FuryExplorer after only a few minutes and even without supervision. Given that we included sophisticated analytical capability in a visual-interactive system which was entirely new to the domain experts, we are satisfied with the results of the usability phase.

Results of the Task-Based Assessment of the Usefulness The results of the second phase show that all participants were able to complete each task (see Figure 5.7 (b)), we did not have to interrupt a single session. Even though, we identify



(a) (b)

Figure 5.7: Results: (a) Identification of interactions by the participants without instruction (left). (b) Task-completion times (right).

different task completion times, indicated by different brightness of green in the stacked bar charts. $\mathbf{T}_1$ and $\mathbf{T}_2$ were solved with identical performance. Two of six participants took more time for the identification of the two-phase and the four-phase gait, and for the relation of the gaits to their domain knowledge (trotting and walking). The performance of $\mathbf{T}_3$ showed a dichotomy. While five attendees succeeded fast, three of them performed rather slow when utilizing the Pose Detail View. However, $\mathbf{T}_4$, i. e. the repetition of $\mathbf{T}_1$, was performed more quickly than $\mathbf{T}_1$ which affirms the familiarization of the participants with FuryExplorer in the initial test. Similarly, all participants easily identified the outlier by comparing different motions ($\mathbf{T}_5$). However, $\mathbf{T}_6$ was performed slower than all other tasks. The problem was that the lame horse (out of the seven horses in the system) was difficult to discern. Only the micro variances in the pose topology of the SOM revealed the differences of the lame horse's motion in comparison to the healthy horse's. However, since this task was rather difficult, we are quite confident that FuryExplorer enabled all domain experts to reason about distinct body configurations and thus, about the health status of horses. Considering the results of the usefulness phase, we can state that all eight domain experts were able to perform analytical tasks with their equine motion capture data for the first time. The task success rate of 100% for all tasks and all participants is quite convincing.

5.7 Discussion and Future Work

Discussion Before we outline options for future work based on the presented system, we discuss some limiting factors. One issue concerning the employed (SOM, PCA) projection methods is that they do not naturally coincide with the intuition users may have of the represented information. Consequently, some users in our study falsely interpreted the given scatterplots as geometric marker information and found solving the tasks harder. Since the gap between intuition and visual representations of projection results is a common phenomenon, it may be subject of further research in general. In particular, ways to close the gap between these two aspects need to be more thoroughly explored. Another matter of concern is scalability. We already provide visual scalability based on the projection and the aggregation techniques employed by FuryExplorer. Currently, the number of equine motion sequences of about ten yields an overall number of 30,000 poses, each containing 21 time-oriented dimensions.

We handle this 600.000 values in real-time (refresh $<<\ 100ms$). However, it would be interesting to also get an algorithmic perspective on the way FuryExplorer will scale for larger data sets.

Future Work Apart from the above mentioned matters, the major focus of future work will be on emerging data acquisition techniques. Relatively novel techniques or such ones that are not yet fully established will expand the input base for FuryExplorer. Only some of the many data types come from gyroscopes, accelerometers, EMG, video and pressure data as well as inertial sensors. The above mentioned visual feedback would also be an interesting item on the agenda for such data acquisition pipelines, as would a transfer of search and retrieval techniques. The system is very well suited to train a (visual) classifier identifying gait abnormality. Therefore, institutions such as university clinics could use information from all the previously vetted cases on any new patient. Also, the subjective impression of medical experts could then be stored and used for comparison with other cases,or with the same case at a later stage, after surgery or against offspring. Throughout the user study, the domain experts proposed a number of additional ideas which may help further improve the system. One of them is a 'Play' button to show coherent animation of equine gaits in real time. Another is having additional view options for the comparison of micro variations of two particular gaits. Furthermore, a highly important feature is being able to feed-in and provenance information to the existing data, e.g., in combination with a visual visual classifier. This classifier could work for a number of different use cases, e.g., to find out if a prospective investment will actually turn out to be a successful competitor or breeding animal. Finally, future work will be to carry out the FuryExplorer 'framework' for other multivariate time series data. It will be interesting to see how architectural design choices to provide generalizability will perform for other application domains with other data, users, and tasks.

5.8 Conclusion

We presented FuryExplorer, a visual-interactive system for the exploratory analysis of equine motion capture data as a special type of multivariate time-oriented data. This system provides an overview of given poses and motions of horses based on two projection techniques, namely the self-organizing map (SOM) and the PCA. From a list of available motion capture sequences, domain experts can select files to be analyzed in detail. A sequence detail list view allows for an alignment of specific motions and for the comparison of respective marker movement over time. To this end, domain experts can apply a slider control in order to interactively browse the synchronized equine motion sequences. We showed the applicability of FuryExplorer in two real-world use cases. One was the comparison of different gaits (trotting and walking) and another was the analysis of diverging locomotive behavior in lame horses compared to healthy individuals. We evaluated the system by a user study with domain experts in the field of veterinary medicine. The results show that FuryExplorer was intuitively usable by the majority of the participants. Moreover, the domain experts confirm the high potential of FuryExplorer enhancing efficiency and effectiveness of performing daily routines in the field of expertise. Analysis results within the study also presented new aspects of the data that had not been taken into consideration or had not been visually communicated before. For the first time, the domain experts were able to visually *compare* the gaits of different horses, or of identical horses recorded at different times. This opens new analytical opportunities like quantifying irregularities in the locomotor system (for biomechanics), or keeping track of medical progress (for veterinarians) - which both has been nearly impossible.

Presenting FuryExplorer, we applied general visualization techniques to the domain of equine motion analyses. We are confident, that the system would work directly on motion data of other quadruped species after defining a suitable motion capture marker set and the according pose normalization. The techniques employed for pose normalization and similarity search were adapted from human motion analysis. Thus, the presented approaches are capable handling human motion data as well. This creates an additional opportunity for comparison of pathological gait patterns.

Table 5.1: Results of the technical requirement analysis process in the domain characterization phase.

Req.	Requirement Description
$\mathbf{R}_1$	*Ease of use*: The system must be as intuitive as possible. Otherwise veterinarians and other professionals will not use it.
$\mathbf{R}_2$	*Making different data sets comparable*: Tackle challenge of heterogeneous data sources.
$\mathbf{R}_3$	*Overview*: Providing visual access to large data sets giving a visual overview of the complete data is crucial.
$\mathbf{R}_4$	*Poses and motions*: Domain experts wish to focus their attention on the analysis of poses but also want options to investigate entire motions. The system should provide means to pursue both demand-actuated.
$\mathbf{R}_5$	*Marker selection*: Domain experts do not want to restrict the set of relevant markers to a single configuration. They would rather specify relevant markers interactively.
$\mathbf{R}_6$	*Comparison of cohorts*: It should be interactively possible to localize and define subsets of the data. Such sets need to be visually highlighted thus allowing for the comparison and the exploration of details.
$\mathbf{R}_7$	*Visual querying*: Domain experts want to search for relevant horse motion (sub-)sequences. An intuitive way to specify a query will facilitate effective retrieval. The visual-interactive specification of queries based on example poses should be supported.
$\mathbf{R}_8$	*Retrieval*: To search for horse motion sequences a retrieval algorithm is needed. The motion retrieval research domain has introduced a variety of approaches, one of these needs to be selected and if necessary, adapted to the targeted domain
$\mathbf{R}_9$	*Details*: The selection of single poses and motions of individual specimen should be possible. Specific properties and details of interest should be highlighted on demand. Entire motion sets could be interactively controlled by an animation slider.
$\mathbf{R}_{10}$	*Metadata relations*: Many datasets contain metadata like the age or the health status of horses. Relating this labeled information about horses to the horse poses/motions is of great interest to domain experts.
$\mathbf{R}_{11}$	*Scalability*: Ideally, user interactions with the system should be processed in real-time. Moreover, it would be nice if the tool ran on a conventional notebook for the use of the tool, e.g., at the recording environment.
$\mathbf{R}_{12}$	*Outliers*: ‘Everything may be interesting’ was a common remark of the domain experts. Thus, not only frequent patterns, but also ‘unique’ patterns should be part of the analytical task repertoire.

Table 5.2: Tasks performed in the user study.

Task	**Task Type [AA06]**	**Task Description**	**Requirements Covered**
$\mathbf{T}_1$	Lookup/Localization	How many gaits can you see?	$\mathbf{R}_1$ $\mathbf{R}_2$ $\mathbf{R}_3$ $\mathbf{R}_4$ $\mathbf{R}_6$ $\mathbf{R}_9$
$\mathbf{T}_2$	Identification	Can you figure out which gaits are shown?	$\mathbf{R}_4$ $\mathbf{R}_9$ $\mathbf{R}_{10}$
$\mathbf{T}_3$	Comparison	Can you show me the differences between the different gaits?	$\mathbf{R}_2$ $\mathbf{R}_4$ $\mathbf{R}_6$ $\mathbf{R}_9$
$\mathbf{T}_4$	Lookup/Localization	How many gaits can you see?	$\mathbf{R}_1$ $\mathbf{R}_2$ $\mathbf{R}_3$ $\mathbf{R}_4$ $\mathbf{R}_6$ $\mathbf{R}_9$ $\mathbf{R}_{10}$
$\mathbf{T}_5$	Comparison	Can you show me the outlier?	$\mathbf{R}_2$ $\mathbf{R}_3$ $\mathbf{R}_6$, $\mathbf{R}_{12}$
$\mathbf{T}_6$	Localization	Can you show me motions of lame horses?	$\mathbf{R}_4$ $\mathbf{R}_9$ $\mathbf{R}_{10}$

CHAPTER 6

LEARNING MODELS OF EQUINE MUSCLE ACTIVATION

> A horse never runs so fast as when he has other horses to catch up and outpace.
>
> Ovid

This chapter introduces fitting Gaussian mixture models (GMMs) to surface EMG data (sEMG) in order to interpret animal muscle activation during locomotion of quadrupedal subjects (equine). Activation patterns depending on recorded gaits (walk, trot) as well as on different muscle location (m. longissimus dorsi, m. gluteus medius) could be isolated by analysis of cycles of activity within locomotion. We found that in an otherwise homogenous group of 14 individuals, there was high variation in sEMG data both on an intra-subject as well as inter-subject level, i. e. activation patterns may vary across different muscle location, gait and individual subject. In order to provide new opportunities for analysis and exploration of sEMG data. we point out opportunities arising from exploration based on hierarchical clustering.

6.1 Introduction

The paper at hand introduces the application of a statistical method to the investigation of animal muscle activity during locomotion which is a common application in the field of biomechanics. The statistical method we propose focuses on the detection of clustered peaks in signals captured by surface electromyography (sEMG) and estimation of multi-modal Gaussian models to describe these data. We specifically selected examples of two different muscles in equine subjects in order to demonstrate what insight can be gained by modeling sEMG by Gaussian mixture models (GMMs).

Over the last decade, monitoring and analysis of muscle activity during locomotion has gained increasing popularity. Novel wireless technology has made recording such data much easier. Consequently, the processing of sEMG signals has become the focus of attention. In the field of animal biomechanics, different data processing techniques are used creating several difficulties when trying to compare the results of different studies (Boudaoud et al. [BAM10], Olsen et al. [OHAP12] Williams et al. [WJB$^+$14]).

The data sets we studied originate from a group of horses of the same breed (Haflinger) and the same sex (mare). Two larger muscles were investigated, the m. longissimus dorsi (trunk muscle) and the m. gluteus medius (leg muscle). Muscle activation was recorded by sEMG at one location on the m longissimus dorsi and at three different locations of the m. gluteus medius. The m. longissimus dorsi (long back muscle) is one of the most commonly investigated equine muscles (E. g. Cottriall et al. [CRW08], Licka et al. [LFP09]). This large surface muscle of the back is an ideal candidate for the investigation of spinal stabilization due to its function of extending the back (during bilateral contraction) and splinting the back against passive deformation (during uni- and bilateral contraction). Besides trunk muscles, limb muscles are also often investigated (cf. Zaneb et al.[ZKS$^+$09], Crook et al. [CCM$^+$10], [SGW13] , Williams et al. [WGRP13], as they directly influence efficient locomotion which is an important area of research.

Therefore, the overall aim of the study was to characterize equine trunk and limb muscle activation during different gaits.

Besides analyzing general characteristics of sEMG signals during full trials of locomotion, the analysis of sEMG peak behavior during individual cycles of motion yields valuable insight. We think that peak patterns are especially important because they could generate more compact models of information. Therefore, one of our objectives is looking for a structure of peaks within cycles of muscle activity (gait cycles of locomotion). In particular, we wanted to identify if there is an answer to the question by what number of peaks sEMG signals can be modeled for each of the investigated muscles. One interesting finding was that the data sets we investigated were much more heterogenous than expected. The amount of individual variation between peak locations, number or peaks and their height exceeded our expectations. Nevertheless, we were able to identify a suitable number of peaks to expect on an average level.

It can be expected that any information gained in these experiments depend on the gait during which sEMG was recorded. In order to be able to detect this dependency, an equal number of trials of walk and trot was captured. Based on the muscle activation patterns identified in the tested group we will compare the results of both gaits hoping to visualize the extend of dependency.

We investigate leg muscle activation at different locations of the m. gluteus medius because there is known to be quite some fiber-variation across its different sections (as described by Bruce [BT85] in the m. gluteus medius of thoroughbreds). We are hoping to find out if this variation has any significant effect on muscle activation between the three parts of the muscle monitored in the experiments. A similar intra-muscular pattern was previously identified in the m. longissimus dorsi where activity was measured at three different positions on the same muscle (cf Licka et al. [LPF04, LFP09]).

Empirical mode-seeking or fitting GMMs has, to our knowledge, never been applied in the investigation of biomechanics of animal movements. Fitting GMMs has applications in a variety of fields from signal processing to pattern recognition (cf Bishop [Bis06]). In particular, the method can be applied to classification-related problems as in human action recognition based on sEMG data (cf. Ju et al. [JOWKL13], Ding et al. [DHZC]).

One important goal is providing new opportunities for exploration of sEMG data. Given the heterogenity of the data at hand, we point out opportunities arising from exploration by hierarchical clustering. This helps restructure the data, (cf Bernard et al. [BWK$^+$13], Wilhelm et al. [WVZ$^+$15]). For the first time, this allows experts in biomechanics to detect subgroups based on sEMG data. We show how clustering complete multivariate time series data (Rodrigues et al. [RGP08], Rani and Sikka [RS12]) helps discover relations between the complete group and subgroups of individual with respect to average, typicality and anomaly.

6.2 Methods

6.2.1 Horses

Fourteen horses without clinical sign of back pain and lameness were used in this study (14 Haflingers mares, mean age was 8 ± 3 years, CI (6, 9), range 4-14 years; mean body mass was 463 ± 42 kg, CI (439, 487), range 396-526 kg; mean height at the withers was 131 ± 5 cm, CI (128, 134), range 125-145 cm.

Longissimus Dorsi

The equine m. longissimus dorsi is the longest major back muscle of the horse, which main function is spinal stabilization. At the level of the 16th thoracic vertebra parallel to the muscle fibre direction - where the maximum lateral movement of the spine occurs in walk and trot - sEMG electrodes were placed over both the left (LDL) and right (LDR) m. longissimus dorsi (cf. Licka et al. [LPF04] Wakeling et al. [WRDN07] Groesel et al. [GZK$^+$10]).

Gluteus Medius

The equine m. gluteus medius (GM) is the largest unsegmented muscle of the horse. Its main function is to extend the hip and retract and abduct the limb. The muscle provides the visible shape of the croup, and due to its location directly under the skin, it is an ideal candidate for detecting intra-muscular activation differences with sEMG. Three electrodes were placed over the left and right GM parallel to the muscle fibre direction and at the midpoint between origin and insertion about 5cm apart on the medial (GM1), middle (GM2), and lateral (GM3) part of the GM. For the rest of the paper, *GML1* will refer to the left-side medial part and, similarly, *GMR1* will refer to the right-side medial part of the GM for abbreviation.

6.2.2 Data Acquisition

For this study, a set of three-dimensional kinematic data and surface electromyography (sEMG) in walk and trot were synchronously collected.

EMG

Surface electromyography measurements were taken in the above mentioned 14 horses without lameness, walking and trotting on a treadmill. The resulting EMG signal was full-wave rectified and sampling rate was reduced to 120 Hz. A Butterworth low-pass filter was applied (fourth order; cut-off frequency, 20 Hz). For each horse three trials or more were captured per gait each of them 10 seconds in walk and in trot.

Measurements by a set of 8 trunk-mounted sEMG sensors of a Delsys Trigno Wireless System (Boston, MA, USA). The sensor units were placed at two different locations on the equine body. Horses were shaved in the region of the muscles on the skin over the *m. longissimus dorsi* and *m. gluteus medius* and electrodes for sEMG recording were applied to the skin. One sensor was placed on each side of the spine over the m. longissimus dorsi, three sensors were placed on each side of the spine over different sections of the m. gluteus medius. The recordings resulted in one data file per trial run. Recordings of 10 seconds of trot and walk respectively of each of the animals. Recorded frame rate: 4000 frames per second. Horses were exercised on a treadmill performing two different gaits, walk and trot. Each trial run was repeated 3 times in a row per horse and gait.

The study followed the guidelines laid out in the works of Luca [DL97], Hermens [Her00], and Merletti [Mer15]. Note that there is not a standard work on guidelines for equine (or any animal) sEMG data processing and reporting so far.

Kinematics Data

Additionally, during each trial, one stream of kinematics data was captured. The motion of the left fore was recorded by optical Motion Capture (10 high speed cameras; Eagle Digital Real Time System, Motion Analysis Corp., Santa Rosa, CA, USA) simultaneously to the sEMG recordings. In particular, the kinematics data of the left fore reflect the same motion patterns as the EMG data.

For kinematic measurements seven reflective skin markers were positioned on each horse using adhesive tape; one on the forehead, one on the highest point of the withers, on the sacrum were placed and on the lateral side of each hoof to identify motion cycles. Horses were warmed up and accustomed to the experimental set up on the treadmill. Three-dimensional kinematic data in walk and trot were collected using ten high-speed cameras recording at 120 Hz.

6.2.3 Data Processing

EMG Data

Consider a data signal M in the set of recorded data describing activation of one muscle. M is a discrete function $M = (f_1, \dots, f_n)$ of length $n \in \mathbb{N}$ in time where each $f_i, 1 \leq i \leq n$ encodes the amount of muscle activation at time stamp i.

The EMG signal is rectified (full wave rectification) in the sense that

$$M_{\text{rect}} = (|f_1|, \dots, |f_n|) \,. \tag{6.1}$$

The sampling rate of the raw data sequence was reduced to to 120 Hz by a factor of $\frac{120}{4000} = 0.03$ while at the same time applying a lowpass filter to prevent aliasing.

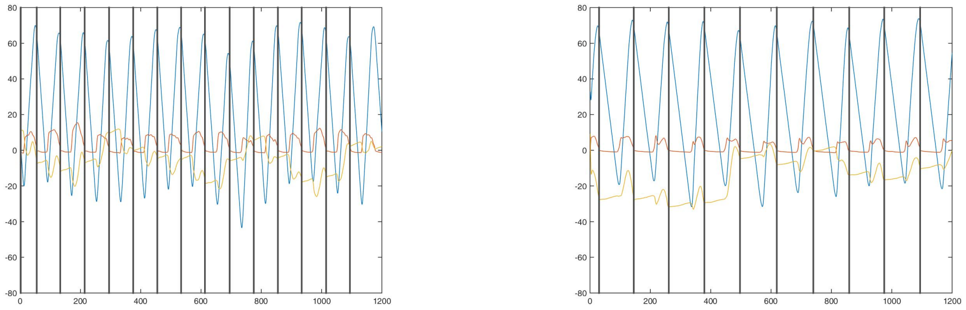
Filtering by a fourth order Butterworth low-pass filter (cut-off frequency: 20 Hz).

All EMG signals were scaled to have a range within the interval $[0, 1]$ and to have a number of 100 frames [LPF04]. The first is a normalization in value space, the latter a normalization in time.

Kinematics Data

Kinematics data are cleaned and interpolated two remove temporal gaps in order to keep track of the movement of the left fore limb.

Three-dimensional coordinates of each marker during the time course of each experiment were calculated from the data using kinematic software. These time series were then smoothed by use of a Butterworth low-pass filtered (cut-off frequency, 10 Hz). The data was split to motion cycles starting with the stance phase of the left fore automatically. We also calculated characteristics of each motion



(a)

(b)

Figure 6.1: Kinematics information on left fore in both trot (a) and walk (b). The orange line illustrates the motion in the y-axis, i. e. the ground contacts occur at the beginning of flatter episodes of this orange curve.

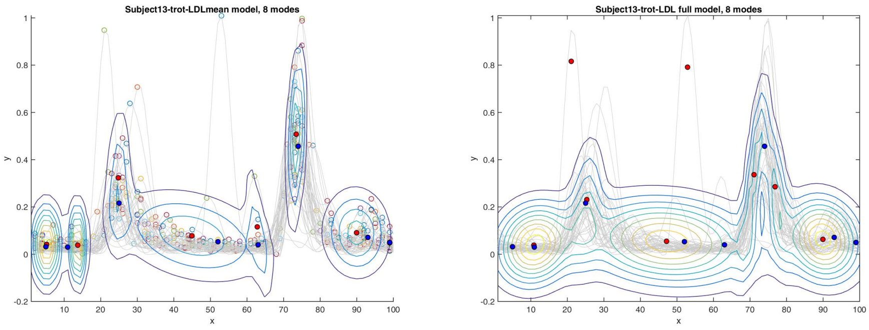
cycle like its duration and range of values. Instead using automatic motion segmentation techniques [ZlTH13, VKK14], we use stance phase information to be consistent with previous work in the biomechanics domain.

Detection of Cycles in Kinematics and EMG Data

The kinematics information on the left fore limb is used for ground contact detection (see Figure 6.1). This detection of ground contacts results yields estimations of stance phases of the associated hoof and thus in the detection of motion cycles. Each cycle of motion includes the sequential ground contacts of all four hooves starting with the contact of the left fore. Note that it is custom to define the notion of footfall patterns consistently starting with the contact of one particular foot (refer to Hill [Hil10] or Robilliard [RPW07] for further readings). Since the two data sets were recorded simultaneously, the segmentation of kinematics data into cycles can be directly transferred to the EMG data. This results in a disjoint separation of the signal M

$$M_f = (s_1, \ldots, s_m) \tag{6.2}$$

where $s_j = (f_{j1}, \ldots, f_{j\lambda}), \lambda \in \mathbb{N}$ for $1 \leq j \leq m$. Each $s_j, 1 < j < m$, represents one complete cycle of the recorded gait. Note that s_1 and s_m may be incomplete cycles due to being at the beginning, respectively at the end. Refer to Figure 6.1 for an example (compare to gait event detection of Olsen et al. [OHAP12]).



(a)

(b)

Figure 6.2: Illustration of GMM model estimated by EM algorithm. Blue markers are peak positions of mean curve. Red markers are optimized means of GMM modes. Gray curves are data. a) peak-based model, b) model based on all data points.

6.2.4 Fitting of Gaussian Mixture Models

Consider a set of cycles $\mathcal{S} = \{s_1, \ldots, s_K\}$ belonging to the same class of trials (trot vs. walk). Let $(p_{j_1}, \ldots, p_{j_l})$ denote the set of all local maximal values in cycle s_j. Let S denote the set of all maxima for all $1 \leq j \leq K$. We hypothesize that the EMG peaks in this set S can be modeled by a Gaussian mixture distribution

$$P(x) = \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i, \Sigma_i), \tag{6.3}$$

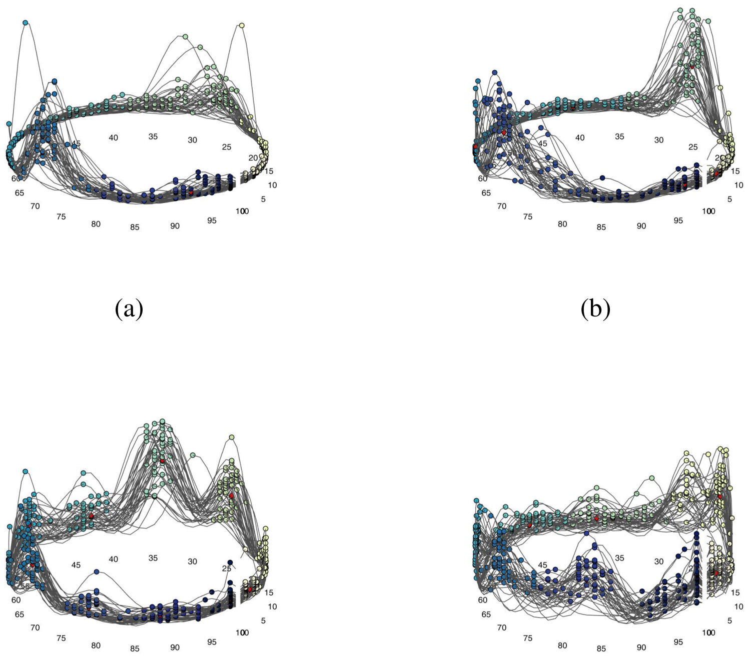
i .e. the peak data are distributed according to a two-dimensional multi-modal density function

$$f(x; \mu_1, \ldots, \mu_n, \Sigma_1, \ldots, \Sigma_n) = \sum_{i=1}^{n} w_i (p(x; \mu_i, \Sigma_i)) \tag{6.4}$$

where $\mu_j = (\mu_{j1}, \mu_{j2})$ is the two-dimensional mean point and Σ_j is the covariance matrix of the jth distribution. In more detail, each of these distributions is of the form

$$p(x, y) = \frac{1}{(2\pi)^k |\Sigma|} e^{-\frac{1}{2}(x-\boldsymbol{\mu})^\top \Sigma^{-1}(y-\boldsymbol{\mu})}. \tag{6.5}$$

In order to fit a suitable GMM model to a given set of peak points S, we first need to determine the number of components needed to represent this model. The other parameters, i. e. model means μ_i, covariances Σ_i and the weights w_i for each



(a)

(b)

(c)

(d)

Figure 6.3: Example of cyclic structure of muscle activation of one individual (subject 13) in trot. a) LDL, b) LDR, c) GML1, d) GMR1. Note the different muscle activation patterns across the two different muscles. The different colors of the data points refer to the different clusters initialized by peaks in the curves describing muscle activation throughout each cycle.

$1 \leq i \leq n$ are then estimated by the expectation maximization (EM) algorithm (cf. Bishop [Bis06]).

One particular difficulty is we do not know how many components can model the data set best. However, this number is crucial for initialization of the GMM. There have been a number of works that discuss initialization of EM algorithms on a more general level. As Blömer and Bujna discuss against the backdrop of theoretical computer science [BB13], the EM algorithm is often sensitive to the choice of the initial parameter vector. The same has also been observed by Melnykov and Melnykov [MM12] especially in settings where the number of mixing components is unknown.

Therefore, it becomes clear that we have to find a suitable initialization for our particular case. Based on this observation and the findings of the more general works [Kwe13], [Wu83], the problem of finding an initialization vector for the EM algorithm is solved by a clustering approach. We exploit the mean sEMG signal per sensor, gait and subject which has found to be a robust estimation of the individual signals per sensor, gait and cycle. Therefore, all peak data of one subject and gait and sensor are assigned to multiple clusters where each individual cluster yields initialization estimations for one mode in the multi-modal Gaussian distribution. The number of clusters is determined by finding the peaks of a mean curve as outlined below.

Clustering Peak Values

The first step is to robustly estimate a parameter $k \in \mathbb{N}$ approximating the number of peaks of a given collection curves representing cycles. Since computation of all maxima per cycle may result in a different number for each given cycle. Since the parameter should be a robustly estimated for complete sets of cycles of one class, a mean curve is computed as a reference and its peak values are computed as a representative of all cycles. Details on how the general algorithm employed to compute the peaks of a discrete function can be found in Mariscotti's works on the subject [Mar67].

The second step is then a temporal pre-clustering of peak locations (position of frame in the time dimension) by the k-means algorithm resulting in a set of k disjoint clusters $\{C_1, \ldots, C_k\}$ for $k \in \mathbb{N}$.

The k-means algorithm (cf Bishop [Bis06]) is a standard method of partitioning points into exactly k clusters by minimization of the total intra-cluster variance

$$\mathcal{J} = \sum_{j=1}^{k} \sum_{i=1}^{m} d(x_i^{(j)}, c_j) \tag{6.6}$$

where c_j denotes the center of cluster C_j and the distance function is the squared distance of two points in the time dimension:

$$d(x, c) = \|x - c\|^2 . \tag{6.7}$$

Finding Modes

Consider the set of all observations $C = \{p_{1_1}, \ldots, p_{k_c}\}$, i. e. the full set of EMG peaks in all given cycles. Each point in C is a point in the space $[0, \ldots, 100] \times [0, 1] \subset \mathbb{Z} \times \mathbb{R}$ and the data set C can be represented by a $N \times 2$ matrix X in which the ith row is given by the ith peak point and where N is the total number of peaks present in the data. A bivariate GMM describing C in the sense of Equation 6.3 can be fitted to the collection of points over all clusters by maximization of the log likelihood function. The log likelihood function is given by

$$\ln p(X|w, \mu, \Sigma) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{K} w_j \mathcal{N}\left(x_i|\mu_i, \Sigma_i\right) \right\}. \tag{6.8}$$

where $p(X|Y)$ denotes the conditional probability of X given Y. It is a well-known fact that there is no closed-form solution for this optimization problem if the number of components K is greated than 1 (cf Bishop [Bis06]). We have to maximize the function $\ln p(X, w, \mu, \Sigma)$ with respect to the mixing proportions w_i such that $\sum_{i=1}^{N} w_i = 1$. This maximization of likelihood problem can be solved by a simple case of an iterative method called expectation maximization algorithm. This special case of the EM algorithm iterates two steps, the *expectation step* and the *maximization step*. In the expectation step, the posterior probabilities of Equation 6.8 are used given the current parameter estimation. In the maximization step, the probabilities resulting from the computation in the expectation step are used to update the estimation of the parameter vectors w, μ and Σ. Technical details on the procedure can be found in [Bis06].

6.2.5 Exploring Data by Hierarchical Clustering

We want to explore if there are subgroups present that can be distinguished by patterns in the muscle activation information. Hierarchical clustering allows for exploring a set of given data D in terms of such subgroups in an unsupervised manner. This is especially convenient when the number of subgroups is unknown.

Agglomerative clustering is achieved by iteratively carrying out three steps (cf Hastie et al. [HTFF05], Olson [Ols95]). The first is step assigns each element of the data set its own cluster, the second step merges the two clusters which have the lowest distance. In a third step, the distances between the newly merged cluster and each remaining cluster is computed. Note that a number of different distance measures can be employed for the computation of cluster proximity in step two. Also, the third step depends on the measure employed to determine this distance, the so-called *linkage*. The three steps are repeated until all clusters have been merged to form one single cluster, that is, after n steps where $|D| = n$. Each iteration yields a partition of the full data set. The partitions of each iteration can be represented by a binary tree (see results in Figures 6.7 and 6.8). In the hierarchical structure of the tree, the different stages of the clustering are reflected as they develop from very fine-grained to coarser partitions (bottom-up).

6.3 Results and Discussion

6.3.1 Results of Cycle Detection

The data set we investigated consisted of several trial runs per gait and subject. There are data of 48 trial runs (subject average: 3.43, mean absolute deviation: 0.55, total maximum 5, total minimum 3) of trot in total and 44 (subject average: 3.14, mean absolute deviation: 0.24, total maximum 4, total minimum 3) trials of walk. The trial runs were segmented by the procedure described in Section 6.2.2 resulting in a total number of 709 motion cycles for trot and 426 cycles for walk. On average, each trial subject performed 51 (mean absolute deviation: 8.39) cycles of trot and 30 (mean absolute deviation: 3.18) cycles of walk. Note that the duration of each trial was 10 seconds, so muscle fatigue is not a contributing factor. It can be observed that there is a smaller number of walk cycles than trot despite the fact that the total trial numbers do not significantly differ. This is explained by the difference in speed between the gaits.

Intra-Subject Variation of sEMG between Different Cycles

The mean curves used to initialize the fitting of GMMs (cf Section 6.2.4) tell us several things. On the one hand, they help determine how many consistently present peaks occur during one average gait cycle. On the other hand, the deviation of all cycles of the signal from this mean curve tell us how reliable this information is. The measure of reliability we chose is the intra-subject variation presented in Tables 6.1 and 6.2.

The intra-subject variation is given per sensor S as the cumulative Euclidean distance between the mean curve of all cycles in the signals to all individual cycles.

$$v_S = \sqrt{\sum_{i=1}^{l_S} |\hat{c} - c_{S_i}|^2} \tag{6.9}$$

where l_S denotes the number of cycles present per sensor. The mean absolute deviation (MAD) is given by

$$MAD_S = \frac{1}{n} \sum_{i=1}^{l_S} |\hat{c} - c_{S_i}|. \tag{6.10}$$

Note that the space in which the deviations are given is the space in which the curves live (remember there was a scaling to $[0, \dots, 100] \times [0, 1] \subset \mathbb{Z} \times \mathbb{R}$).

What can be seen from the numbers together with the visual presentations (Figure 6.5, 6.6) that the activation patterns of individual subjects are consistent in a similar way as the represent subject 13 throughout their trials. That is, in each cycle of one gait the same type of activation pattern is found. Subject 13 showed the highest variation for sensor GML2 and GMR2 in trot. This is consistent with what is seen in Figure 6.5.

Table 6.1: Intra-Subject Deviations Trot

Deviations from cycles to mean curve															
	Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LDL	∅ deviation	0.93	1.04	1.22	1.08	0.94	0.67	1.52	1.49	1.32	1.87	1.0	1.02	0.71	1.04
	MAD	0.26	0.41	0.26	0.25	0.19	0.16	0.24	0.26	0.36	0.29	0.21	0.23	0.29	0.25
GML1	∅ deviation	1.22	1.25	1.22	1.11	0.64	1.33	0.77	1.15	1.21	1.18	1.58	1.22	1.03	1.12
	MAD	0.27	0.24	0.23	0.23	0.18	0.3	0.15	0.26	0.22	0.29	0.26	0.33	0.18	0.24
GML2	∅ deviation	1.18	1.13	0.78	1.21	1.12	0.97	1.29	1.15	1.11	0.99	0.84	1.0	1.4	0.89
	MAD	0.26	0.28	0.29	0.28	0.29	0.28	0.28	0.37	0.34	0.19	0.16	0.29	0.25	0.2
GML3	∅ deviation	1.05	1.15	1.1	1.48	0.17	1.19	1.08	1.09	0.98	1.09	1.18	1.22	1.12	1.06
	MAD	0.21	0.25	0.23	0.2	0.24	0.18	0.2	0.25	0.29	0.23	0.22	0.24	0.19	0.24
LDR	∅ deviation	1.18	0.94	1.03	1.26	0.81	1.0	1.1	0.94	1.24	1.09	1.04	0.71	1.07	0.01
	MAD	0.27	0.2	0.2	0.22	0.28	0.28	0.24	0.19	0.6	0.24	0.22	0.22	0.28	0.25
GMR1	∅ deviation	1.32	1.14	0.98	1.55	0.81	1.09	0.88	0.82	1.33	1.06	1.1	0.11	1.1	1.02
	MAD	0.29	0.28	0.3	0.3	0.16	0.2	0.25	0.2	0.25	0.21	0.26	0.16	0.18	0.26
GMR2	∅ deviation	1.31	1.14	0.97	1.12	1.06	1.05	1.05	1.3	1.32	0.97	1.11	1.12	1.54	0.91
	MAD	0.23	0.26	0.27	0.26	0.21	0.2	0.21	0.51	0.34	0.17	0.19	0.33	0.25	0.22
GMR3	∅ deviation	1.40	1.38	1.27	0.83	1.28	1.35	1.34	1.28	1.23	1.13	1.11	1.17	0.86	1.42
	MAD	0.21	0.32	0.36	0.3	0.22	0.24	0.24	0.3	0.23	0.18	0.18	0.39	0.2	0.23

Inter-Subject Variation of sEMG between Different Individuals

In addition to the consistency discussed in the Section above, we were interested to find out if there is a number of peaks which characterizes the signal recorded at each sensor location. By evaluation of the images comparing the individual mean curves of all subjects per sensor and by computing the global mean curve out of these. The inter-subject variation is given analogously to Formulas 6.9 and 6.10 for the mean curves of all individual per sensor. We found that there is roughly a two-peak structure visible in the data.

6.3.2 Results of Mode Estimation

Biomechanics experts are interested in the peaks of EMG signals because they were assumed to be characteristic landmarks for the investigation of muscle activity. We wanted to show that the activation pattern of a muscle can indeed be characterized by the location and height of peaks in sEMG cycles. The models we have created are based on this information. However, they could as well have been based on the full set of discrete samples of the data curve given in our set of raw data.

Table 6.2: Intra-Subject Deviations Walk

Deviations from cycles to mean curve															
	Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LDL	∅ deviation	1.01	1.21	1.22	1.0	0.98	1.21	1.15	1.06	1.47	1.25	1.05	1.33	0.99	1.41
	MAD	0.24	0.19	0.25	0.19	0.22	0.19	0.24	0.25	0.41	0.22	0.27	0.19	0.34	0.28
GML1	∅ deviation	1.31	1.15	0.76	0.96	0.91	1.11	1.14	1.38	0.93	1.47	1.43	0.61	1.64	0.95
	MAD	0.29	0.2	0.22	0.14	0.18	0.28	0.24	0.21	0.37	0.25	0.26	0.36	0.24	0.2
GML2	∅ deviation	0.83	0.98	0.83	1.59	0.93	0.88	0.78	0.87	1.13	1.05	1.04	0.47	1.16	0.85
	MAD	0.25	0.19	0.25	0.3	0.19	0.22	0.16	0.28	0.24	0.22	0.2	0.35	0.16	0.34
GML3	∅ deviation	1.1	0.7	1.02	0.7	1.28	1.04	0.63	1.06	1.1	1.3	0.8	0.98	0.8	0.93
	MAD	0.25	0.14	0.19	0.18	0.17	0.2	0.17	0.18	0.22	0.25	0.22	0.37	0.22	0.39
LDR	∅ deviation	1.21	1.23	1.05	1.6	1.13	1.05	1.14	0.93	1.46	1.1	1.3	0.67	1.27	1.27
	MAD	0.34	0.36	0.25	0.23	0.21	0.23	0.25	0.19	0.21	0.2	0.27	0.23	0.23	0.22
GMR1	∅ deviation	1.05	1.27	1.09	1.14	1.14	1.41	1.23	0.94	1.32	0.97	1.07	0.1	1.24	1.07
	MAD	0.3	0.19	0.13	0.17	0.2	0.55	0.29	0.21	0.28	0.18	0.16	0.01	0.2	0.19
GMR2	∅ deviation	1.13	0.99	1.19	1.18	0.92	1.1	1.28	1.22	1.17	1.0	1.0	0.83	1.02	0.89
	MAD	0.33	0.2	0.26	0.26	0.19	0.17	0.23	0.26	0.21	0.18	0.18	0.38	0.28	0.22
GMR3	∅ deviation	1.17	1.17	0.77	1.25	1.14	1.29	1.43	1.37	1.09	1.26	1.04	1.01	0.96	1.19
	MAD	0.17	0.21	0.3	0.22	0.19	0.14	0.28	0.16	0.28	0.19	0.16	0.2	0.17	0.26

One hypothesis based on the theory of GMM models indicates why it makes sense to rather model the peaks than the full set of points. The full point set is much more likely to produce overlapping components respectively modes with shared covariance. This will result in a mix of Gaussian modes in space and is also known to slow down the speed of convergence (cf Naim and Gildea [NG12]). In order to separate the components better, we can benefit from modeling by peaks instead of modeling the full data set.

In practice, we can evaluate the difference between fitting the multi-modal model to the peaks of cycles as opposed to all available data. The different outcomes in both cases will be discussed in the following.

A multi-modal bivariate Gaussian distribution was fitted to the full data set in the same way and with the same initialization as for the peak-based model. Since this produces the same number of Gaussian modes as for the peak-based model, this allows a straightforward comparison between the GMMs. As can be seen in the example image showing both scenarios (Fig 6.2 shows an example for the left-side longissimus of subject 13), the peak-based models appear similar in quality to the models based on the complete signal. More precisely, the models based on the full data set overlap the models fitted to the peaks. In order to give a quantitative notion of this perceived similarity and in order to show that the peak-based models also suffice to characterize the signal, we need to compute the similarity of the associated GMMs.

The similarity of probability distributions can generally be assessed by a distance measure between the respective distribution functions. In the classical case, i. e. for uni-modal probability models, this distance measure is given by the

Table 6.3: Inter-Subject Deviations Trot

Deviations from cycles to global mean curve per sensor															
	Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LDL	∅ deviation	1.46	1.44	1.4	1.31	1.18	1.23	2.9	2.43	1.74	1.89	1.23	1.34	1.41	1.38
	MAD	0.16	0.23	0.32	0.22	0.18	0.12	0.37	0.29	0.41	0.25	0.23	0.24	0.14	0.21
GML1	∅ deviation	1.44	1.63	1.53	1.43	1.57	1.93	1.36	1.54	1.43	2.65	1.79	1.57	1.58	1.55
	MAD	0.26	0.3	0.3	0.2	0.11	0.25	0.12	0.27	0.22	0.21	0.33	0.27	0.17	0.24
GML2	∅ deviation	1.44	1.65	1.32	1.61	1.38	1.92	1.59	1.61	1.39	1.4	1.2	1.23	1.82	1.34
	MAD	0.23	0.39	0.16	0.22	0.29	0.25	0.35	0.29	0.35	0.22	0.18	0.23	0.47	0.23
GML3	∅ deviation	1.35	1.46	1.35	1.82	1.69	1.43	1.29	1.52	1.42	2.44	1.54	1.43	1.46	1.49
	MAD	0.18	0.22	0.19	0.24	0.18	0.12	0.19	0.19	0.24	0.26	0.18	0.27	0.17	0.14
LDR	∅ deviation	1.47	1.15	1.25	1.86	1.31	1.37	2.11	1.45	1.72	1.49	1.83	1.02	1.45	1.25
	MAD	0.17	0.16	0.2	0.37	0.2	0.36	0.29	0.16	0.35	0.23	0.24	0.18	0.29	0.23
GMR1	∅ deviation	1.56	1.39	1.4	2.05	1.19	1.43	1.24	1.58	1.51	1.41	1.33	1.92	1.38	1.21
	MAD	0.37	0.35	0.2	0.72	0.13	0.15	0.27	0.13	0.27	0.22	0.33	0.02	0.25	0.22
GMR2	∅ deviation	1.45	1.38	1.27	1.32	1.36	1.39	1.29	1.61	1.69	1.25	1.41	1.26	1.63	1.1
	MAD	0.26	0.33	0.2	0.33	0.29	0.25	0.18	0.38	0.37	0.22	0.22	0.31	0.27	0.2
GMR3	∅ deviation	1.77	1.55	1.68	1.45	1.81	1.56	1.72	1.67	1.71	1.65	1.7	1.45	1.46	1.58
	MAD	0.26	0.31	0.37	0.25	0.2	0.35	0.4	0.37	0.25	0.24	0.24	0.23	0.2	0.3

Kullback-Leibler distance. Unfortunately, the definition of this distance measure does not translate to multi-modal distributions which are of the form described by Equation 6.4. However, there is a similar distance measure, the *Cauchy-Schwarz distance* which is based on the following idea (cf Kampa and Principe [KHP11]).

Given two multi-modal Gaussian distributions P and Q, the Cauchy-Schwarz distance between them is defined by

$$\mathrm{dist}_{CS}(P,Q) = -\log\left(\frac{\int Q(x)P(x)dx}{\sqrt{\int Q(x)^2dx \int P(x)^2dx}}\right). \tag{6.11}$$

This is a symmetric measure for arbitrary probability distribution functions such that $0 \leq \mathrm{dist}_{CS} < \infty$. In particular, this measure translates to the special case of multi-modal Gaussian distributions when the distributions accommodate the same number of modes.

The two sub-tables in Table 6.5 show the results of Cauchy-Schwarz distance computation per gait, sensor and individual of the group.

In sum, for all walk trials, the distances show that the two models are quite closely aligned (median: 0.09). This shows that the models based on the maximally available number of peaks represent the signal sufficiently well. For the trot data, the same holds true even though the distances are generally higher than in walk (median: 0.22). As can be seen in the table, the Cauchy-Schwarz distances also vary quite a lot between subjects as well as between different muscles. This is one observation that relates with our findings of intra-subject and inter-subject deviations discussed in Sections 6.3.1, 6.3.1.

Table 6.4: Inter-Subject Deviations Walk

Deviations from cycles to global mean curve per sensor															
	Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LDL	∅ deviation	1.14	1.31	1.46	1.25	1.14	1.37	1.67	1.46	1.62	1.37	1.34	1.4	1.36	1.53
	MAD	0.24	0.44	0.27	0.25	0.22	0.27	0.21	0.26	0.27	0.3	0.3	0.19	0.19	0.34
GML1	∅ deviation	1.61	1.48	1.18	1.46	1.32	1.29	1.45	1.5	1.28	1.76	1.71	1.5	2.05	1.21
	MAD	0.36	0.44	0.17	0.21	0.16	0.22	0.17	0.21	0.16	0.43	0.31	0.18	0.3	0.13
GML2	∅ deviation	0.97	1.48	0.96	3.71	1.07	1.23	1.04	1.18	1.61	1.21	1.34	1.36	1.39	1.23
	MAD	0.23	0.35	0.17	0.32	0.18	0.26	0.09	0.29	0.22	0.29	0.28	0.11	0.26	0.16
GML3	∅ deviation	1.24	0.9	1.12	1.18	1.58	1.25	0.95	1.4	1.34	1.51	1.33	1.23	1.05	1.26
	MAD	0.3	0.21	0.29	0.23	0.18	0.18	0.13	0.25	0.28	0.3	0.15	0.26	0.16	0.25
LDR	∅ deviation	1.28	1.44	1.26	2.91	1.29	1.65	1.46	1.31	1.81	1.27	1.87	1.35	1.4	1.41
	MAD	0.39	0.34	0.21	0.49	0.28	0.21	0.2	0.2	0.48	0.28	0.17	0.16	0.38	0.31
GMR1	∅ deviation	1.3	1.84	1.34	1.93	1.37	1.64	1.66	1.25	1.47	1.4	1.36	1.72	1.45	1.28
	MAD	0.21	0.33	0.23	0.19	0.35	0.35	0.37	0.17	0.28	0.19	0.25	0.01	0.28	0.2
GMR2	∅ deviation	1.4	1.54	1.28	1.82	1.13	1.35	1.45	1.61	1.28	1.14	1.25	1.44	1.23	1.17
	MAD	0.16	0.32	0.25	0.33	0.24	0.25	0.19	0.38	0.25	0.16	0.23	0.17	0.18	0.15
GMR3	∅ deviation	1.32	1.82	1.31	1.51	1.48	1.53	1.78	1.84	1.29	1.46	1.34	1.15	1.33	1.38
	MAD	0.19	0.35	0.26	0.37	0.18	0.2	0.21	0.28	0.33	0.23	0.23	0.22	0.11	0.2

In order to rate the distance values presented in Table 6.5 mean, comparative distance values are helpful. Therefore, a multi-modal model based on the same initialization was fitted to a stream of normally distributed noise. The Cauchy-Schwarz distances between this random multi-modal GMM and the peak-based GMM were computed. For trot, the distance values to the peak-based model were, on average, more than 4 times higher than the ones from the complete set of original data (displayed in Table 6.5) with a maximum factor of 9 times and a minimum factor of 2 times. For walk, The distance values to the peak-based model were, on average, more than 9 times higher than the ones from the complete set of original data (displayed in Table 6.5) with a maximum factor of 33 times and a minimum factor of 3 times.

Based on the discussion above, one important finding is that muscle activation signals of walk and trot can represented by GMMs of the collective peak patterns per cycle (one GMM accommodating all cycles of a set of trials capturing the same gait).

The specific results we got when doing so are discussed in the following. GMMs were fitted by the procedure described in Section 6.2.4. A most important observation is that the number of modes found by this method was always greater than one (minimum for trot: 5, minimum for walk: 6). This is especially surprising since, to date, we expect one significant peak in sEMG data per cycle [ZVK+15]. This number also varied across different subjects and sensor positions (the ranges of this number of components for each subject is listed in Table 6.6.

In particular it is clear that characterizing muscle activation by investigation of just one highest peak per cycle is not sufficient for either walk and trot and also

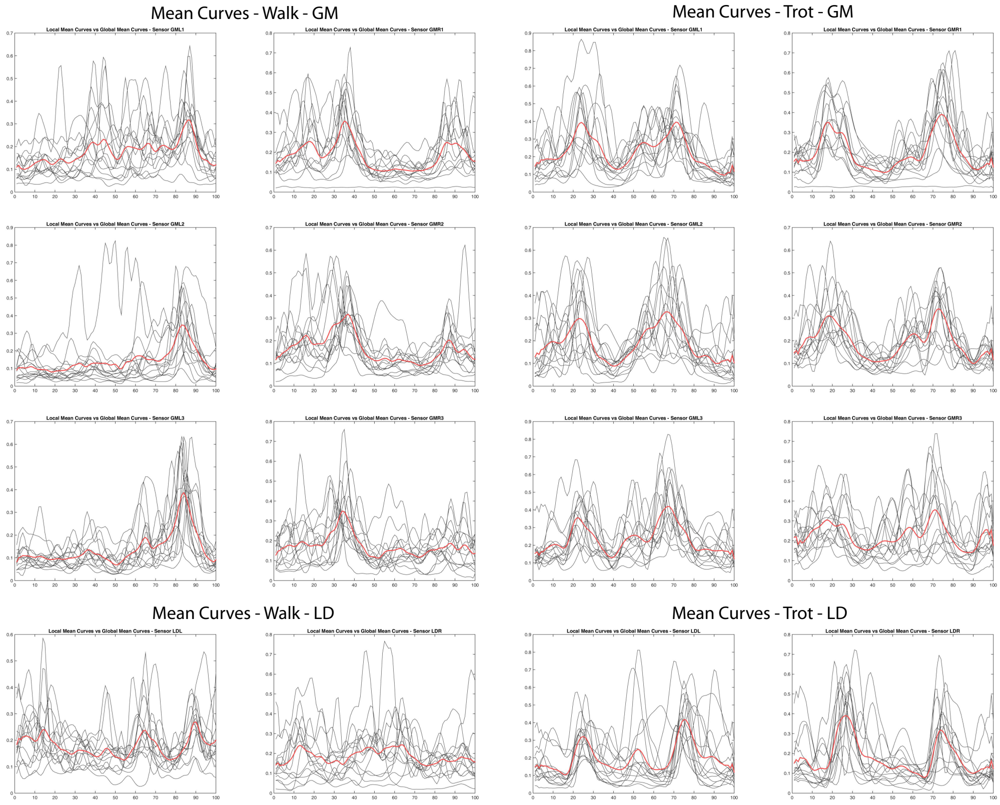


Figure 6.4: Mean curves of all subjects per sensor (gray), global mean curve per sensor (red). It can be seen that there is a general two-peak activation pattern. Though this is observed on average data, individual cases differ vastly.

for neither of the two different groups of muscles (longissimus dorsi and gluteus medius) investigated in our study. (cf Zsoldos et al. [ZKV$^+$15, ZVK$^+$15]).

In order to find out what number of modes to focus on instead, we individually look at the two types of muscles in more detail.

Peak Patterns of the M. Longissimus Dorsi

For the m. longissimus dorsi there is a strong indication of a two-peak pattern in trot. This can be seen especially well on average, i. e. by considering the mean curves per sensor over all individual trials and cycles (Figure 6.4). However, this two-peak pattern is not as apparent for some individuals. This is why it makes sense to distinguish groups of individuals for whom this is more apparent then for others (refer to Section 6.3.3).

Based on the assumption of a two-peak activation pattern in the sEMG signals

Table 6.5: Cauchy-Schwarz Distances of Multi-Modal Models Full vs. Peak-Based

Cauchy-Schwarz Distances Trot

Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Median	MAD
LDL	0.27	0.15	0.16	0.13	0.18	0.13	0.32	0.23	0.53	0.28	0.2	0.18	0.13	0.12	0.18	0.05
GML1	0.31	0.27	0.22	0.2	0.71	0.33	0.21	0.16	0.71	0.33	0.54	0.67	0.26	0.73	0.32	0.11
GML2	0.24	0.26	0.2	0.19	0.55	0.13	0.25	0.17	0.1	0.22	0.18	0.09	0.21	0.1	0.19	0.05
GML3	0.18	0.23	0.13	0.23	0.27	0.15	0.17	0.25	0.3	0.4	0.63	0.47	0.3	0.21	0.24	0.06
LDR	0.25	0.2	0.15	0.7	0.14	0.16	0.32	0.12	0.11	0.71	0.18	0.21	0.13	0.13	0.17	0.04
GMR1	0.2	0.3	0.2	0.24	0.2	0.24	0.29	0.21	0.13	0.26	0.21	0.41	0.29	0.16	0.23	0.03
GMR2	0.09	0.15	0.17	0.2	0.34	0.19	0.22	0.16	0.17	0.21	0.22	0.24	0.18	0.1	0.18	0.03
GMR3	0.18	0.12	0.13	0.24	0.29	0.21	0.23	0.16	0.22	0.27	0.24	0.38	0.19	0.24	0.23	0.05
Median	0.22	0.22	0.17	0.21	0.28	0.18	0.24	0.17	0.2	0.28	0.22	0.31	0.19	0.14		
MAD	0.04	0.06	0.03	0.03	0.09	0.04	0.04	0.03	0.09	0.05	0.03	0.17	0.06	0.04		

Cauchy-Schwarz Distances Walk

Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Median	MAD
LDL	0.08	0.09	0.1	0.12	0.07	0.11	0.31	0.07	0.04	0.21	0.07	0.07	0.06	0.08	0.08	0.02
GML1	0.07	0.18	0.04	0.14	0.15	0.09	0.09	0.04	0.05	0.10	0.1	0.02	0.16	0.09	0.09	0.05
GML2	0.05	0.1	0.08	0.25	0.1	0.1	0.07	0.04	0.05	0.1	0.09	0.03	0.1	0.03	0.09	0.02
GML3	0.13	0.19	0.07	0.15	0.18	0.13	0.14	0.15	0.10	0.12	0.10	0.03	0.11	0.06	0.12	0.04
LDR	0.13	0.09	0.07	0.22	0.11	0.04	0.22	0.05	0.12	0.08	0.12	0.08	0.05	0.06	0.10	0.03
GMR1	0.06	0.17	0.1	0.2	0.1	0.05	0.14	0.07	0.12	0.08	0.03	0.05	0.16	0.04	0.09	0.03
GMR2	0.05	0.16	0.1	0.14	0.1	0.14	0.1	0.18	0.06	0.08	0.1	0.03	0.12	0.07	0.1	0.03
GMR3	0.07	0.28	0.05	0.11	0.15	0.11	0.03	0.12	0.11	0.08	0.07	0.09	0.09	0.06	0.09	0.02
Median	0.07	0.16	0.08	0.14	0.1	0.1	0.12	0.07	0.05	0.08	0.08	0.04	0.1	0.06		
MAD	0.01	0.04	0.02	0.02	0.02	0.01	0.04	0.03	0.01	0.009	0.01	0.01	0.01	0.02		

of m. longissimus dorsi, we computed the average distances between the two highest peaks in the signals (separating the sensors bilaterally). Table 6.7 lists these average distances found between the two highest peaks for all subjects. The distance values fall into categories according to quite consistently across the entire group of subjects.

Note that the two-peak assumption cannot be made for walk in the same way as for trot. However, the distances between the two highest peaks in the signals of the m. longissimus in walk are also quite consistent. All in all, this is a very interesting finding because this shows a significant difference between the activation patterns found in sEMG recordings of walk vs. trot.

Peak Patterns of the M. Gluteus Medius

Longissimus and gluteus have different tasks and belong to different parts of the equine body. Therefore, it is not surprising that the activation patterns of the m. gluteus medius in individuals are entirely different compared to the ones of the longissimus dorsi. In fact, for individuals (cf Figure 6.5) the pattern can seem even chaotic.

Table 6.6: Maximum and Minimum Number of Modes Per Subject S

Max. and Min. in Trot														
Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
max # comps	10	9	9	9	10	9	11	9	10	11	9	11	10	8
min # comps	6	6	7	7	6	7	6	6	7	8	6	7	7	5
Max. and Min. in Walk														
Subject number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
max # comps	11	12	10	13	10	11	12	11	11	10	10	11	12	11
min # comps	8	9	7	8	7	6	8	6	8	7	7	7	8	7

Even more surprising is that for the m. gluteus medius there is also an indication of a two-peak pattern in trot. Again, this is a phenomenon which is observed well on the mean data, i. e. the mean curves per sensor over all individual trials and cycles (Figure 6.4), whereas for individuals the two-peak pattern is not as apparent. The classification of groups of individuals with different types of activation patterns is relevant here also (Section 6.3.3).

On average, this suggests that there is a significant peak pattern in sEMG data of the m. gluteus medius which can be characterized by less than 5 peaks for different parts of this muscle.

Anyway, it can be seen that different parts of the m. gluteus respond differently to stimuli in terms of muscle activation. This is especially interesting in light of the results of Bruce [BT85] and Licka et al. [LPF04, LFP09].

In the paper of Zsoldos et al. [ZVK$^+$15], the authors investigated just a single highest peak per cycle of signals acquired from the activation of different parts of the m. gluteus medius. This would be the equivalent of a single mode Gaussian model. However, as can be observed in Figure 6.5, the location of the highest peak per sensor can be scattered across the complete interval $[1, \ldots, 100]$. This makes describing EMG signals by uni-modal Gaussians unreliable. Using the mean curve as initialization is a step towards more robust models and taking into account more than one peak allows for a more detailed analysis.

In summary, the two-peak structure may serve as a pointer towards a hypothesis about the relevant number of peaks to look out for in EMG signals of both types of muscle. However, we can expect differences in the signal structure between muscles, between individuals and also between different gaits. Using multi-modal GMMs is a more precise way of modeling muscle activation patterns of cycles in sEMG data.

Table 6.7: Distances between the two highest peaks found in the sEMG signal per cycle. Note that the distance is measured on a cycle, i. e. we always "‘take the shortest route"’ even across the first respectively last frame (the distance between the first and last frame will be 1 instead of 99).

	WALK	WALK	WALK	WALK		TROT	TROT	TROT	TROT
	10-20	20-30	30-40	40-50		10-20	20-30	30-40	40-50
GML1	1	0	5	8		1	1	5	7
GML2	2	3	2	7		3	4	1	6
GML3	1	3	7	3		2	2	2	8
GMR1	0	2	5	7		1	2	3	8
GMR2	2	2	4	6		1	5	3	5
GMR3	1	3	2	8		0	3	3	8
LDL	1	2	6	5		1	4	4	5
LDR	1	0	6	7		2	2	2	2

6.3.3 Results of Hierarchical Clustering

When we see highly heterogenous data, it makes sense to explore those by hierarchical clustering in the way introduced in Section 6.2.5. In our particular case, we were surprised by the variability of data in every direction. Employing the clustering step is an important step toward restructuring the data set. In this way, we will be able to compare the results of individual clusters with those of the complete group in order to identify candidates for the typical or average specimen and also to detect outliers in the groups. Hence, from this approach, new possibilities arise for analysis and exploration.

Physiologically, the m. longissimus dorsi and m gluteus medius serve for different tasks. So it makes sense to separate the two for a clustering.

Clustering of M. Longissimus Dorsi

Of the m. longissimus dorsi we used all information of corresponding sensors (’LDL’ and ’LDR’). That is, for each of the two sensors

- The locations of the 5 highest peaks.
- The values of these peaks.
- The total number of modes for this sensor.

This sums up to a feature vector $f_{LD_i} \in \mathbb{R}^{11}$ per sensor ($1 \leq i \leq 2$). Note that using 5 highest peaks per cycle is consistent with the upper limit for a number of peaks. This is due to fact that this was a minimum number of components for one of the subjects when fitting GMMs (Section 6.3.2, Table 6.6, subject 14).

The features were used to compute a correlation matrix and from that, a dissimilarity matrix showing the pairwise Euclidean distances between any pair of feature values. According to these distances, agglomerative hierarchical clustering by using single linkage. We used single link clustering with Euclidean distance measure. Single link means, in step three (see Section 6.2.5) when the distance of a new compound cluster to all other clusters is computed, the distance from the merged cluster to another cluster is equal to the shortest distance from any of its members to the outside cluster.

Clustering of M. Gluteus Medius

Of the m. gluteus medius we used all information of corresponding sensors ('GML1-GML3' and 'GMR1-GMR3'). That is, for each of the 6 sensors

- The locations of the 5 highest peaks.
- The values of these peaks.
- The total number of modes for this sensor.

This sums up to a feature vector $f_{GM_j} \in \mathbb{R}^{54}$ per sensor ($1 \leq j \leq 6$). From there, the procedure was as described above.

The resulting clusterings for both groups of muscles are represented by cluster trees (Figure 6.8 and 6.7). Note that the images only show the cluster hierarchies from the level at which there were 4 different clusters to the top. Since agglomerative clustering is a bottom-up approach, this leaves out the first stages at which each individual was in its own cluster.

The results of the clustering differ vastly between m. longissimus dorsi and m. gluteus medius. This is coherent with the fact that the two groups at hand serve different tasks in the locomotion of the equine body.

6.4 Conclusions

We introduced new modeling techniques to analysis of quadrupedal sEMG processing. Combining GMMs and hierarchical clustering techniques with a state of the art pre-processing pipeline, we detected patterns in sEMG data and made a step towards their analysis. Since there is a yet lack of standardization of animal sEMG signals, this work tested a variety of methods toward the identification of coherent patterns in such data sets. With the help of these tools, experts in the field will be able to acquire new insight in the structure of sEMG readings on different levels.

On a more global level, i.e. as a finding on the complete group of 14 equine subjects, a multi-peak pattern was identified which represents the general activation pattern per sensor location.

All in all, the data set used in the current study proved to be more variable than expected, even though the group of horses was homogenous (all were mares, all were from the same breed, all were close in age). Despite this, the muscle activation patterns showed marked individuality.

This poses an interesting question for further research. Namely, whether it is more important to analyze individuality in heterogenous data of otherwise consistent groups of subjects or is it more desirable to focus on analyzing the common features within such groups. While this question has not been answered in the field of animal electromyography, questions have risen about how heterogenous muscle activation estimates can be and how to deal with this (cf. Staudenmann et al. [SSvD13]). Both objectives, investigation of individuality and analysis of general features, could be of equal importance. In fact, a hierarchical exploration of data brings interesting opportunities to pursue both. We have outlined a way of doing this using hierarchical clustering.

We think, the study will contribute to progress in the field of animal muscle mechanics by introducing new techniques to explore sEMG signals of quadrupeds in terms of individual and common aspects. It is especially important to further standardize this type of research, thus closing the gap between animal and human electromyography.

With the help of the methods presented in this paper the differences in muscle activation between different gaits were found on both general and individual level. Such differences as might be expected also occurred between the two different groups of muscles. Especially in walk, the differences between m.longissimus dorsi and m. gluteus medius were clearly visible. It could be interesting to investigate the cause of this divergence further. Moreover, it is important to find out in what way the peaks observed at different locations of the same muscle will differ. The application of three different sensors to the m. gluteus medius muscles of both sides has shown that, in each individual, there are strong indications of a large difference, while on group average level, the difference was less apparent but still present.

6.4.1 Outlook

Employing statistical methods for modeling of sEMG signals has shown that these signals are sufficiently represented by a maximally available number of peaks. It would be interesting to see if creating more compact GMMs is possible. We propose merging adjacent modes of one GMM based on their Kullback-Leibler distance and or the divergence in direction of principal axes. The Kullback-Leibler divergence could well be computed between uni.modal Gaussians. This could yield a more accurate classification of individuals by allowing a controlled overlap of peaks.

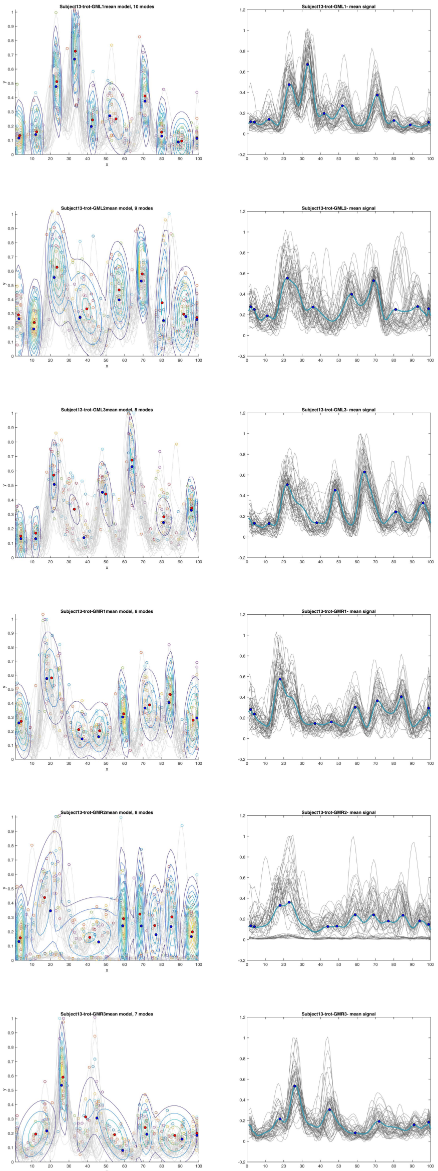
Since the number of subject was too small to allow for a larger number of clusters, we would ideally create larger groups of subjects. This could also involve capturing a larger data pool allowing for comparison of sEMG patterns between species.

In our GMM fitting approach, estimation of modes was initialized by the maximally available number of peaks of a mean curve per sensor and was optimized based on that number. Doing this more automatically by optimizing for different error functions could help. E. g. the works of Melnykov and Melnykov [MM12] propose a strategy initializing mean vectors by choosing points with higher concentrations of neighbors and using a truncated normal distribution for the preliminary estimation.

Furthermore, subsequent work investigating muscle activation patterns during other gaits and movements could show how our results transfer to other scenarios.

From segmenting the data into cycles of gaits the question arises whether splitting sequences at this point causes a problem with redundant modes. Though locating ground contacts of a specific hoof has been a simple task, this does not answer the question if this may have split data in a way that one group of peaks will be separated such that some of its members are in the beginning of the cycles and some will be located at the end. In the worst case, this would lead to one additional mode (at the end) where there is actually just one (in the beginning). Splitting cycles at a different point would only shift the same problem to a different stage of the gait. Creating an overlap of semi-cycles at the beginning and at the end of each cycle could help alleviate this. A *panoramic* view of frame interval $[f_{-50}, \ldots, f_{150}]$ of each cycle could make it easier to exclude errors by inept segmentation.

TROT – GM - SUBJECT 13



Subject 13 GML1	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.60	0.90	0.10	0.30	0.40
WhMC%	55	91	3	23	33
WhValue	0.15	0.13	0.10	0.10	0.10
HhMC%	33	23	71	55	43
HhRelValue	0.25	0.09	0.13	0.51	0.73

Subject 13 GML2	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.44	0.33	0.89	0.11	0.56
WhMC%	40	24	92	3	57
WhValue	0.16	0.13	0.12	0.12	0.11
HhMC%	24	70	57	80	40
HhRelValue	0.33	0.63	0.30	0.29	0.47

Subject 13 GML3	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.88	0.50	0.63	0.13	1.00
WhMC%	81	33	50	4	96
WhValue	0.17	0.15	0.13	0.13	0.13
HhMC%	64	22	50	96	33
HhRelValue	0.28	0.33	0.44	0.15	0.34

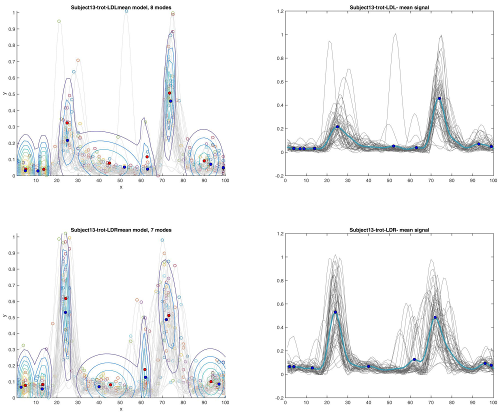
Subject 13 GMR1	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.25	1.00	0.13	0.75	0.50
WhMC%	21	97	4	73	47
WhValue	0.16	0.16	0.14	0.13	0.12
HhMC%	21	85	73	59	97
HhRelValue	0.58	0.28	0.27	0.39	0.20

Subject 13 GMR2	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.38	0.25	1.00	0.88	0.13
WhMC%	41	17	96	85	4
WhValue	0.30	0.16	0.15	0.10	0.09
HhMC%	17	68	85	59	76
HhRelValue	0.16	0.44	0.20	0.30	0.16

Subject 13 GMR3	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.86	0.57	0.14	0.43	0.29
WhMC%	87	55	12	39	27
WhValue	0.24	0.21	0.21	0.09	0.09
HhMC%	27	39	71	99	12
HhRelValue	0.19	0.19	0.19	0.31	0.59

Figure 6.5: Trot of subject 13, m. gluteus medius. Leftmost column: GMMs per sensor. Middle column: curves (gray) of subject 13 per sensor location and mean curve (blue). Tables are based on the ith $(1 \leq i \leq 5)$ highest peaks. WhRelValue: relative location (wrt total number of modes in GMM) of mode with ith highest contribution, *WhMC%*: frame number of mode with ith highest contribution within GMM (relative location wrt cycle length), *WhValue*: contribution value at WhMC%, *HhMC%*: frame number of ith highest peak within motion cycle (relative location wrt cycle length), *WhValue*: peak height at HhMC%.

TROT – LD - SUBJECT 13



Subject 13 LDR	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.50	0.88	0.13	0.75	0.25
WhMC%	45	90	5	73	14
WhValue	0.34	0.17	0.13	0.10	0.07
HhMC%	73	25	63	90	45
HhRelValue	0.08	0.09	0.04	0.51	0.04

Subject 13 LDR	1st peak	2nd peak	3rd peak	4th peak	5th peak
WhRelValue	0.57	1.00	0.86	0.14	0.43
WhMC%	45	93	73	5	24
WhValue	0.28	0.22	0.13	0.12	0.10
HhMC%	24	73	62	93	13
HhRelValue	0.08	0.10	0.51	0.08	0.62

Figure 6.6: Trot of subject 13, m. longissimus dorsi. Leftmost column: GMMs per sensor. Middle column: curves (gray) of subject 13 per sensor location and mean curve (blue). Tables are based on the ith $(1 \leq i \leq 5)$ highest peaks. WhRelValue: relative location (wrt total number of modes in GMM) of mode with ith highest contribution, *WhMC%*: frame number of mode with ith highest contribution within GMM (relative location wrt cycle length), *WhValue*: contribution value at WhMC%, *HhMC%*: frame number of ith highest peak within motion cycle (relative location wrt cycle length), *WhValue*: peak height at HhMC%.

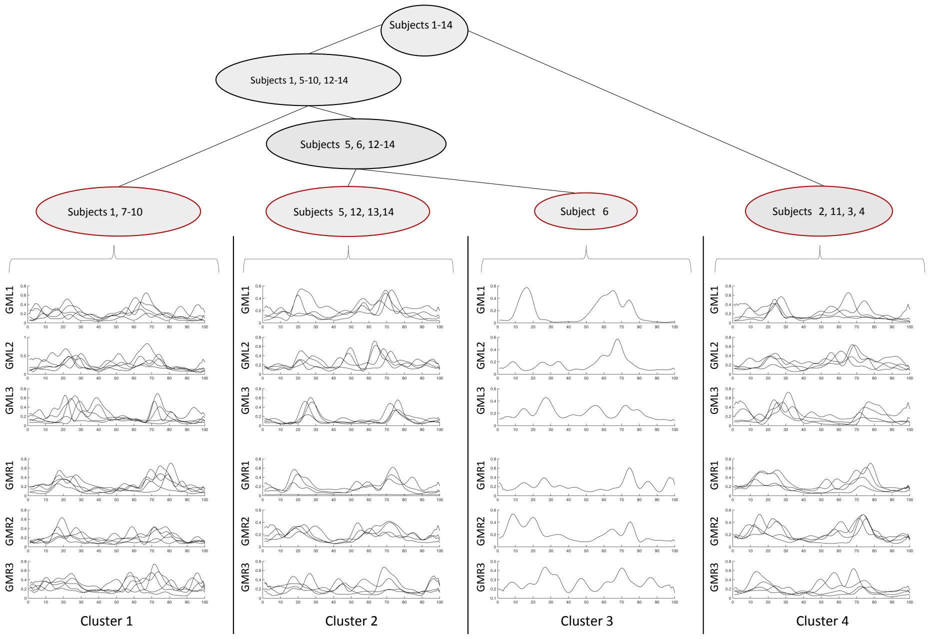


Figure 6.7: Cluster tree visualization of hierarchical clustering for the m. gluteus medius (three sensors per side) in trot. Note that only the upper part of the tree is shown, namely at a level where 4 clusters are created (red highlighting).

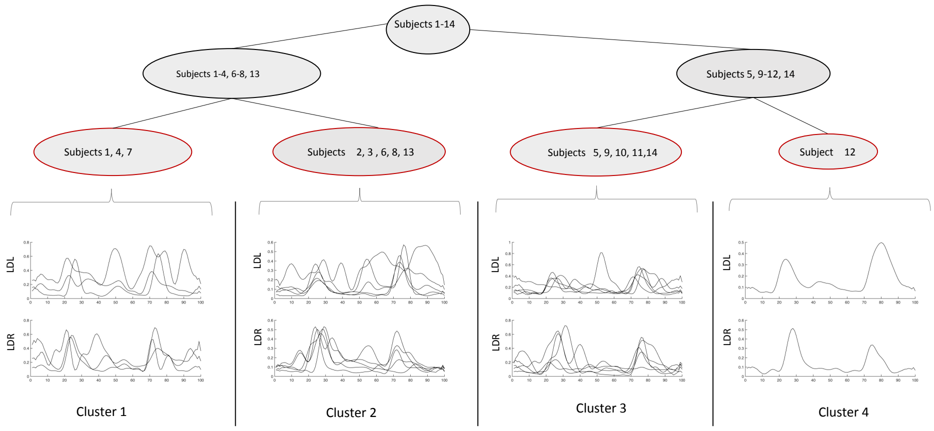


Figure 6.8: Cluster tree visualization of hierarchical clustering for the m. longissimus dorsi (one sensor per side) in trot. Note that only the upper part of the tree is shown, namely at a level where 4 clusters are created (red highlighting).

CHAPTER 7

Estimation of Soft Biometrics by Classifying One Inertial Sensor

Science is the systematic
classification of experience.

George Henry Lewes

This chapter will give a brief overview on solving a classification of human soft biometrics. Details can be found in Riaz et al. [RVKW15].

Sparse representations of human motions have been around for a while. Classification by these is possible (e. g. point light walker [Joh73, Tro02]) This has been done a few times before. As opposed to action recognition where information about activities is found ([SLC04, MHK06, OGB14, BBBG14, JDLP11, LN07])), e.g. estimation of poses and skeletons from video and motion capture data

Information on kinematics properties of the acting subjects may also be retrieved by classification of motion representations. This can be done using motion capture data from passive or active devices as well as contact forces measurements (Venture et al. [VAN08], Kirk et al. [KOF05]).

As the market for wearable devices is rapidly growing (Liew et al. [LWS$^+$15], Son et al. [SLQ$^+$14], Tao et al. [TLZF12]), numerous methods to analyze the data sets captured by these are emerging. It has been shown that reconstruction of motion from very sparse sets of reference sensors is possible (Tautges et al. [TZK$^+$11], Riaz et al. [RTKW15]) even though the quality of such measurements is much less accurate than optical motion capture (Le Masurier et al. [LMTL03], Foster et al. [FLFM$^+$05]). In particular, data collected using triaxial accelerometers turn out to be suitable for such classification tasks.

In our paper (Riaz et al. [RVKW15]), we proved that estimation of soft biometrics possible based on data recorded by a single wearable device. The biometrics classified in our experiments are gender, age and height as examples of inertial properties of moving subjects. The classification method processes recordings of

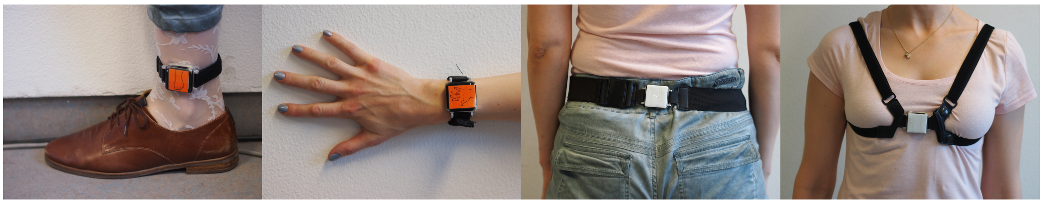


Figure 7.1: Placement of four APDM Opal IMUs on different body parts. The sensors were placed on four different locations: left ankle, right wrist, lower back, and chest.

one accelerometer of single steps (in particular, it does not rely on the existence of longer data sequences, Neugebauer et al. [NHB12]). In the following, the method and results are briefly outlined.

Standardized gait tasks were recorded of a mixed group of 26 subjects (12 male, 14 female). 10-meter straight walk, turn-around, walk back to start. Each subject was asked to walk naturally and repeat the task four times. Four inertial sensors (APDM Opal wireless inertial measurement units) placed at the wrist, the ankle, the lower back and the chest recorded the performance of each subject. Each unit measured tri-axial acceleration as well as tri-axial angular velocities. After filtering, the input data are segmented temporally into representations of single steps. Note that this was done by detection of peaks and valleys in the signal ([LZD$^+$12, DNBB10, Zij04]). Based on these isolated steps, extraction of several features was performed. The complete list of features contains 50 items in total. Some of them are statistics related such as step length, step duration, average, standard deviation, global minimum, global maximum, root mean square, and entropy. Some more are energy and maximum amplitude of frequency spectrum.

Excerpts of the above-mentioned features are used to train a classifier (Random Forest) in order to estimate gender, height and age of participants.

Modern machine learning techniques like decision trees can target pattern recognition and prediction tasks based on many different representations of motion (Brand et al. [BOP97], Bao et al. [BI04], Kwapisz et al. [KWM11]). We used random forests, a learning method based on the construction of multiple decision trees which can be used for classification as well as regression tasks. While learning predictive models by using decision trees on their own may result in over-fitting to a training set (Phan et al. [Pha14]), random forests are less prone to this problem. For an overview of random forests, refer to the works of Breimann [Bre01] or Liaw and Wiener [LW02].

Per classification task and sensor, the classifier was trained and validated for three different sets of features:

1. 3-d accelerations (26 features),
2. 3-d angular velocities (26 features),
3. 6-d accelerations and angular velocities (50 features).

A 10-fold cross validation as well as a subject-wise cross validation (making sure that the attributes were trained rather than the subject). Classification rate, specificity, sensitivity and positive predictive value (PPV) were calculated for each feature set.

In sum, high accuracy values were achieved in all classification tasks. Thus, it was proven that the method is able to recover soft biometrics information consistently at various sensor positions.

Part III

Synthesis of Motion Data

CHAPTER 8

SYNTHESIS OF MOTION DATA

Art is science made clear.

Jean Cocteau

Synthesis is defined as the act of combining given elements to form something new. Synthesis of motion data can be defined as the creation of artificial time series from existing data. This definition is independent of the type of motion data at hand. Any recordings of motion ranging from optical motion capture (with or without markers) to accelerations, gyroscopes, video, electromyography etc. may be re-arranged and re-combined forming some new motion data.

One popular example is re-use of data sets gained by optical motion capture. These data sets can be used in order to create more motion capture data by re-combining particular motions. This is discussed for optical motion capture data in the works of Krüger [Krü12]. This can be done in order to create entirely new material or to complete data sequences from sparse (key frames) input. This is especially interesting since optical motion capture and data cleaning are costly and time-consuming procedures. We have already seen a related approach in Section 2.4 where symmetry of motions was discussed. Even though the symmetry was not used in the context of a synthesis application, it obviously provides a simple way of enhancing data sets by symmetrical motion.

When a synthesis task is solved, there is often a segmentation and analysis task involved before the actual synthesis can take place (cf Kwon and Shin [KS05]). This is why synthesis is described as a last step in a greater pipeline given in Figure 1.1. Given segmented motion, the data units can be used in order to create a motion model. Chapter 10 will discuss an example how this can be done by creating a motion graph from segmented motion capture data. Based on this motion graph, new material can be synthesized by taking any possible walk on this motion graph.

When we combine segments of motion we have already accommodated in a model like a motion graph, we may want the new motion to suffice given con-

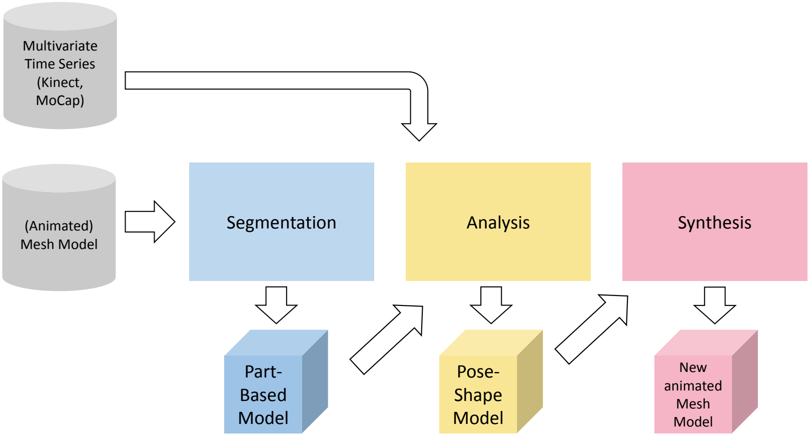


Figure 8.1: Synthesis pipeline combining different data types in order to create new animated input.

straints such as length, duration, number of stops, order of different activities, number and kind of repetitions, occurrence of particular actions, management of obstacles/collision detection. Examples of example-based synthesis with multiple constraints can be found in the works of Hartmann et al. [HTK+15].

The examples described above characterize cases where synthesis of one type of data results in data of the same type (see Figure 8.2).

A slightly different notion of motion synthesis is given when data of different types are combined in order to form new motion material.

For example, static mesh models or characters can be combined with animated input. This type of animation, character rigging or steering, is common in the entertainment sector. This naturally extends to animated mesh models, i. e. sequences of the same mesh model changing over time, e. g. from a movie (see Figure 8.1). These models can be re-combined with motion data from other sources. Chapter 9 will discuss how animated meshes can be combined with motion capture data and Kinect data to create new animations. It will also be explained how to transfer different styles from a motion (given by MoCap data) to an articulated character in this process.

In summary, there are two different notions of data-driven motion synthesis:

(S1) Rigging static or animated mesh models by time series input thus creating new animated mesh models (Chapter 9, see Figure 8.1 for overview).

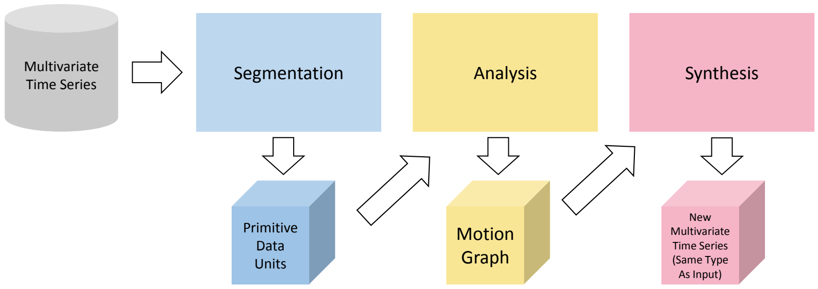


Figure 8.2: Synthesis pipeline re-using multivariate time series input in order to create new motion data of the same type.

(S2) Creating new time series from existing time series data (Chapter 10, see Figure 8.2 for overview).

Examples of each of each of the two will be discussed in the following chapters.

CHAPTER 9

STEERING ANIMATIONS

Creativity takes courage.

Henri Matisse

This chapter describes a synthesis technique in the sense of Item S1 on the list given in Chapter 8.

Creating geometrically detailed mesh animations is an involved and resource-intensive process in digital content creation. In this work we present a method to rapidly combine available sparse motion capture (MoCap) data with existing mesh sequences to produce a large variety of new animations. We show the effective transfer of realistic movement data from very few MoCap markers to a specific mesh animation sequence without the need for a full kinematic skeleton.

The key idea is to model shape changes correlated to the pose of the animated object via a part-based statistical shape model. We observe that compact linear models suffice for a segmentation into nearly rigid parts. The same segmentation further guides the parameterization of the pose which is learned in conjunction with the marker movement.

Together, the shape and pose model enable the synthesis of characteristic and detailed shape deformations already from very sparse marker motions. We show that predicting shape deformations from examples in this way presents an attractive alternative to traditional blend skinning approaches in animating articulated models. Besides the inherent high geometric detail, further benefits of the presented method arise from its robustness against errors in segmentation and pose parameterization. Therefore, nearly rigid parts of the object are identified by entropy minimization in the shape model which in turn is complemented by a simple and robust parameterization of the pose based on the detected parts. Once the shape and pose model is established, reconstruction of the mesh animation from sparse marker data can be performed in realtime. Due to efficiency of both learning and synthesis phase, our model allows to interactively steer virtual avatars based on few markers extracted from video data or input devices like the Kinect

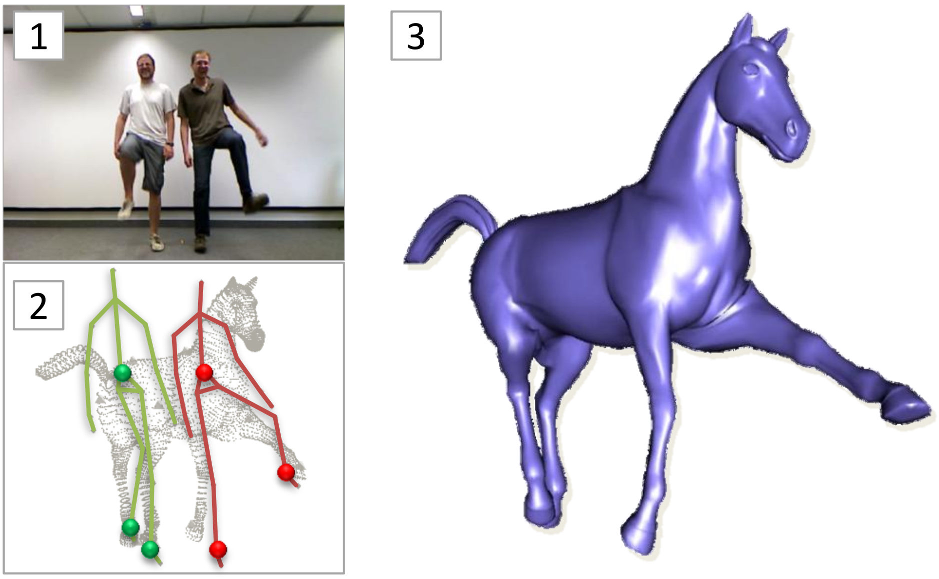


Figure 9.1: Based on Kinect input (1,2) a novel animation of an existing mesh sequence is created (3). Three markers per person are sufficient such that each person controls a pair of legs.

sensor.

A small set of optical motion capture markers or positional data captured with consumer devices like Microsoft Kinect are sufficient to steer mesh animations. We envision two main application areas: Firstly, our method enables new animation modeling paradigms which allows animators to perform deformations on a sequence by intuitive gestures. For example, new walking styles for a virtual character can be posed via simple gestures. To demonstrate this, we present examples of style transfer between different equine gaits and human locomotion. Secondly, interactive applications such as controlling virtual avatars in computer games are facilitated.

Inspired by previous work in the field of modifying mesh sequences we found that, commonly, sophisticated input is required to produce variations of a mesh animation, either in terms of additional mesh animations or registered high quality scans or detailed manual user input. This motivates the question as to what is the sparsest user input that still allows for concerted creation of novel animations. This topic has been investigated in a different context from several specific angles: Tena et al. [TDlTM11] and Weise et al. [WBLP11] concentrate on facial animation while Huang et al. [HZY$^+$11] focus on hand deformations. These works are complemented by the presented motion transfer system to steer articulated full body movement.

We come up with a part-based model, that distinguishes the mesh into limb and body parts. This separation allows a concise and robust parameterization of shape (per part) as well as pose (as relationship between parts). Pose and shape analysis produce low dimensional parameterizations for each part which in turn are connected via regression functions. The final synthesis step is formulated as a shape optimization problem that moderates between pose and shape predictions.

Aside from direct motion transfer our method is also capable of semantic deformation transfer [BVGP09]. By adapting the marker input carefully to the example mesh, differing bone length and movement styles can be compensated. Furthermore, since parts are treated individually, articulated parts of the input can be mapped arbitrarily to mesh parts as in the example shown in Fig. 9.1. This for example allows to steer the four legs of a horse mesh by two human actors in a side-by-side setup.

There is a cardinal advantage which emerges with our approach: The proposed method closes the information gap between the sparse input marker data and the resulting detailed shape deformation. By resorting to example deformations of shape parts to train a part-based shape model, the following important properties are ensured:

In summary, our method meets the following requirements for an interactive motion transfer system:

1. The motion style of the MoCap input is preserved.

2. Synthesis of poses not contained in the training sequence is possible.

3. Faithful reproduction of characteristic shape deformations from mesh examples.

4. Analysis is efficient (within minutes) and synthesis is interactive.

9.1 Related Work

Working with mesh sequences has recently become an active field of research with two major trends. Trend one is to perform high level editing of mesh sequences, e.g. by transferring the global pose to different models [SP04, BVGP09], combining this with mesh editing approaches [XZY$^+$07, FB11] or modifying the sequence in an abstract 2-D parameterization [CH12]. A second trend are data-driven deformation and enveloping approaches, e.g. in cloth simulation [KG08], anatomy-based animation [WPP07] and most recently in wrinkle synthesis of human hands [HZY$^+$11].

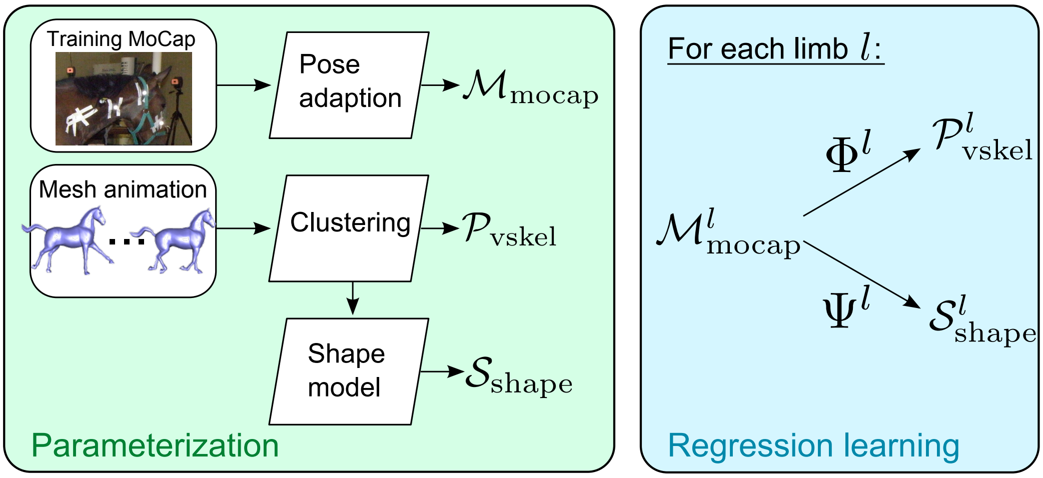

Training MoCap
Pose adaption
$\mathcal{M}_{\mathrm{mocap}}$
Mesh animation
Clustering
$\mathcal{P}_{\mathrm{vskel}}$
Shape model
$\mathcal{S}_{\mathrm{shape}}$
Parameterization
For each limb l:
Φ^l
$\mathcal{P}^l_{\mathrm{vskel}}$
$\mathcal{M}^l_{\mathrm{mocap}}$
Ψ^l
$\mathcal{S}^l_{\mathrm{shape}}$
Regression learning

Figure 9.2: Analysis phase of presented method (Sec. 9.3).

Data-driven deformation In [HZY$^+$11], high quality 3D scans of human hands serve as training examples for wrinkle deformation with respect to hand poses. Huang et al. perform non-linear regression between a sparse set of control points and normal displacement maps derived from the scans. By restricting the influence region of each control point, a geometrically local deformation model is trained. In comparison with global methods such as [WL08], better extrapolation qualities are observed on datasets where the examples do not cover the full range of poses.This result is part of the motivation for the part-based design of our approach. Wang et al. [WPP07] propose to replace traditional linear blend skinning by shape spaces learned from examples. Their approach is based on deformation gradients [SP04] trained on anatomy-based deformation examples. Contrary to our approach, correct kinematic skeletons for the examples are available, and the focus of [WPP07] is on quality, as opposed to robustness or sparse marker input.

Editing of mesh sequences Deformation transfer [SP04, BSPG06] is a popular editing method for meshes, extended to mesh animations by Xu et al. [XZY$^+$07]. The general approach is based on deformation gradients representing triangle rotations between compatible meshes. Vertex positions can be reconstructed from interpolated deformation gradients by solving a Poisson problem. Applying deformation gradients of one mesh animation to a different mesh, Pose transfer is facilitated. However, changing the style of motion is not possible.

Automatic rigging/skinning Automatic skeleton rigging and skinning approaches such as [JT05, BP07] also take additional shape information into account, although no shape models are employed. Example meshes are solely used to train blend skinning approaches, either by fitting skeletons to or by deriving blending weights from meshes. In [BP07], a kinematic skeleton is fitted robustly into a single static mesh which can in turn be animated. On the other hand, De Aguiar et al. [DATTS08] show how to fit a kinematic skeleton to a mesh animation. In [JT05] a large number of virtual bones with skinning weights is derived from clustering triangle rotations, leading to a hardware efficient representation. The recent work by Jacobson et al. [JBK$^+$12] automatically finds shape-aware blend functions which allows animation of the object based on sparse user handles.

9.2 Overview of the Method

We come up with a part-based model that distinguishes the mesh into limb and body parts. This separation allows a concise and robust parameterization of shape (per part) as well as pose (as relationship between parts). Pose and shape analysis produce low dimensional parameterizations for each part which in turn are connected via regression functions. The final synthesis step is formulated as a shape optimization problem that moderates between pose and shape predictions. The whole approach decomposes into an analysis phase and a synthesis phase, illustrated in Figures 9.2 and 9.3.

9.2.1 Separation of body and limbs

Distinction between *limbs* and *body* of a given model makes sense due to the following observation. Suitable input marker sets are required to provide information unique to articulated poses. These are typically given by positions of feet and hands (humans), hooves, wings or ears (animals) in relation to a reference (e.g. hip or spine) within the given body. In the context of searching human motion databases, Krüger et al. [KTWZ10] have successfully restricted to such information as well. Consequently, *limbs*, i.e. articulated parts, are analyzed independently of one another and controlled directly by the input markers. Hence application of shape deformations to limbs happens regardless of the global pose of the examples. This enables the synthesis of novel poses not possible with related approaches.

Transfer of different equine gaits to a given canter sequence (discussed in more detail in Section 9.5) demonstrates the benefits of the technique.

The marker data considered here usually stems from the movement of articulated bodies. In the sparsest setting there has to be at least one marker on each end-effector and some few markers identifying the remaining non-articulated parts. For a human actor the minimum requirements consists of markers at hands and feet and some markers on the body, to be able to infer the articulated pose. These

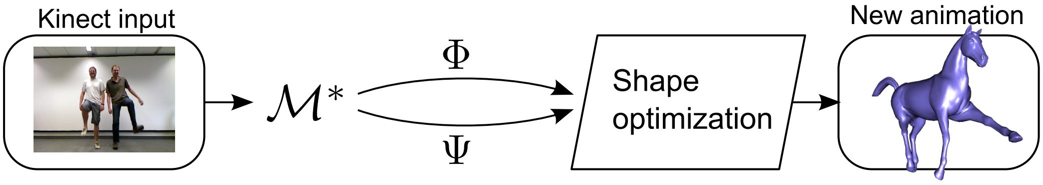


Figure 9.3: Synthesis phase of presented method (Sec. 9.4).

basic requirements have also been successfully used in other contexts. Krüger et al. [KTWZ10] compared different feature sets for searching human motion databases. They conclude that a feature set containing only relative positions of the hand, feet and head is sufficient to describe human motions. Tautges et al. [TZK$^+$11] were able to reconstruct full body animations from accelerometer readings of hand and feet only.

Consequently, *limbs* refers to the articulated parts which are directly controlled by the input markers and modeled independently of one another. On the other hand, *body* will denote the remaining, non-articulated parts.

Throughout the principal stages of our approach we focus solely on the limbs which are modeled independently. Treating the limbs independently capacitates learning shape deformations of limbs regardless of the global pose of the examples. This enables the synthesis of novel poses which is not possible with related deformation transfer approaches. One example is the transfer of different gaits to a given equine sequence discussed in more detail in Section 9.5.

In the analysis phase regression models Φ and Ψ for each limb are learned on an input mesh animation and a training MoCap sequence (Sec. 9.3). Given (novel) marker input these models predict shape and pose of each individual limb in the synthesis part. The resulting synthesized limbs are jointly aligned with the *body* parts while optimizing the shape parameters to best conform to the predicted shape at limb intersections (Sec. 9.4). The resulting mesh will thus exhibit the most probable shape according to the part-shape model with respect to a given limb configuration.

9.3 Learning a Combined Model for Pose and Shape

The combined model of pose and shape has to relate marker positions to mesh pose and shape. To this end suitable representations of marker input and mesh pose are required. A linear shape model computed on nearly rigid clusters of the mesh delivers a compact set of shape parameters (Sec. 9.3.4). Marker input, pose and shape of each limb l are linked by regression functions Φ^l and Ψ^l as

$$\mathcal{M}^l_{\text{mocap}} \xrightarrow{\Phi^l} \mathcal{P}^l_{\text{vskel}} \quad \text{and} \quad \mathcal{M}^l_{\text{mocap}} \xrightarrow{\Psi^l} \mathcal{S}^l_{\text{shape}} \tag{9.1}$$

where $\mathcal{M}^l_{\text{mocap}}$ and $\mathcal{P}^l_{\text{vskel}}$ are the parameter sets for pose and $\mathcal{S}^l_{\text{shape}}$ for shape, respectively (Sec. 9.3.3).

Pose parameters are derived from a *marker graph* capturing the topology of the MoCap marker setup and a *virtual skeleton* derived from the clustered mesh (Sec. 9.3.1), where the same clustering as for the shape model is used (Sec. 9.3.5). User specified correspondences from markers to specific mesh vertices connect both representations. Note that the marker graph is usually very sparse and its nodes do in general not correspond to joints. The purpose of Φ is to relate the sparse marker input to the virtual skeleton. The virtual skeleton on the other hand has nodes at intersections of nearly rigid parts of the mesh and thus reflects the joint analogy. Mindful parameterization of pose information based on the marker graph and the virtual skeleton is required to achieve robust regression results later on (Sec. 9.3.1).

The parameterization $\mathcal{M}^l_{\text{mocap}}$ is based on quantity observations intrinsic to the sparse marker graph, like edge length or angles between edges. Since the denser virtual skeleton employs virtual joints, quaternions are used in $\mathcal{P}^l_{\text{vskel}}$ to encode local orientation.

Input to the training phase are m marker and n mesh vertex trajectories given as

$$\boldsymbol{m}_i : \{1, \ldots, F_M\} \to \mathbb{R}^3 \tag{9.2}$$

for $i = 1, \ldots, m$ and

$$\boldsymbol{v}_j : \{1, \ldots, F_N\} \to \mathbb{R}^3 \tag{9.3}$$

with $j = 1, \ldots, n$ where F_M denotes the number of MoCap frames and F_N denotes the number of mesh animation frames. Additionally, the user selects a corresponding vertex for each marker which results in a correspondence map

$$\tau \colon \{1, \ldots, m\} \to \{1, \ldots, n\}. \tag{9.4}$$

9.3.1 Pose Representation and Parameterization

Pose parameters $\mathcal{M}^l_{\text{mocap}}$ are derived from the marker graph $G = (M, E_G)$ consisting of the used markers $M = \{1, \ldots, m\}$ and edges $E_G \subset M^2$, roughly describing limb and body topology. The body is identified with a reference edge $e_{\text{ref}} \in E_G$, whereas limbs are simple paths in G. Considering the sparsest MoCap

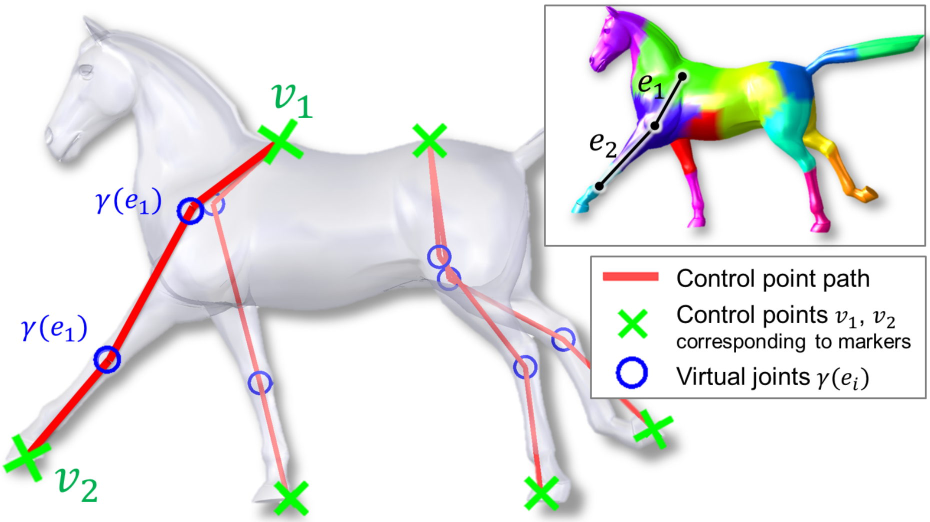

v_1
v_2
$\gamma(e_1)$
$\gamma(e_1)$
e_1
e_2
Control point path
Control points v_1, v_2 corresponding to markers
Virtual joints $\gamma(e_i)$

Figure 9.4: Control points for the equine model. Green control points correspond to MoCap markers. Virtual joints (blue) are given at cluster intersections based on the clustering shown in the inset. Configuration of a single limb controlled by two markers is highlighted while the inset shows the corresponding edges $\{e_1, e_2\}$ of the virtual skeleton.

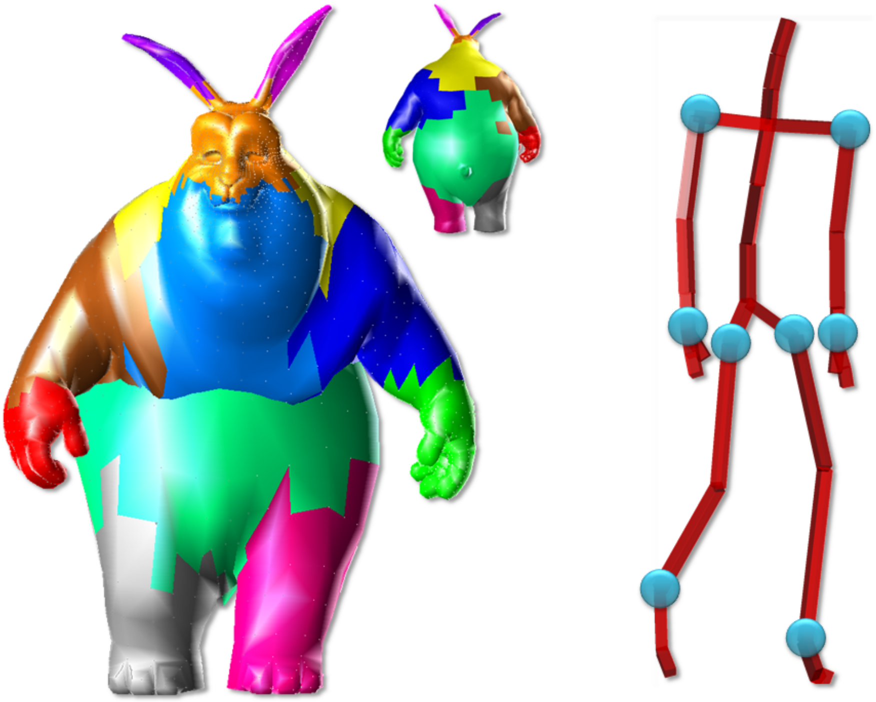


Figure 9.5: *(Left:)* Automatic clustering of the bunny mesh into 13 clusters. *(Right:)* Human MoCap skeleton with used control points highlighted.

setting with one marker per end-effector, each limb path p_G will consist of a single edge (m_1, m_2) between two markers. Limb orientation is later parameterized relative to e_{ref}. For humans and animals e_{ref} is usually given by markers placed near the backbone, but in general, any nearly rigid edge can serve as reference edge.

On the mesh, a virtual skeleton graph $H = (C, E_H)$ is defined capturing adjacency of the mesh clusters and driving the parameterization $\mathcal{P}^l_{\text{vskel}}$. Nodes in $C = \{1, \dots, k\}$ represent clusters while $E_H \subset H^2$ contains undirected edges for adjacent clusters. Since the mesh is dissected into (nearly) rigid parts the clustering reflects, to some extent, the bone and joint semantics of common animation skeletons.

In order to parameterize the orientation between two clusters, *virtual joints* J are introduced at the barycenters of the cluster intersections. This is accomplished by attaching a virtual joint

$$\gamma : E_H \to J, \tag{9.5}$$

to each edge.

Limbs in H are characterized by extracting a limb path from the marker graph G as follows. For each limb path $p_G = (m_1, m_2)$ in G we define a set of *control points*

$$p^l_{\text{ctrl}} = (v_1, \gamma(e_1), \dots, \gamma(e_{|p^l_{\text{ctrl}}-2|}), v_2) \tag{9.6}$$

where vertices $v_1 = \tau(m_1)$ and $v_2 = \tau(m_2)$ correspond to markers m_1 and m_2. Intermediate virtual joints $\gamma(e_i)$ are defined by edges e_i along the shortest path in H between the clusters containing v_1 and v_2, see Fig. 9.4 for an illustration.

Pose Parameterization The pose parameters in $\mathcal{M}^l_{\text{mocap}}$ and $\mathcal{P}^l_{\text{vskel}}$ are outlined in Table 9.1. Rotation angles and edge lengths are derived from each limb path p_G within the marker graph. Based on the denser set of control points (9.6) in the virtual skeleton, unit quaternions representing local rotations can be computed. The direction of movement of the associated markers m_i and vertices v_i, computed via forward differences between the frames $f + 1$ and f

$$d_{m_i}(f) = \frac{m_i(f+1) - m_i(f)}{\|m_i(f+1) - m_i(f)\|} \tag{9.7}$$

and similarly for d_{v_i}, is considered for both parameter sets. Velocity would provide an alternative parameter but is less robust due to different sampling rates and movement styles of mesh animation and marker input.

For $\mathcal{M}^l_{\text{mocap}}$ it is important to take edge lengths $\|e_j\|$ into account since we do not require any markers near real joint positions. Thus, the edges in the marker graph do not correspond to otherwise employed bones in animation skeletons. Accordingly, a change in length of these edges is a strong indicator for a possible bend of in-between joints which are available in the virtual skeleton.

Parameters for marker graph $\mathcal{M}^l_{\text{mocap}}$	
α_1	Angle between the e_{ref} and the first edge in p_G.
α_j	Angles between successive edges in p_G (for $j > 1$).
$\|e_j\|$	Lengths of edges in p_G.
d_{m_i}	Direction of movement of marker m_i.

Parameters for virtual skeleton $\mathcal{P}^l_{\text{vskel}}$	
q_1	Quaternion rotating between the mesh vertices corresponding to e_{ref} and the first edge $(v_1, \gamma(e_1))$ in p^l_{ctrl}.
q_j	Quaternion rotating between successive edges in p^l_{ctrl}.
d_{v_j}	Direction of movement of control point v_j.

Table 9.1: Pose parameterization

9.3.2 Pose Adaption

As implied, we do not expect the settings of marker input and and mesh model to be the same in an anatomical sense, nor do we require an overall equivalence of proportions between the two. Also, poses of animation input do not necessarily occur in the original mesh sequence. Contrarily, within a certain range, they are the basis on which new motion styles will be trained.

This requires sensible pose adaption as preparation for further computations. To begin with, a best-frame fit between mesh sequence and input marker sequence is key to successful training. A mapping

$$f_l^* \colon \{1, \dots, F_M\} \to \{1, \dots, F_N\} \tag{9.8}$$

fits the former setting to the latter according to rotation angle conformity. That is angles θ_m, θ_v between (m_1, m_2) and (v_1, v_2) on each limb path are considered and we minimize

$$f_l^*(j) := \min_i \Big(|\Delta_m(j) - \Delta_v(i)| + \frac{\sigma}{2}(1 - \text{sgn}(\Delta_m \Delta_v)) \Big) \tag{9.9}$$

where $\Delta_m(f) = \theta_m(f+1) - \theta_m(f)$, Δ_v accordingly and $\sigma = \text{stdev}(\theta_m)$.

If limb movement of MoCap and mesh sequence are similar alternatively a global least squares fit can be used as well:

$$f^*(j) := \min_{f^*} \sum_{i=1}^{m} \left\| v_{\tau(i)}(f^*) - m_i(j) \right\|^2 \tag{9.10}$$

Furthermore, the total and local ranges of motion between the settings will severely differ as exemplified by comparison of leg rotation angles in different equine gaits. To allow reasonable motion transfer, the variances within both sets need to agree. As a matter of fact, so should the variances of other corresponding parameter sets such as the lengths of graph edges in G and M. Meeting both above conditions calls for inverse kinematics to restore relative positions correctly.

9.3.3 Pose and Shape Regression

Training the relationship between parameterized pose information of the original input $\mathcal{M}^l_{\text{mocap}}$ and corresponding pose parameters $\mathcal{P}^l_{\text{vskel}}$ will achieve meaningful pose transfer. Multivariate multiple regression serves well as a training mechanism for Φ^l in (9.1) and supports interpolation between trained motion parameters [Ren03]. Posed as a multivariate least squares problem

$$\min_{\Phi^l} \left\| \Phi^l \cdot \begin{bmatrix} \mathcal{M}^l_{\text{mocap}} \\ 1 \end{bmatrix} - \mathcal{P}^l_{\text{vskel}} \right\|^2_{\text{Frob}} \tag{9.11}$$

this means optimizing a prediction $\Phi^l \in \mathbb{R}^{s \times s'}$ with s, s' the respective dimensions of the parameter sets. The underlying statistical relationship between the same parameter input $\mathcal{M}_{\text{mocap}}$ and the part-shape information $\mathcal{S}^l_{\text{shape}}$ establishes the *shape* characteristics Ψ^l of the output animation in the same fashion.

$$\min_{B^l_q} \left\| \mathcal{S}^l_{\text{shape}} \cdot B^l_q - \mathcal{M}^l_{\text{mocap}} \right\| . \tag{9.12}$$

Deriving such information from the relationship of pose-regression results as

$$\Psi^l_2 \colon \mathcal{P}^l_{\text{vskel}} \mapsto \mathcal{S}^l_{\text{shape}}$$

will also serve, but is observed to accumulate inaccuracy.

Kernels and Canonical Correlation Straightforward linear regression imposes limitations on the nature and extent of information which can be trained. Since the actual relationship between given parameter sets need not be linear, using statistical based methods that allow non-linear generalization such as canonical correlation analysis is better suited.

9.3.4 Part-Shape Model

A separate linear shape model [BV99] is established for each cluster of the mesh. Given n cluster vertices with index set $\{i_1, \ldots, i_n\}$ as a matrix

$$X = \begin{bmatrix} v_{i_1}(1) & \cdots & v_{i_1}(F_N) \\ \vdots & \ddots & \vdots \\ v_{i_n}(1) & \cdots & v_{i_n}(F_N) \end{bmatrix} = [\mathbf{x}_1 \cdots \mathbf{x}_{F_N}] \in \mathbb{R}^{3n \times F_N},$$

a set of eigenshapes V is derived via eigendecomposition of the scatter matrix $\frac{1}{F_N - 1}(X - \overline{X})(X - \overline{X})^T = V\Sigma^2 V^T$ with $\Sigma^2 = \text{diag}(\sigma_1^2, \ldots, \sigma_{F_N}^2)$. This leads to a linear shape model

$$\hat{\mathbf{x}} = \overline{\mathbf{x}} + V\lambda \tag{9.13}$$

where $\overline{\mathbf{x}} = \sum_i^{F_N} \mathbf{x}_i$ is the mean shape and $\lambda = (\lambda_1, \ldots, \lambda_k)^T$ are shape parameters. The Gaussian model assumption underlying (9.13) implies that λ_i should be normally distributed according to the shape variances σ_i^2.

Shape Parameterization. In the analysis phase, we employ the shape model (9.13) for parameterization while, throughout the shape optimization it is used as described in Sec. 9.4.2. In analysis, Eq. (9.13) provides a sparse and concise parameterization of the shape variation of the near rigid clusters.

Before the shape parameters $\mathcal{S}_{\text{shape}}$ can be derived, the mesh sequence has to be brought into a common coordinate system. This is done by rigidly aligning all mesh frames against an (arbitrarily chosen) reference frame [Gow75]. Afterwards, principal modes V of shape variation are computed. Due to clustering into nearly rigid parts, the shape variation of a single part can be compactly described by few linear modes. In our experiments we observe that the first two modes explain almost always more than 95 percent of shape variability. Projection of an aligned shape x into the space spanned by the first k modes $V_{1\ldots k}$ results in k shape coefficients:

$$s = (s_1, \ldots, s_k)^T = V_{1\ldots k}^T(\mathbf{x} - \overline{\mathbf{x}}), \quad s \in \mathcal{S}_{\text{shape}}$$

9.3.5 Clustering

Developing a part-based shape model depends upon a suitable method to derive cluster parts from a given mesh model. In our case, a clustering into near-rigid parts was performed by the compression method proposed by Sattler et al. [SSK05], comprising clustered PCA. This accommodates shape models of separate mesh parts with respect to trajectory resemblance. The inset in Fig. 9.4

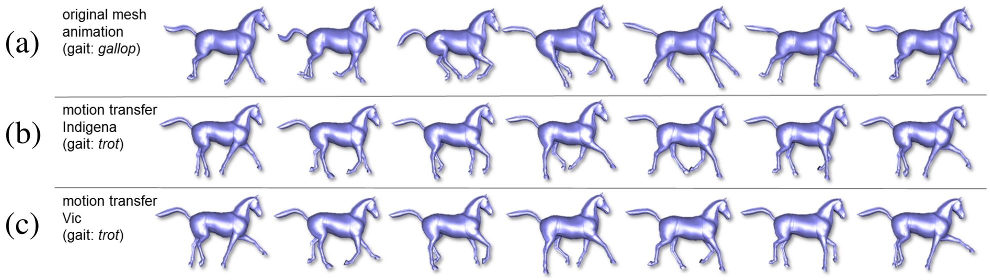

(a)
original mesh animation (gait: *gallop*)
(b)
motion transfer Indigena (gait: *trot*)
(c)
motion transfer Vic (gait: *trot*)

Figure 9.6: Motion transfer from MoCap data of horses to the motion sequence of Sumner. For comparison, frames with similar pose in right fore leg are aligned column wise. (a) Original Sumner equine sequence. (b) Motion transfer synthesized on Indigena trot sequence. (c) Motion transfer synthesized on Vic trot sequence. Note that the trot sequences are similar to each other but differ severely in motion style from the gallop of the original mesh animation.

shows an example clustering for the equine sequence. Fig. 9.5 shows an example clustering for the bunny sequence.

For later rigid alignment and shape optimization we extend the disjoint partitioning from the clustering algorithm by overlaps between adjacent clusters. Vertices in the overlaps serve as constraints in mentioned optimizations connecting adjacent cluster shapes and orientations. For this purpose it is appropriate to simply extend a disjoint partitioning by adjacent vertices along cluster intersections.

Similar results can be obtained by dissecting the given mesh into parts of similar motion relying on more complex clustering methods: Mean shift clustering on local triangle rotations was first presented in [JT05] and also used to cluster example deformations in [FKY08] and [HZY$^+$11]. In [DATTS08] a spectral clustering based on an affinity matrix measuring pairwise point distances is succesfully applied. Since our method is to some degree robust against clustering artifacts we resort to the more efficient method from [SSK05].

9.4 Synthesis of New Animations

After a short training phase, the conditioned pose and shape regression functions Φ and Ψ can be applied to new marker input.

This results in pose and shape parameters required for motion transfer. In this section we describe now the synthesis process and how to combine the parameters to achieve plausible mesh animations following the input marker motion.

The synthesis is started by computing the parameterization $\mathcal{M}^*$ from the new marker input. $\mathcal{M}^*$ is expressed in terms of angles and lengths between nodes in

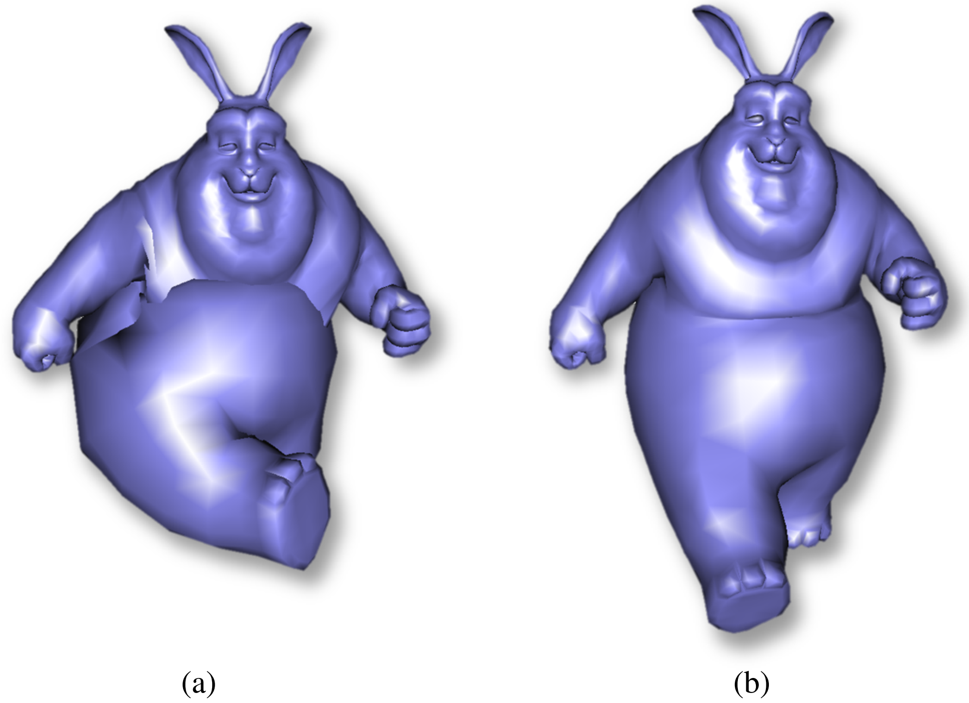


Figure 9.7: Illustration of the impact of our shape optimization on the bunny sequence. (a) Greedy alignment, (b) optimized shape.

the marker graph. Subsequently, $\Phi(\mathcal{M}^*)$ gives the correlated pose of the virtual skeleton, in terms of quaternions, whereas $\Psi(\mathcal{M}^*)$ yields the according shape parameters for the limbs.

Shape and pose parameters can be conflicting such that direct assembly of synthesized shapes rotated by predicted quaternions leads to unpleasant artifacts as shown in the inset.

In combining shape and pose predictions we face three main challenges:

(i) conflicting shape and pose parameters,

(ii) unknown shape parameters of body clusters,

(iii) mismatching vertex positions in cluster overlaps.

These issues are addressed in an iterative optimization process. Its final goal is to find a rigid alignment for all clusters as well as shape parameters that produce an artifact free mesh respecting pose and shape predictions. Simultaneous optimization of both rigid alignment and shape parameters leads to a non-linear problem. Similar to [XZY$^+$07, BBW$^+$11] we approach the solution by an alternating least squares method. It efficiently solves for a rigid alignment (Sec. 9.4.1) keeping the shape parameters fixed and vice versa.

Shape optimization is initialized with limb parts synthesized according to pose and shape prediction. During shape optimization, the shape of the limbs is constrained to stay close to the predicted shape parameters.

The same holds for the pose such that quaternion rotations around limb joints are preserved, facilitating motion transfer. Body parts are initialized with their respective mean shapes.

Executing the shape optimization for a few iterations will return a consistent alignment and shape parameters for all clusters (Sec. 9.4.2). After blending multiple occurring clusters (Sec. 9.4.3) the final output mesh is assembled.

9.4.1 Greedy Rigid Alignment

Rotation and translation have to be estimated simultaneously for all mesh clusters. Avoiding an intricate global solution we greedily align pairwise adjacent clusters according to their overlap. Starting with the largest cluster, the adjacent cluster with most overlapping vertices is chosen. The two clusters are rigidly aligned [Gow75] and merged into a super-cluster for further alignment. Repeating this greedy procedure will finally return a single super-cluster with all mesh clusters consistently aligned. Keeping track of rotations and translations in this process a rigid transformation for each mesh cluster is found.

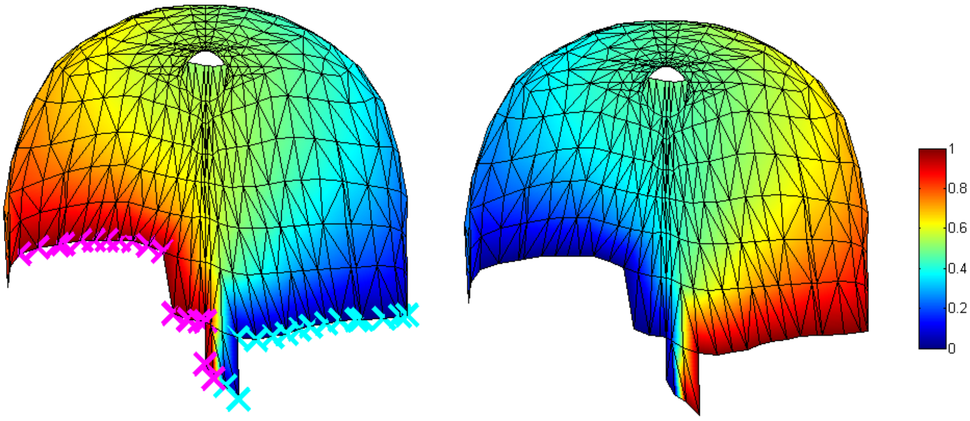
9.4.2 Shape Optimization

Given a set of prescribed vertex coordinates $\{x_i\}$, $x_i \in \mathbb{R}^3$ we optimize the shape $\hat{\mathbf{x}} \in \mathbb{R}^{3n}$ to match $\hat{x}_i = x_i$. As a further constraint we want the shape to be "plausible" that is force it to stay close to the Gaussian shape model. We use the formulation of Berner et al. [BBW+11] and express the shape optimization as the following minimization problem:

$$\lambda^* := \min_{\lambda} \sum_{i \in C} \|x_i - \hat{x}_i(\lambda)\|^2 + \beta \cdot \sum_{i=1}^{k} \frac{\|\lambda_i - \lambda_i^+\|^2}{\sigma_i^2} \tag{9.14}$$

which yields optimized shape parameters λ^*. The first term assures that the fixed vertices x_i in the overlap to adjacent clusters are matched while the second term regularizes the predicted shape weighted with a user parameter β. We introduce prior knowledge about "plausible" shape parameters here via λ_i^+ and penalize deviations from these. This allows the limbs to adhere to the shape predicted by $\Psi(\mathcal{M}^*)$ while body clusters are initialized with the mean shape prior $\lambda_i^+ = 0$.

Eq. (9.14) leads to an over-determined linear system $A\lambda = b$ of $3k$ equations which is solved efficiently in a least squares fashion. Note that the number of fixed



(a) (b)

Figure 9.8: Geodesic blend functions for shared cluster at rear of equine mesh. In the left image (a) boundary vertices to the two associated limbs are marked by crosses.

vertices k in the cluster overlaps is only a small fraction of the mesh vertices:

$$\frac{\partial}{\partial \lambda_j} E_{\{x,y,z\}} = 0, j = 1, \ldots, k \tag{9.15}$$

9.4.3 Blending of Shared Clusters

So far we omitted the fact that a mesh cluster can be contained in several limbs. This is a typical situation in coarse clustering where the limbs meet at a common point (e.g. the hip in a human model). We call this *shared clusters*. For the optimization process we simply duplicate shared clusters in the virtual skeleton graph and treat them independently. This is important, since the shape parameters can be conflicting for poses very different from the ones in the mesh animation. Optimized shapes of a shared cluster are blended smoothly to match the prediction of the participating limbs at the corresponding border vertices.

Simply averaging shared clusters for shape optimization would produce results close to the average of the conflicting shape parameters. Since the average is de facto the mean shape the necessary and predicted shape deformations would not be applied. Therefore we treat shared clusters separately for optimization and blend the optimized results appropriately.

Blending a cluster shared by m limbs is realized through blend functions w_i, one for each limb. At any boundary vertex v to a specific limb l weights are $w_l(v) = 1$ and $w_i(v) = 0$ for $i \neq l$.

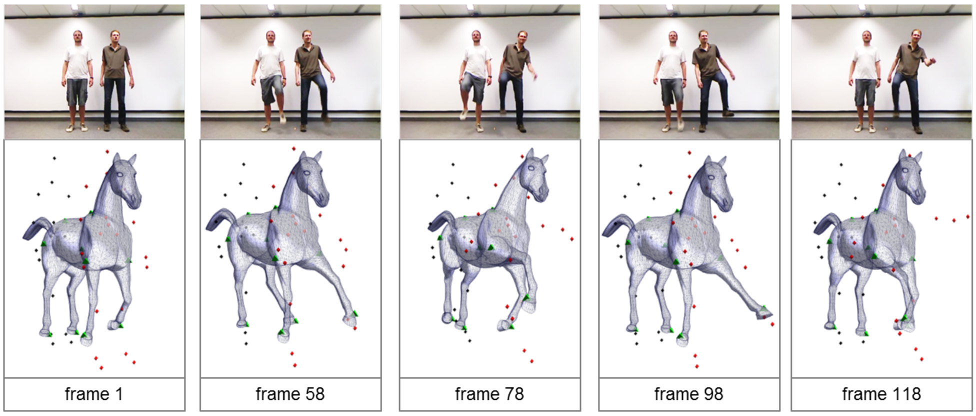


Figure 9.9: The Kinect input setting is shown in the above excerpt of animation frames - with complementing mesh sequence frames below. The latter are visualized along with motion capture markers of the training sequence. The semantic character of the transfer is demonstrated by the relatively high distances of motion capture markers from the mesh. Full set of kinect markers for both actors is shown as black and red dots. The corresponding control points on the mesh are drawn in green.

Based on minimum geodesic distance [NK02] $d_l(v)$ from vertex v to the boundary vertices of limb l, smooth linear interpolation weights are given by

$$w_l(v) = \frac{\sum_{i=1}^{m} d_i(v) - d_l(v)}{\sum_{i=1}^{m} d_i(v)} \tag{9.16}$$

Since (9.16) defines a partition of unity, the shared clusters can be linearly blended using the weights w_i. Fig. 9.8 shows an example for two limbs sharing a cluster. Note that the blending functions do not depend on the synthesis parameters and can be pre-computed.

9.5 Results

We implemented a MATLAB prototype of our motion transfer system which, though not optimized, synthesizes several frames per second.

Motion Transfer of Quadrupeds Our method manages the translation of motion attributes tracked from quadruped locomotion to a given mesh sequence. Results of training equine gait samples (trot) on a sequence of horse mesh frames displaying a different gait (canter) are shown in Fig. 9.6. As expected, they expose characteristic differences in overall pose as well as local attributes according to the pose dissimilarities of the featured gaits. As a matter of fact, these examples also display slight shape distortion within certain parts of the model, e.g. hooves. These were exhibited by original mesh samples in mild form and are exaggerated by restriction to linear regression as training method. Resorting to kernel CCA [FKY08] for training purposes should help suppress such phenomena. Synthesis on a motion sequence performed by a different individual subsequent to the above training was conducted in addition. The results constitute the third row of Fig. 9.6 and display gait style characteristics of the second individual compared to the first.

Motion Transfer of Bipeds Similarly, a biped bunny mesh was trained to perform a straightforward walk learned on an example sequence of human locomotion. Alternative synthesis on a different walking sequence involving a curve results in the correct change of walking direction and posture in the succeeding animation, please see the accompanying video. Within this context, the semantic quality of the transfer is apparent since the skeleton proportions of selected mesh and motion capture input differ considerably. All motion capture sequences were taken from the HDM05 database [MRC$^+$07].

Interactive Steering of Animations With the Kinect Since creating mesh animations from motion capture data has proved to be feasible, steering mesh animations by user interaction can be tackled as well. Kinect input of human locomotion - performed jointly by two actors - were transferred to the mesh model setup. Results are shown in Fig. 9.1 and the accompanying video.

9.6 Conclusion and Future Work

We presented an efficient way towards more effortless creation of new digital content from existing material. The key idea is combining available mesh sequences with a variety of motion sensing input. In particular, since motion sensing input devices are emerging at consumer level, the method points to new paradigms in the field. We presented a method for effective motion transfer from sparse marker input data to mesh sequences. Our pose adaption and shape optimization achieves plausible results in different scenarios. We demonstrated that the proposed method comes with a variety of applications such as semantic deformation transfer, interactive steering of mesh animations and motion style transfer to bi- and quadrupeds.

Current limitations are foot skating artifacts and lack off ground contact, which can e.g. be modeled as hard constraints in the synthesis step. So far, extrapolation quality depends on the particular input sequences. Considering more complex examples are expected to improve results, but will probably require non-linear methods in the analysis phase.

Future work will focus on generalizing the essential idea of transferring sparse input signals to part-based pose and shape models. Investigating alternative input signals, e.g. audio, is one such generalization. Employing more abstract shape spaces which model features of given data on a higher semantic level is another. The latter could involve modeling motion properties of more complex nature.

CHAPTER 10

Creating New Motion Capture Data By Motion Graph Synthesis

It has happened more than once that a composition has come to me, ready-made as it were, between the demands of other work.

Amy Beach

This chapter describes a synthesis technique in the sense of Item S2 on the list given in Chapter 8. The idea is to create new time series from existing material. In order to do so, we an make use of the preceding steps in the pipeline, namely segmentation and clustering/building a model. As an example, we use pre-computed motion primitives which can be used for motion synthesis with classical motion graph techniques or as input for statistical models as used in [MC12, KTMW08]. We have seen that with our segmentation and clustering method (see Chapter 2) we obtain motion segments that are superiorly suited for synthesis of motion by a graph-based approach.

In order to compare our technique once again to the techniques by Zhou et al. ([ZlTH08, ZlTH13]), a motion graph (see Figure 10.1) was built of motion primitives from the examples in the CMU data base (specific example: subject 86, trial 03). One graph was built of the primitives obtained by our method and one graph was built from the primitives found by the HACA-based method. The HACA-graph (seen in the lower part of Figure 10.1) contains one walk segment, all runs found in the entire sequence and the first segment after running. In our graph (seen in the upper part of Figure 10.1), there are nine primitives classified as 'run' (double step) and two transition phases. The two transition phases mark transfers between different activities, in this case from walking to running and between running and jumping. For each graph, the corresponding adjacency matrix is given at the right-hand side of both images. The green edges in the upper graph denote accessibility of primitives by similarity, i. e. possible transition from the

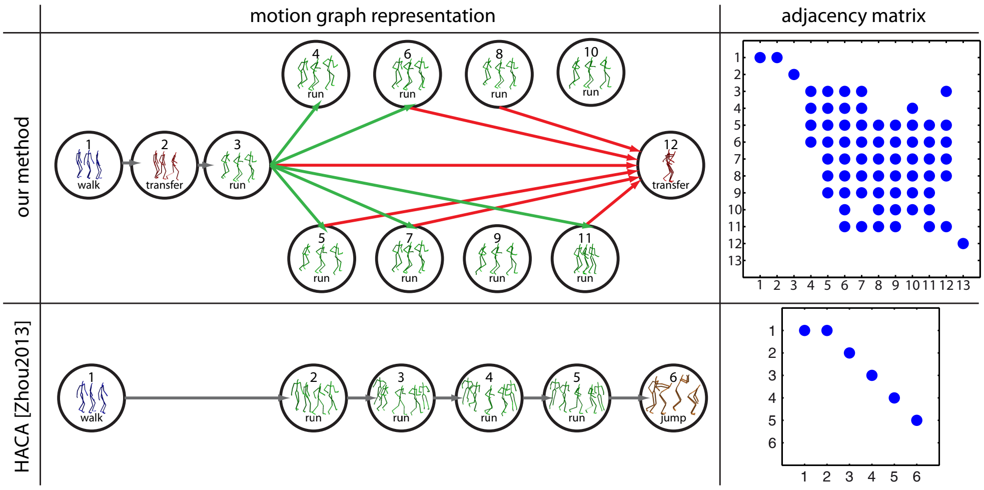


Figure 10.1: Motion graphs for synthesizing running motions based on segments computed with our and the HACA method. Upper graph: green edges show possible transitions out of first run node, red edges represent transitions to transfer node. For clarity we omit all other edges within the run cluster. The given adjacency matrix encodes this information.

end of one primitive to the beginning of another. Note that, for visual overview, some possible transitions have been left out in the graph representation. They can, however, be found as entries in the associated adjacency matrix. Red edges in the upper graph denote possible transitions to a different type of primitive (a transition to a jump sequence), i. e. ways out of the activity and into another. In this case, the full set of edges is displayed.

Both graphs demonstrate how to synthesize new motions from existing motion data by building motion graphs as introduced by Krüger [Krü12]. The difference between the two is in the underlying primitives. The HACA-induced graph has four primitives labeled ’run’ (four steps each) and only a ’jump’ as optional target node. This graph allows only for synthesis by revisiting the ’walk’ segment and then linearly traversing the other nodes. Re-combining different phases of the segmented ’run’ sequence is not possible and also, there is only one way out of it, via the last step of running from the original sequence. As opposed to this, our graph creates much more variation: Multiple ’run’ cycles may be visited from the first run node and from many of them, a transfer to ’jump’ is a valid option. Every node labeled ’run’ in our graph contains only one double-step while HACA-induced nodes contain four. Observe that all nodes in our ’run’ cluster could be accumulated to form one super-node which corresponds to a single node in a **Motion Graph++** as introduced in [MC12].

Part IV

Conclusion

CHAPTER 11

Conclusion and Future Work

A fool thinks himself to be wise, but a wise man knows himself to be a fool.

William Shakespeare

In this thesis, several different aspects of investigating motion patterns have been introduced. Examples of processing multi-variate time series were discussed within different areas, segmentation, analysis and synthesis. These were introduced as exemplary steps in a more complex data processing pipeline.

On a more local level, each area associated with one data processing steps comes with its own challenges for future work depending on the state of the art and the data of interest. In the following, there will be a brief overview on the respective matter.

First of all, although the exemplary pipeline can not claim to represent a complete outline of general data processing, one processing step has proven to be essential. Throughout the studies, it became clear that particularly segmentation techniques are indispensable for any kind of further analysis. With that idea in mind, our works have introduced novel and effective methods solving the data segmentation challenge (Part I). The techniques do not depend on one specific data type but have turned out to be quite versatile in terms of data formats and motion types. Generalizing our methods even further, e. g. to mesh sequences or data sets captured in more unconstrained surrounding would be a feasible challenge for future work.

Furthermore, a new and interesting field of research deals with the question how algorithmic segmentation relates to natural perception of motion by humans. Evaluation of the relation between algorithmically derived motion primitives and motor primitives as investigated in Kinesiology might bring interesting insight for both disciplines. Finally, the role of distinguishing different movement styles and emotions as well as the description of movement by natural language in the context of motion segmentation is a fruitful subject for further interdisciplinary research.

In the part about analysis (Part II) of multivariate time series, various techniques for multi-variate data processing were outlined. Given the broader context of data science which was addressed in Chapter 3, the goals of data analysis may differ a lot. Therefore, several applications were chosen in order to illustrate some of the techniques. As the amounts of data are rapidly growing with the development of available tools and techniques for data acquisition, challenges for future work remain dealing with the resulting information. Faster methods that support intelligent automatization will be needed in order to deal with so called *big data*.

In the part about synthesis (Part III), two different types of data driven motion synthesis have been covered. We have shown how synthesis of motion data is possible by sparse input in mixed or single modal input scenarios. The dependence of graph-based synthesis methods on a fine grained segmentation of input motion data was observable in the given examples. This confirms the role of segmentation for synthesis tasks. Challenges for future work remain extrapolation in existing data sets and also setting up more complex systems, e. g. two-actor interaction and multi-constraint synthesis.

While this shall conclude the discussion of possible future work on a more task-oriented level, the benefits of the general pipeline introduced in this thesis will be conclusively discussed. Hypothetically, if all motion data in the world could be processed in the discussed manner, what would be the positive outcome. We have seen (cf Krüger [Krü12]) how motion capture data sets can be enhanced by synthesis. In a similar fashion, it would be an easy matter to generate more data from the existing sets of general motion recordings. Given the data science problem of dealing with growing amounts of data (refer to Chapter 3), it becomes clear that merely generating even more data can not be considered a benefit per se. However, the important distinction is that what is needed is not more data but more valuable information. Hence, the idea is structuring existing data sets and inter-connecting given information (e. g. by learning meta data relations or by classification). Making data denser in this way as opposed to randomly generating more data (or insisting on generating less data) is a way to dealing with an inevitable challenge.

Finally, the proposed pipeline serves as a common thread, but can not expected to be exhaustive. Generally, it would be interesting to extend the proposed pipeline in order to accommodate other subjects. This could help relate the discussed topics to other important data processing steps such as retrieval, data management or data acquisition.

BIBLIOGRAPHY

[AA06] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[AC99] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428 – 440, 1999.

[AK14] R. Araujo and M.S. Kamel. Semi-supervised kernel-based temporal clustering. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 123–128, Dec 2014.

[AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[BAM10] S. Boudaoud, F. Ayachi, and C. Marque. Shape analysis and clustering of surface emg data. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4703–4706, Aug 2010.

[BB01] J. F. Burn and C. Brockington. Quantification of hoof deformation using optical motion capture. *Equine Veterinary Journal*, 33(S33):50–53, 2001.

[BB13] Johannes Blömer and Kathrin Bujna. Simple methods for initializing the em algorithm for gaussian mixture models. *CoRR*, abs/1312.5946, 2013.

[BBBG14] Mathieu Barnachon, SaÃŕda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. *Pattern Recognition*, 47(1):238 – 247, 2014.

[BBW$^+$11] Alexander Berner, Oliver Burghard, Michael Wand, Niloy Mitra, Reinhard Klein, and Hans-Peter Seidel. A morphable part model for shape manipulation. 2011.

[BCvdPP08] Philippe Beaudoin, Stelian Coros, Michiel van de Panne, and Pierre Poulin. Motion-motif graphs. In *Proc. of the 2008 ACM SIGGRAPH*, SCA '08, pages 117–126, 2008.

[Ber06] P. Berkhin. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71, 2006.

[BI04] Ling Bao and StephenS. Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2004.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[BKC$^+$13] Rita Borgo, Johannes Kehrer, David H.S. Chung, Eamonn Maguire, Robert S. Laramee, Helwig Hauser, Matthew Ward, and Min Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *EG STARs*, pages 39–63. EG Ass., 2013.

[BOP97] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 994–, Washington, DC, USA, 1997. IEEE Computer Society.

[BP$^+$04] Jernej Barbič, , Jia-Yu Pan, Christos Faloutsos, Jessica K Hodgins, and Nancy Pollard. Segmenting motion capture data into distinct behaviors. In *In Proc. of Graphics Interface 2004*, pages 185 – 194, May 2004.

[BP07] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. In *ACM Transactions on Graphics (TOG)*, volume 26, page 72. ACM, 2007.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BRS$^+$12] Jürgen Bernard, Tobias Ruppert, Maximilian Scherer, Tobias Schreck, and Jörn Kohlhammer. Guided discovery of interesting relationships between time series clusters and metadata properties. In *Proc. of TAVA*, pages 22:1–22:8, NY, NY, USA, 2012. ACM.

[BSL$^+$14] Joachim Bottger, Alexander Schafer, Gabriele Lohmann, Arno Villringer, and Daniel S. Margulies. Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain. *IEEE TVCG*, 20(3):471–480, 2014.

[BSPG06] Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization*, pages 357–364. Citeseer, 2006.

[BSW$^+$14] Jürgen Bernard, Martin Steiger, Sven Widmer, Hendrik Lücke-Tieke, Thorsten May, and Jörn Kohlhammer. Visual-interactive Exploration of Interesting Multivariate Relations in Mixed Research Data Sets. *Comp. Graph. Forum*, 33(3):291–300, 2014.

[BT85] VIRGINIA BRUCE and RJ Turek. Muscle fibre variation in the gluteus medius of the horse. *Equine veterinary journal*, 17(4):317–321, 1985.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[BVGP09] Ilya Baran, Daniel Vlasic, Eitan Grinspun, and Jovan Popović. Semantic deformation transfer. In *ACM Transactions on Graphics (TOG)*, volume 28, page 36. ACM, 2009.

[BvLBS09] J. Bernard, T. von Landesberger, S. Bremm, and T. Schreck. Micro-macro views for visual trajectory cluster analysis. In *Eurographics/IEEE Symposium on Visualization*, 2009.

[BVLBS11] Jürgen Bernard, Tatiana Von Landesberger, Sebastian Bremm, and Tobias Schreck. Multiscale visual quality assessment for cluster analysis with self-organizing maps. In

IS&T/SPIE Electronic Imaging, pages 78680N–78680N. International Society for Optics and Photonics, 2011.

[BW95] Armin Bruderlin and Lance Williams. Motion signal processing. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 97–104, New York, NY, USA, 1995. ACM.

[BWK$^+$13] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation. *IEEE TVCG*, 19:2257–2266, 2013.

[BWS$^+$12] Jürgen Bernard, J Wilhelm, Maximilian Scherer, Thorsten May, and Tobias Schreck. TimeSeriesPaths: projection-based explorative analysis of multivariate time series data. In *Journal of WSCG*. University of West Bohemia, Plzen, 2012.

[CCM$^+$10] TC Crook, SE Cruickshank, CM McGowan, N Stubbs, Alan M Wilson, E Hodson-Tole, and RC Payne. A comparison of the moment arms of pelvic limb muscles in horses bred for acceleration (quarter horse) and endurance (arab). *Journal of anatomy*, 217(1):26–37, 2010.

[CH12] Thomas J Cashman and Kai Hormann. A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape. In *Computer Graphics Forum*, volume 31, pages 735–744. Wiley Online Library, 2012.

[CKG$^+$92] Roger J. Cole, Daniel F. Kripke, William Gruen, Daniel J. Mullaney, and J.Christian Gillin. Technical note automatic sleep/wake identification from wrist activity. *Sleep*, 15(5):461–469, 1992.

[CMU13] CMU. Carnegie Mellon University Graphics Lab: Motion Capture Database, 2013. `mocap.cs.cmu.edu`.

[CRA$^+$13] Rubana H. Chowdhury, Mamun B. I. Reaz, Mohd Alauddin Bin Mohd Ali, Ashrif A. A. Bakar, Kalaivani Chellappan, and Tae G. Chang. Surface electromyography signal processing and classification techniques. *Sensors*, 13(9):12431–12466, 2013.

[CRW08] Suzanne Cottriall, Pattama Ritruechai, and James M Wakeling. The effects of training aids on the longissimus dorsi in the equine back. *Comparative Exercise Physiology*, 5(3-4):111–114, 2008.

[DATTS08] Edilson De Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. In *Computer Graphics Forum*, volume 27, pages 389–397. Wiley Online Library, 2008.

[DDD05] F. Desobry, Manuel Davy, and C. Doncarli. An online kernel change detection algorithm. *Signal Processing, IEEE Transactions on*, 53:2961–2974, Aug 2005.

[DG68] Charles Darwin and Asa Gray. *The variation of animals and plants under domestication / by Charles Darwin ; authorized edition, with a preface by Asa Gray.*, volume v.2 (1868). New York :Orange Judd & Co., 1868.

[Dha13] Vasant Dhar. Data science and prediction. *Commun. ACM*, 56(12):64–73, December 2013.

[DHZC] Qichuan Ding, Jianda Han, Xingang Zhao, and Yang Chen. Missing-data classification with the extended full-dimensional gaussian mixture model: Applications to emg-based motion recognition.

[DL97] Carlo J. De Luca. The use of surface electromyography in biomechanics. *Journal of Applied Biomechanics*, 13(2):135–163, 1997.

[DNBB10] M.O. Derawi, C. Nickel, P. Bours, and C. Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, pages 306–311, Oct 2010.

[EF10] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE TVCG*, 16(3):439–454, 2010.

[FB11] Stefan Fröhlich and Mario Botsch. Example-driven deformations based on discrete shells. In *Computer Graphics Forum*, volume 30, pages 2246–2257. Wiley Online Library, 2011.

[Fea06] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213, 2006.

[FKY08] Wei-Wen Feng, Byung-Uck Kim, and Yizhou Yu. Realtime data driven deformation using kernel canonical correlation analysis. *ACM Transactions on Graphics (TOG)*, 27:91, 2008.

[FLFM^{+}05] Randal C. Foster, Lorraine M. Lanningham-Foster, Chinmay Manohar, Shelly K. McCrady, Lana J. Nysse, Kenton R. Kaufman, Denny J. Padgett, and James A. Levine. Precision and accuracy of an ankle-worn accelerometer-based pedometer in step counting and energy expenditure. *Preventive Medicine*, 41(3âĂŞ4):778 – 783, 2005.

[FMKN12] Simon Fothergill, Helena M. Mentis, Pushmeet Kohli, and Sebastian Nowozin. Instructing people for training gestural interactive systems. In *CHI*, pages 1737–1746. ACM, 2012.

[GA08] Robert L Gillette and T Craig Angle. Recent developments in canine locomotor analysis: a review. *The Veterinary Journal*, 178(2):165–176, 2008.

[GFA07] G. Guerra-Filho and Y. Aloimonos. A language for human action. *Computer*, 40(5):42–51, May 2007.

[GGAM12] Theresia Gschwandtner, Johannes Gärtner, Wolfgang Aigner, and Silvia Miksch. A taxonomy of dirty time-oriented data. In *Multidisciplinary Research and Practice for Information Systems*, pages 58–72. Springer, 2012.

[GJL^{+}09] Edward Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily L.C. Shepard. Visualisation of sensor data from animal movement. *Computer Graphics Forum*, 28(3):815–822, 2009.

[GMZ14] Dian Gong, G. Medioni, and Xuemei Zhao. Structured time series analysis for human action segmentation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1414–1427, July 2014.

[Gow75] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[GZK$^+$10] M Groesel, RR Zsoldos, A Kotschwar, M Gfoehler, and C Peham. A preliminary model study of the equine back including activity of longissimus dorsi muscle. *Equine Veterinary Journal*, 42(s38):401–406, 2010.

[HB03] Mark Harrower and Cynthia A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *Cartographic Journal, The*, pages 27–37, 2003.

[HCHMT15] Junhui Hou, Lap-Pui Chau, Ying He, and Nadia Magnenat-Thalmann. Human motion capture data tailored transform coding. *IEEE TVCG*, 2015.

[Her00] Hermie J. Hermens. *European Recommendations for Surface Electromyography: Results of the Seniam Project (SENIAM)*. Roessingh Research and Development, 2000.

[HG07] Rachel Heck and Michael Gleicher. Parametric motion graphs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, pages 129–136, New York, NY, USA, 2007. ACM.

[Hil10] Cherry Hill. *101 Horsemanship & Equitation Patterns: A Western & English Ringside Guide for Practice & Show*. Storey Publishing, 2010.

[HLDlT11] M. Hoai, Zhen-Zhong Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3265–3272, June 2011.

[HMB09] Zaïd Harchaoui, Eric Moulines, and Francis R. Bach. Kernel change-point analysis. In *Advances in Neural Information Processing Systems 21*, pages 609–616. Curran Associates, Inc., 2009.

[Hol06] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE TVCG*, 12(5):741–748, Sept 2006.

[HTFF05] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[HTK$^+$15] Stefan Hartmann, Elena Trunz, Björn Krüger, Reinhard Klein, and Matthias B. Hullin. Efficient multi-constrained optimization for example-based synthesis. *The Visual Computer / Proc. Computer Graphics International (CGI 2015)*, 31(6-8):893–904, June 2015.

[HZY$^+$11] Haoda Huang, Ling Zhao, KangKang Yin, Yue Qi, Yizhou Yu, and Xin Tong. Controllable hand deformation from sparse examples with rich details. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 73–82. ACM, 2011.

[IAICftAAoSM07] C. Iber, S. Ancoli-Israel, A. Chesson, and S.F. Quan for the American Academy of Sleep Medicine. The aasm manual for the scoring of sleep and associated events: Rules, terminology and technical specifications, 1st ed. American Academy of Sleep Medicine, 2007.

[Inc] Vandrico Inc. List of wearable devices. `http://vandrico.com/database`. Accessed: 2014-09-22.

[JBK$^+$12] Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. Fast automatic skinning transformations. *ACM Transactions on Graphics (TOG)*, 31:77, 2012.

[JDLP11] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):172–185, Jan 2011.

[JM04] Odest Chadwicke Jenkins and Maja J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 56–, New York, NY, USA, 2004. ACM.

[Joh73] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

[Jol86] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[JOWKL13] Zhaojie Ju, Gaoxiang Ouyang, Marzena Wilamowska-Korsak, and Honghai Liu. Surface emg based hand manipulation identification via nonlinear feature extraction and classification. *Sensors Journal, IEEE*, 13:3302–3311, 2013.

[JT05] Doug L James and Christopher D Twigg. Skinning mesh animations. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 399–407. ACM, 2005.

[KCHP01] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296, 2001.

[KG04] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23:559–568, August 2004.

[KG08] Scott Kircher and Michael Garland. Free-form motion processing. *ACM Transactions on Graphics (TOG)*, 27:12, 2008.

[KGP02] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph.*, 21:473–482, jul 2002.

[KHG$^+$10] Daniel F Kripke, Elizabeth K Hahn, Alexandra P Grizas, Kep H Wadiak, Richard T Loving, J Steven Poceta, Farhad F Shadan, John W Cronin, and Lawrence E Kline. Wrist actigraphic scoring for sleep laboratory patients: algorithm development. *Journal of sleep research*, 19(4):612–619, 2010.

[KHP11] Kittipat Kampa, Erion Hasanbelliu, and Jose C Principe. Closed-form cauchy-schwarz pdf divergence for mixture of gaussians. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2578–2585. IEEE, 2011.

[KOF05] Adam G. Kirk, James F. O'Brien, and David A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2005*, pages 782–788, jun 2005.

[Krü12] Björn Krüger. *Synthesizing Human Motions*. Dissertation, Universität Bonn, April 2012.

[KS05] Taesoo Kwon and Sung Yong Shin. Motion modeling for online locomotion synthesis. In *Proc. of the 2005 ACM SIGGRAPH*, SCA '05, pages 29–38, 2005.

[KSH01] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[KTMW08] Björn Krüger, Jochen Tautges, M. Müller, and Andreas Weber. Multi-mode tensor representation of motion data. *Journal of Virtual Reality and Broadcasting*, 5(5), July 2008.

[KTWZ10] Björn Krüger, Jochen Tautges, Andreas Weber, and Arno Zinke. Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–10. Eurographics Association, 2010.

[KVH$^+$14] Björn Krüger, Anna Vögele, Lukas Herwartz, Thomas Terkatz, Andreas Weber, Carmen Garcia, Ingo Fietze, and Thomas Penzel. Sleep detection using a depth camera. *Lecture Notes in Computer Science, ICCSA 2014, Part I*, LNCS 8579:824–835, June 2014.

[KVL$^+$14] Björn Krüger, Anna Vögele, Marouane Lassiri, Lukas Herwartz, Thomas Terkatz, Andreas Weber, Carmen Garcia, Ingo Fietze, and Thomas Penzel. Sleep detection using de-identified depth data. *Journal of Mobile Multimedia*, 10:327–342, December 2014.

[KVW$^+$15] Björn Krüger, Anna Vögele, Tobias Willig, Angela Yao, Reinhard Klein, and Andreas Weber. Efficient unsupervised temporal segmentation of motion data. *CoRR, (Submitted to IEEE TPAMI, 2015)*, abs/1510.06595, 2015.

[Kwe13] Wojciech Kwedlo. A new method for random initialization of the em algorithm for multivariate gaussian mixture learning. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 81–90. Springer, 2013.

[KWM11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, mar 2011.

[KYZW14] Björn Krüger, Hashim Yasin, Rebeka Zsoldos, and Andreas Weber. Retrieval, recognition and reconstruction of quadruped motions. In *GRAPP 2014 - International Conference on Computer Graphics Theory and Applications*. SCITEPRESS, 2014.

[LF04] ChunMei Lu and Nicola J. Ferrier. Repetitive motion analysis: segmentation and event classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:258–263, Feb 2004.

[LFP09] T Licka, A Frey, and C Peham. Electromyographic activity of the longissimus dorsi muscles in horses when walking on a treadmill. *The Veterinary Journal*, 180(1):71–76, 2009.

[Lia05] Warren T. Liao. Clustering of time series data-a survey. *Pattern Recogn.*, 38(11):1857–1874, 2005.

[LM06] Guodong Liu and Leonard McMillan. Segment-based human motion compression. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 127–135, 2006.

[LMGCG12] Adolfo López-Mendez, Juergen Gall, Josep R. Casas, and Luc J. Van Gool. Metric learning from poses for temporal clustering of human motion. In *BMVC*, pages 1–12, 2012.

[LMTL03] Guy C Le Masurier and Catrine Tudor-Locke. Comparison of pedometer and accelerometer accuracy under controlled conditions. *Medicine and science in sports and exercise*, 35(5):867–871, May 2003.

[LN06] Fengjun Lv and Ramakant Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Computer Vision âĂŞ ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 359–372. Springer Berlin Heidelberg, 2006.

[LN07] Fengjun Lv and Ramakant Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[LPF04] Theresia F Licka, Christian Peham, and Alexander Frey. Electromyographic activity of the longissimus dorsi muscles in horses during trotting on a treadmill. *American journal of veterinary research*, 65(2):155–158, 2004.

[LPZ01] Theresia F Licka, Christian Peham, and Elisabeth Zohmann. Treadmill study of the range of back movement at the walk in horses without back pain. *American journal of veterinary research*, 62(7):1173–1179, 2001.

[LS13] Rongyi Lan and Huaijiang Sun. Automated human motion segmentation via motion regularities. *The Vis. Comp.*, pages 1–19, 2013.

[LW02] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[LWS$^+$15] Chee Sun Liew, Teh Ying Wah, Junaid Shuja, Babak Daghighi, et al. Mining personal data using smartphones and wearable devices: A survey. *Sensors*, 15(2):4430–4469, 2015.

[LZD$^+$12] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.

[Mar67] M.A. Mariscotti. A method for automatic identification of peaks in the presence of background and its application to spectrum analysis. *Nuclear Instruments and Methods*, 50:309 – 320, 1967.

[MBS09] Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and robust annotation of motion capture data. SCA 2009, pages 17–26, 2009.

[MC12] Jianyuan Min and Jinxiang Chai. Motion graphs++: A compact generative model for semantic motion analysis and synthesis. *ACM Trans. Graph.*, 31:153:1–153:12, November 2012.

[Mer15] Roberto Merletti. Standards for reporting emg data. *Journal of Electromyography and Kinesiology*, 25:I – II, 2015.

[MG12] Meinard Müller and Peter Grosche. Automated segmentation of folk song field recordings. In *Proceedings of the ITG Conference on Speech Communication*, Braunschweig, Germany, 2012.

[MGW09] Meinard Müller, Peter Grosche, and Frans Wiering. Robust segmentation and annotation of folk song recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740, Kobe, Japan, October 2009.

[MHK06] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.

[MM12] Volodymyr Melnykov and Igor Melnykov. Initializing the em algorithm in gaussian mixture models with an unknown number of components. *Comput. Stat. Data Anal.*, 56(6):1381–1395, June 2012.

[MMKN08] Peter McLachlan, Tamara Munzner, Eleftherios Koutsofios, and Stephen North. LiveRAC: interactive visual exploration of system management time-series data. In *Proceedings of SIGCHI*, CHI '08, pages 1483–1492, New York, NY, USA, 2008. ACM.

[MR06] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. SCA 2006, pages 137–146. Eurographics Association, 2006.

[MRC05] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24:677–685, July 2005.

[MRC$^+$07] M. Müller, T. Röder, M. Clausen, B. Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[Mun09] Tamara Munzner. A nested model for visualization design and validation. *IEEE TVCG*, 15(6):921–928, 2009.

[Muy78] Eadweard Muybridge. The horse in motion. "Sallie Gardner," owned by Leland Stanford; running at a 1:40 gait over the palo alto track. Online available: http://www.loc.gov/pictures/item/97502309/, 1878.

[MZT$^+$02] RC Murray, N Znaor, KE Tanner, RM DeBowes, EM Gaughan, and AE Goodship. The effect of intra-articular methylprednisolone acetate and exercise on equine carpal subchondral and cancellous bone microhardness. *Equine veterinary journal*, 34(3):306–310, 2002.

[NG12] Iftekhar Naim and Daniel Gildea. Convergence of the em algorithm for gaussian mixtures with unbalanced mixing coefficients. In *ICML*. icml.cc / Omnipress, 2012.

[NHB12] Jennifer M Neugebauer, David A Hawkins, and Laurel Beckett. Estimating youth locomotion ground reaction forces using an accelerometer-based activity monitor. *PloS one*, 7(10):e48182, 2012.

[NK02] Marcin Novotni and Reinhard Klein. Computing geodesic distances on triangular meshes. In *The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2002 (WSCG'2002)*, February 2002.

[NS12] Sebastian Nowozin and Jamie Shotton. Action points: A representation for low-latency online human action recognition. Technical report, 7 J J Thomson Ave, CB30FB Cambridge, UK, July 2012.

[ODK96] M. Ostendorf, V.V. Digalakis, and O.A. Kimball. From hmm's to segment models: a unified view of stochastic modeling for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 4(5):360–378, Sep 1996.

[OGB14] Olusegun Oshin, Andrew Gilbert, and Richard Bowden. Capturing relative motion and finding modes for action recognition in the wild. *Computer Vision and Image Understanding*, 125(0):155 – 171, 2014.

[OHAP12] Emil Olsen, Pia Haubro Andersen, and Thilo Pfau. Accuracy and precision of equine gait event detection during walking with limb and trunk mounted inertial sensors. *Sensors*, 12(6):8145, 2012.

[Ols95] Clark F. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21:1313 – 1325, 1995.

[Pen06] Thomas Penzel. *Sleep Laboratory.* Wiley Encyclopedia of Biomedical Engineering, 2006.

[Pha14] Thomas Phan. Improving activity recognition via automatic decision tree pruning. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 827–832, New York, NY, USA, 2014. ACM.

[PM14] Thomas Prätzlich and Meinard Müller. Frame-level audio segmentation for abridged musical works. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[Pop07] Ronald Poppe. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst.*, 108(1-2):4–18, October 2007.

[PPC$^+$03] Florence Portier, Adriana Portmann, Pierre Czernichow, Lionel Vascaut, Etienne Devin, Benhamou, Antoine Cuvelier, and Jean FranÃğois Muir. Evaluation of a portable device based on peripheral arterial tone for unattended home sleep studies. *American Journal of Respiratory and Critical Care Medicine*, 162(3):814–818, 2003.

[RA00] Yong Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 111–118 vol.1, 2000.

[Ren03] Alvin C Rencher. *Methods of multivariate analysis*, volume 492. John Wiley & Sons, 2003.

[RFF$^+$08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE TVCG*, 14(6):1325–1332, 2008.

[RGP08] Pedro Pereira Rodrigues, João Gama, and Joao Pedro Pedroso. Hierarchical clustering of time-series data streams. *Knowledge and Data Engineering, IEEE Transactions on*, 20(5):615–627, 2008.

[RPW07] Justine J Robilliard, Thilo Pfau, and Alan M Wilson. Gait characterisation and classification in horses. *Journal of Experimental Biology*, 210(2):187–197, 2007.

[RS12] Sangeeta Rani and Geeta Sikka. Recent techniques of clustering of time series data: a survey. *Int. J. Comput. Appl*, 52:1–9, 2012.

[RTKW15] Qaiser Riaz, Guanhong Tao, Björn Krüger, and Andreas Weber. Motion reconstruction using very few accelerometers and ground contacts. *Graphical Models*, 79:23–38, 2015.

[RVKW15] Qaiser Riaz, Anna Vögele, Björn Krüger, and Andreas Weber. One small step for a man: Estimation of gender, age, and height from recordings of one step by a single inertial sensor. *Sensors*, December 2015. accepted for publication.

[SAIB$^+$07] Michael H Silber, Sonia Ancoli-Israel, Michael H Bonnet, Sudhansu Chokroverty, Madeleine M Grigg-Damberger, Max Hirshkowitz, Sheldon Kapen, Sharon A Keenan, Meir H Kryger, Thomas Penzel, et al. The visual scoring of sleep in adults. *J Clin Sleep Med*, 3(2):121–131, 2007.

[SBM$^+$14] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel Keim, Thorsten May, and Jörn Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer Graphics Forum*, volume 33, pages 401–410. Wiley Online Library, 2014.

[SBVLK09] Tobias Schreck, Jürgen Bernard, Tatiana Von Landesberger, and Jörn Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, 2009.

[Sci] Science daily, big data, for better or worse: 90% of world's data generated over last two years. http://www.sciencedaily.com/releases/2013/05/130522085217.htm. Accessed: 2015-12-30.

[Sco92] David W. Scott. *Multivariate density estimation : theory, practice, and visualization.* Wiley series in probability and mathematical statistics : Applied probability and statistics section. Wiley-Interscience, New York, Chichester, Brisbane, 1992.

[SCW^{+}10] Scott Spurlock, Remco Chang, Xiaoyu Wang, George Arceneaux, Daniel F. Keefe, and Richard Souvenir. Combining automated and interactive visual analysis of biomechanical motion data. In *ISVC (2)*, volume 6454 of *LNCS*, pages 564–573. Springer, 2010.

[See91] HJ Seeherman. The use of high-speed treadmills for lameness and hoof balance evaluations in the horse. *The Veterinary clinics of North America. Equine practice*, 7(2):271–309, 1991.

[SGW13] L St. George and JM Williams. Electromyographic evaluation of approach stride, jump stride and intermediate stride in selected superficial muscles of the jumping horse: a preliminary study. *Comparative Exercise Physiology*, 9(1):23–32, 2013.

[SH07] Alla Safonova and Jessica K. Hodgins. Construction and optimal search of interpolated motion graphs. *ACM Trans. Graph.*, 26, July 2007.

[SLC04] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[SLQ^{+}14] Donghee Son, Jongha Lee, Shutao Qiao, Roozbeh Ghaffari, Jaemin Kim, Ji Eun Lee, Changyeong Song, Seok Joo Kim, Dong Jun Lee, Samuel Woojoo Jun, et al. Multifunctional wearable devices for diagnosis and therapy of movement disorders. *Nature nanotechnology*, 9(5):397–404, 2014.

[SP04] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, 23(3):399–405, 2004.

[SSK05] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Simple and efficient compression of animation sequences. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217. ACM, 2005.

[SSvD13] Didier Staudenmann, Dick F Stegeman, and Jaap H van Dieën. Redundancy or heterogeneity in the electric activity of the biceps brachii muscle? added value of pca-processed multi-channel emg muscle activation estimates in a parallel-fibered muscle. *Journal of Electromyography and Kinesiology*, 23(4):892–898, 2013.

[SZB15] Samer Salamah, Liang Zhang, and Guido Brunnett. Hierarchical method for segmentation by classification of motion capture data. In *Virtual Realities*, volume 8844 of *Lecture Notes in Computer Science*, pages 169–186. Springer International Publishing, 2015.

[TDlTM11] J Rafael Tena, Fernando De la Torre, and Iain Matthews. Interactive region-based linear 3d face models. In *ACM Transactions on Graphics (TOG)*, volume 30, page 76. ACM, 2011.

[TLZF12] Weijun Tao, Tao Liu, Rencheng Zheng, and Hutian Feng. Gait analysis using wearable sensors. *Sensors*, 12:2255–2283, 2012.

[Tro02] Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2, 2002.

[TZK$^+$11] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30:18:1–18:12, May 2011.

[VAN08] Gentiane Venture, Ko Ayusawa, and Yoshihiko Nakamura. Motion capture based identification of the human body inertial parameters. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4575–4578, aug 2008.

[VJPSK14] Mikael Vejdemo-Johansson, Florian T. Pokorny, Primoz Skraba, and Danica Kragic. Cohomological learning of periodic motions. 2014.

[VKK14] Anna Vögele, Björn Krüger, and Reinhard Klein. Efficient unsupervised temporal segmentation of human motion. In *2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, July 2014.

[VMP03] Domitilla Del Vecchio, Richard M. Murray, and Pietro Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12):2085 – 2098, 2003.

[vW05] P. Rena van Weeren. Equine ergonomics: a new era? *Equine Veterinary Journal*, 37(1):4–6, 2005.

[vWB14] P. Rena van Weeren and Willem Back. Technological advances in equestrian sports: Are they beneficial for both performance and welfare? *The Veterinary Journal*, 199(3):313 – 314, 2014.

[VWVS99] Jarke J. Van Wijk and Edward R. Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings of the Symposium on Information Visualization*, Washington, DC, USA, 1999. IEEE Computer Society.

[VZKL16] Anna Vögele, Rebeka Zsoldos, Björn Krüger, and Theresia Licka. Novel methods for surface emg analysis and exploration based on multi-modal gaussian mixture models. *PLOS ONE*, June 2016.

[WBLP11] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, page 77. ACM, 2011.

[WG11] Matthew O. Ward and Zhenyu Guo. Visual exploration of time-series data with shape space projections. *Computer Graphics Forum*, 30(3):701–710, 2011.

[WGRP13] J Williams, P Gundry, J Richards, and L Protheroe. A preliminary evaluation of surface electromyography as a tool to measure muscle fatigue in the national hunt racehorse. *The Veterinary Nurse*, 4(9):566–572, 2013.

[WJB+14] JM Williams, C Johnson, R Bales, G Lloyd, L Barron, and D Quest. Analysis of temporalis and masseter adaptation after routine dental treatment in the horse via surface electromyography. *Comparative Exercise Physiology*, 10(4):223–232, 2014.

[WL08] Yu-Shuen Wang and Tong-Yee Lee. Example-driven animation synthesis. *The Visual Computer*, 24:765–773, 2008.

[WPP07] Robert Y Wang, Kari Pulli, and Jovan Popović. Real-time enveloping with rotational regression. *ACM Transactions on Graphics (TOG)*, 26:73, 2007.

[WRDN07] James M Wakeling, Pattama Ritruechai, Sarah Dalton, and Kathryn Nankervis. Segmental variation in the activity and function of the equine longissimus dorsi muscle during walk and trot. *Equine and Comparative Exercise Physiology*, 4:95–103, 5 2007.

[WSJF09] Alan S. Willsky, Erik B. Sudderth, Michael I. Jordan, and Emily B. Fox. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems 22*, pages 549–557. Curran Associates, Inc., 2009.

[Wu83] C. F. Jeff Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11:pp. 95–103, 1983.

[WVZ+15] Nils Wilhelm, Anna Vögele, Rebeka Zsoldos, Theresia Licka, Björn Krüger, and Jürgen Bernard. Furyexplorer: Visual-interactive exploration of horse motion capture data. In *Visualization and Data Analysis (VDA 2015)*, February 2015.

[XM07] Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM.

[XZY+07] Weiwei Xu, Kun Zhou, Yizhou Yu, Qifeng Tan, Qunsheng Peng, and Baining Guo. Gradient domain editing of deforming mesh sequences. In *ACM Transactions on Graphics (TOG)*, volume 26, page 84. ACM, 2007.

[YGGU11] Angela Yao, Juergen Gall, Luc V Gool, and Raquel Urtasun. Learning probabilistic non-linear latent variable models for tracking complex activities. In *Advances in Neural Information Processing Systems 24*, pages 1359–1367. Curran Associates, Inc., 2011.

[YLY15] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Computer Vision – ACCV 2014*, volume 9007 of *Lecture Notes in Computer Science*, pages 50–65. Springer International Publishing, 2015.

[Zij04] Wiebren Zijlstra. Assessment of spatio-temporal parameters during unconstrained walking. *European Journal of Applied Physiology*, 92(1-2):39–44, 2004.

[ZKS$^+$09] Hafsa Zaneb, Verena Kaufmann, Christian Stanek, Christian Peham, and Theresia F Licka. Quantitative differences in activities of back and pelvic limb muscles during walking and trotting between chronically lame and nonlame horses. *American journal of veterinary research*, 70(9):1129–1134, 2009.

[ZKV$^+$15] Rebeka Zsoldos, Björn Krüger, Anna Vögele, Ulrike Schröder, and Theresia Licka. Variability of the occurrence of peak muscle activity of the equine longissimus dorsi during trot. In *Progress in Motor Control X. Conference*, July 2015.

[ZKWL13] Rebeka Zsoldos, Björn Krüger, Andreas Weber, and Theresia Licka. Equine neck movement: Comparing segmental kinematics and precission recall of trajectories and velocities. In *Congress of the Int. Society of Biomechanics (ISB)*, 2013.

[ZlTH08] Feng Zhou, F. De la Torre, and Jessica K Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conference on Automatic Face and Gestures Recognition*, September 2008.

[ZlTH13] Feng Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2013.

[ZMI06] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28:1530–1535, Sept 2006.

[ZVK+15] Rebeka Zsoldos, Anna Vögele, Björn Krüger, Ulrike Schröder, and Theresia Licka. Comparison of the activities measured at three electrode locations on equine gluteus medius muscle in walk and trot. In *XXV Congress of the International Society of Biomechanics (ISB 2015)*, July 2015. accepted as oral presentation.