Efficient Unsupervised Temporal Segmentation of Motion Data

Björn Krüger, Anna Vögele, Tobias Willig, Angela Yao,
Reinhard Klein, *Member, IEEE* and Andreas Weber, *Member, IEEE*

Abstract—We introduce a method for automated temporal segmentation of human motion data into distinct actions and compositing motion primitives based on self-similar structures in the motion sequence. We use neighbourhood graphs for the partitioning and the similarity information in the graph is further exploited to cluster the motion primitives into larger entities of semantic significance. The method requires no assumptions about the motion sequences at hand and no user interaction is required for the segmentation or clustering. In addition, we introduce a feature bundling preprocessing technique to make the segmentation more robust to noise, as well as a notion of motion symmetry for more refined primitive detection. We test our method on several sensor modalities, including markered and markerless motion capture as well as on electromyograph and accelerometer recordings. The results highlight our system's capabilities for both segmentation and for analysis of the finer structures of motion data, all in a completely unsupervised manner.

Index Terms—Temporal segmentation, time series clustering, human motion analysis

1 Introduction

Human motion capture, once associated with producing special effects for films and video games, is common today in diverse applications ranging from health care to consumer electronics. 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 reptitions 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 [1], [15], [27], [38], [43], [50], [51], [52] and the computer graphics [2], [17], [25], [34], [45] communities.

We propose a segmentation method which 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.

We pose segmentation as an efficiently solvable graph problem, as first presented in [45] for segmenting motion capture data. To further improve computational efficiency, we employ a Neighborhood Graph [23]. In addition, we propose three new contributions, making the method robust and applicable to motion data from varying sensor modalities. First, we propose a novel feature bundling technique for preprocessing motion features. The feature bundling allows for compact model representations of the motions, as well as robustness to noise, to accommodate modalities such as markerless 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 [29], [51], [52]. However, it has a direct impact on synthesis methods, such as motion graphs [22], [29]; not handling the transitions forces synthesized sequences to include additional primitives coming from possibly unrelated transitions, although this is neither convenient nor intrinsically motivated.

- *B. Krüger is with Gokhale Method Institute, Stanford (CA), United States. E-mail: kruegerb@cs.uni-bonn.de*
- *A. Vögele, T. Willig, A. Yao, R. Klein and A. Weber are with the Department of Computer Science, University of Bonn, Germany.*

Manuscript received April, 2015; revised October, 2015.

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 [51]. 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.

The rest of this paper is organized as follows. First, the feature bundling technique is introduced in Section 3. Details of the motion segmentation into actions and subsequent motion primitives are described in Section 4. In Section 5, we introduce a notion of symmetry to help refine the segmented motion primitives. In Section 6, the clustering of the motion primitives is discussed. We present segmentation results from motion sequences of varying modalities in Section 7, and conclude by discussing the achievements and limitations of our novel method as well as possible extensions in Section 8. Source code for the algorithm is available online upon acceptance of this work.

2 Related Work

Temporal segmentation is related to a number of different fields such as data mining [9], [20], audio and speech processing [31], [32], [36], [37], and behavioural pattern recognition [47]. In statistical terms, the segmentation problem can be posed as a change point detection task [9] which has been extended to a multi-dimensional setting [47] based on the established Bayesian techniques [46]. From a more general signal processing point of view, kernel-based methods for change-point analysis [8], [13], Hidden Markov Models as means for training and recognition [36], and audio thumbnailing combined with enhanced similarity matrices [31], [32], have been introduced. There have been a number of attempts to solve the non-trivial 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 [3], [4], [26]. Beaudoin et al. [3] proposed to extract motion motifs as building blocks of graphs. Gall et al. [26] create temporally meaningful pose clusters associated with unlabeled actions. Bernard et al. [4] 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 pre-computed templates and match them to test sequences. For example, Lv et al. [28] uses a combination of HMMs and AdaBoost to learn discriminative feature templates from labeled segments to perform action recognition and segmentation. Müller et al. [30], [33], [34] developed a framework based on geometric features to learn templates for solving and accelerating solutions to matching problems such as annotation and retrieval [30], [33]. Adaptive segmentation [34] 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 [24], [40] 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 [22] and motion parameterization [21]. 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 [21] has become the standard for finding similar motion clips at interactive speeds.

Later works [14], [29], [39] combine both these ideas to accomplish motion synthesis techniques for high quality interactive applications. Min and Chai's Motion Graphs++ [29] 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 Barbic et al. [2]. This segmentation focuses on detecting behavior segments and is similar in spirit to the first step of our approach for isolating distinct actions. In [2], the quality of local PCA models is tracked temporally; new activities are defined when the old PCA model cannot 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. [51], [52] 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. [11] 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.

3 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 [5], [16], 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 1 for an overview). Vejdemo-Johansson et al. [44] 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 7). 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 [23] 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:

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

Since our data is assumed to be Gaussian, we use a general approximation of the bandwidth matrix which minimizes the MISE (refer to Scott [42], Chapter 6, Scott's rule) by

$$H_{jj} = \sigma_j k^{\frac{-1}{d+4}}, \tag{2}$$

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

$$K(x, \mu, \sigma) = e^{-\frac{1}{2}(x-\mu)' \Sigma^{-1} (x-\mu)}, \tag{3}$$

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 [42]. 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 [2], [12], [18].

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 $\hat{F}_i$:

$$\hat{F}_i = \underset{O_i}{\operatorname{argmin}} K_{H^i}(F + O_i). \tag{4}$$

An example of the feature bundling is given in Figure 1 a). A 3D projection of input and output feature points of a motion sequence are given in Figure 1 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 a) and b) respectively.

4 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.

4.1 Local Neighbors

A motion sequence M is given as a collection of m subsequent data points $p_1, \dots, p_m$, each of which is represented by a feature vector $F = (f_1, \dots, f_N)$ of dimension N capturing the information over time. The features are modality

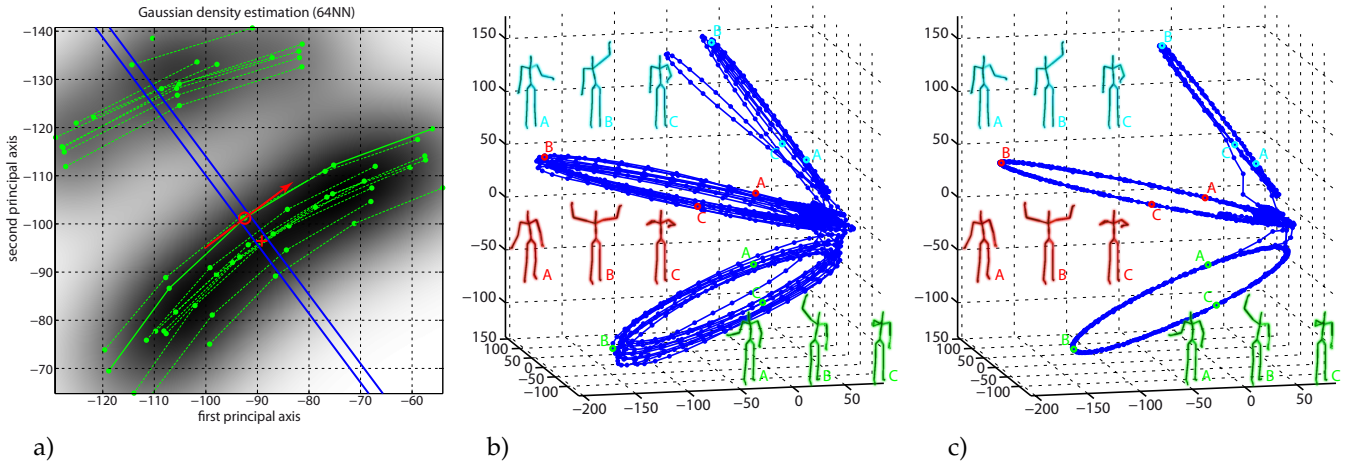


Fig. 1: a) 2D example for **density estimation**: Original data points (red circle), 64 **nearest neighbors** (green dots connected in time), the **direction of movement** (red arrow), the 1D **subspace** used for optimization (blue lines) and the resulting position (red cross). b) 3D **PCA projection** 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. c) **Bundled feature points** for the same motion capture sequence. Figure is best viewed in colour.

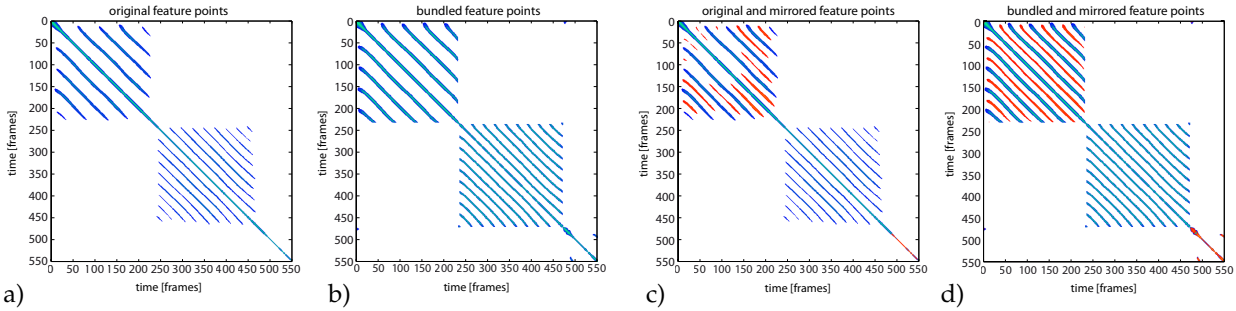


Fig. 2: Comparison of SSSMs of two different situations: original and bundled features. Part a) shows the SSSM of original features as they were extracted from the input motion. Part b) shows the same SSSMs bur of the bundled features. Images c) and d) include the mirrored features also (highlighted 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)).

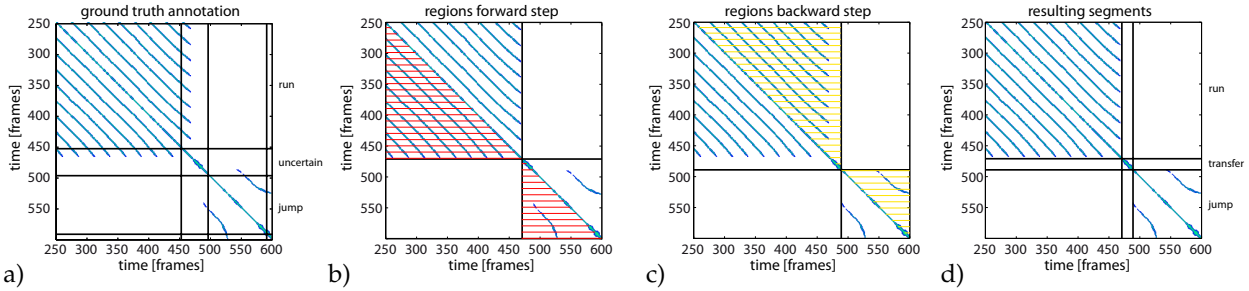


Fig. 3: Region growing for activity separation: a) Sparse self similarity matrix with ground truth annotations; the label 'uncertain' indicates an area of inconclusive user annotations. b) Same matrix without main diagonal; also results for forward step of region growing. c) Results backward step of 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.

specific and stacked in the time dimension, yielding a vector representation $[p_{i-f_1}, p_i, p_{i+f_2}]'$ of features over a window $w = [i - f_1, i, i + f_2]$ 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.

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 [21] to multi-view

action recognition [19]. A SSSM, as shown in Figure 2 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 4 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.

4.2 Segmentation into Distinct Activities

Figure 3 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 nosy 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. 3b) 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. 3c). 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. [45], 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 [45], 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

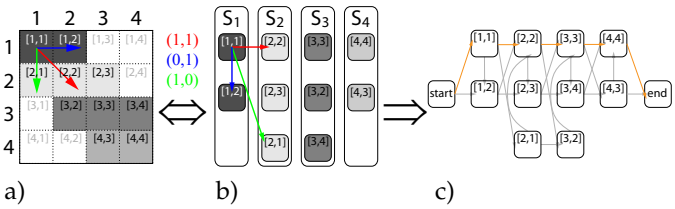


Fig. 4: Toy example illustrating the relationship between SSSM and neighborhood graph G_M. a): SSSM of four consecutive points with allowed steps indicated by arrows. Red arrow: step $(1, 1)$, blue arrow: step $(0, 1)$, green arrow: step $(1, 0)$. b): Neighborhoods of each of the four points, e. g. S_i is the neighborhood of the point i. c): Resulting neighborhood graph.

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.

4.3 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 Figure 5; 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. 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 path. We note that if no reoccuring primitives are found, the action segment is considered itself a single primitive.

Alternative Representation by a Neighborhood Graph

All of the neighbors p_j within one specific activity stored in the set of neighborhoods $S = \{S_i,\ i \in 1, \ldots, n\}$ 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. 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 4 shows a toy example to illustrate a possible scenario.

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 5 for a visualization of connected components).

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 $v_A = (a_1, \ldots, a_\lambda)$ with $a_i \in A$ meeting constraints such as $a_i \leq \nu a_{i+1}$ for all indices and $v_B = (b_1, \ldots, b_\lambda)$ 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. [23] 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.

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 $O(ksn)$.

5 Symmetry of Motion Data

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. 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.

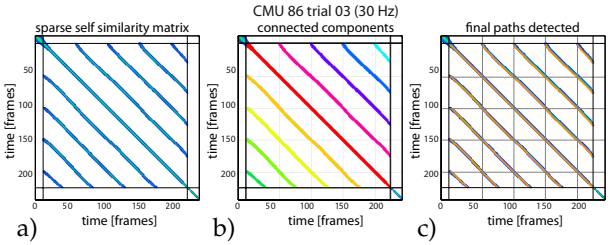


Fig. 5: Illustration of connected components in G_{act}. a) SSSM corresponding to walking. b) Same matrix with its connected components color coded. c) Optimal warping paths highlighted by orange lines.

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. 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.

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.
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.

Naturally, a mixture of two or more situations is possible when an action contains different types of motion primitives (see Appendix B 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. We make use of phase-shifted symmetry in order to distinguish phase-shifted primitives like single steps in walking.

6 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 [29], 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 4.3) 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 4.3) we add an edge between the corresponding nodes in $\mathbf{G}_{\mathbb{M}}$. 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. [45]. 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.

7 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 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.

7.1 Segmenting Markered Motion Capture Data

We apply our segmentation method to the the motion sequences of subject 86 of the CMU database [7], as was done by Zhou et al [51], [52]. We compare our segmentations to theirs, as well as to our previous work [45] in Figure 6. We show a number of improvements in comparison to [2], [51], 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. [51] group all primitives together as the same type (dark grey blocks), while we are able to distinguish between back and forth wiping motions versus circular wiping movements (various shades of grey).

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.

Accuracy Comparison

Our method produces the same action classes as [45] and nearly the same classes as [51], [52]. We use the same

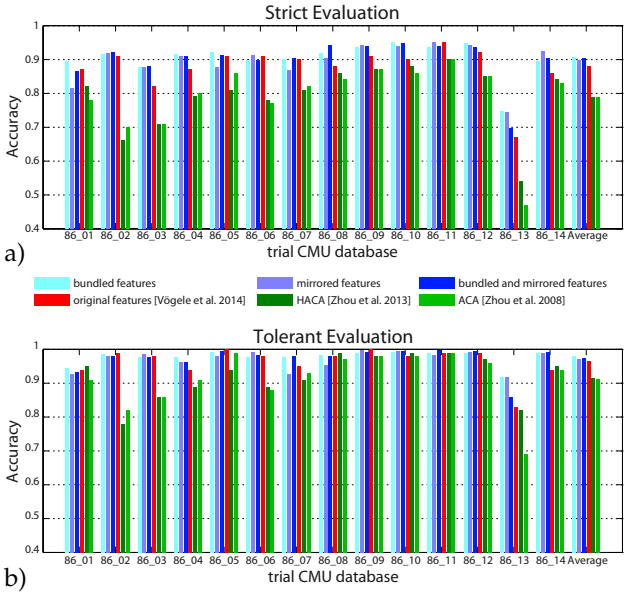


Fig. 7: Accuracy values of different segmentation methods: Light blue bars refer to our method (bundled features only), purple: our method (mirrored features only), blue: (bundled and mirrored features combined), red: Vögele et al. [45], dark green: Zhou et al.(ACA) [52] light green: Zhou et al.(HACA) [51] 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.

methods as [45] 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 7. 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

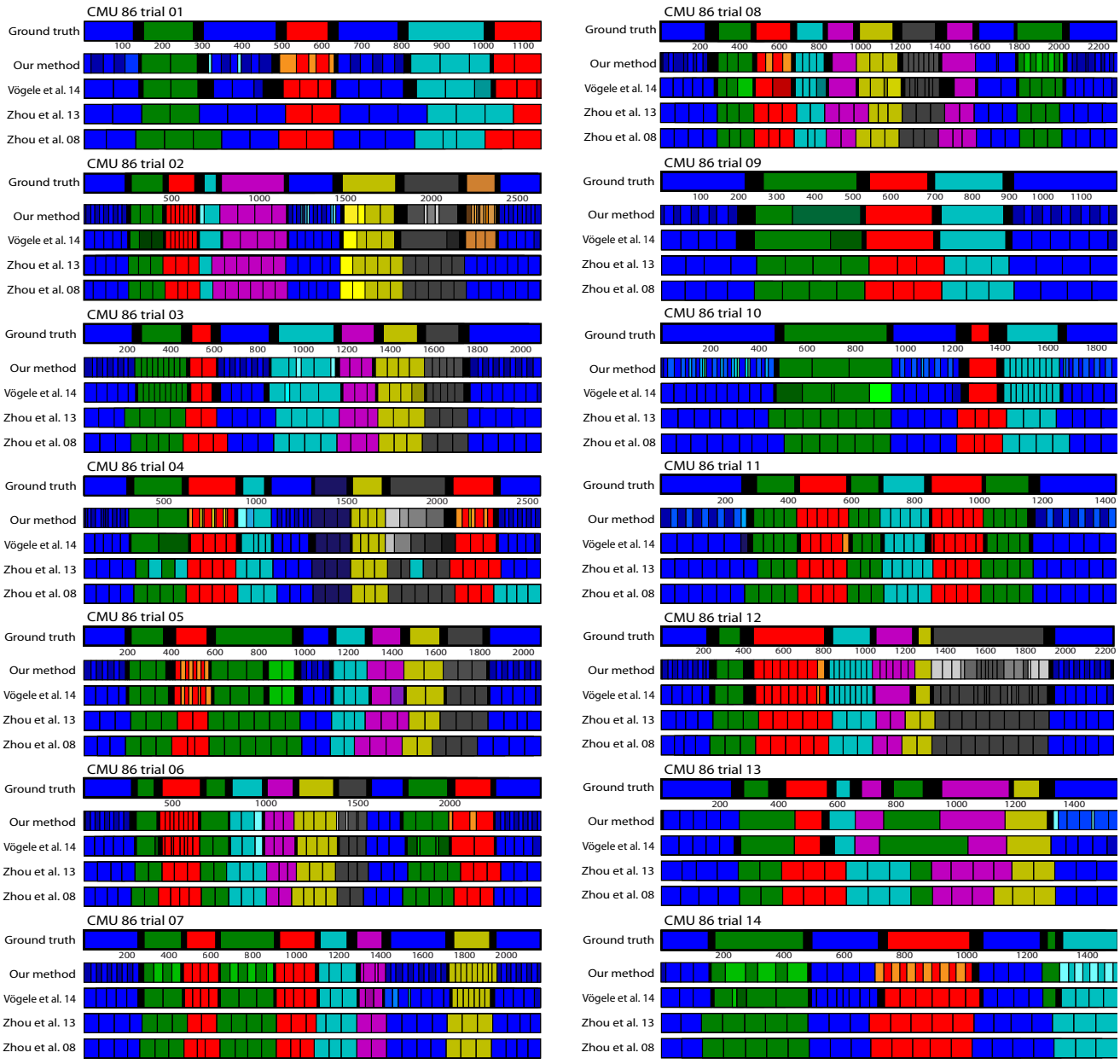


Fig. 6: Segmentation results for CMU 86 trial 01 to 14. For each of the 14 trials, the first row displays the human ground truth annotations, the second row compares them to our results. The results of the (H)ACA methods (Zhou et al. [51], [52]) and Vögele et al. [45] are given in the two lower rows. Note that there is a variety of units of different sizes indicating that the actual length of motion primitives may vary considerably.

contained in the cluster can be defined as:

$$D = \sum_{i=1, j \neq i}^{\|C\|} \left(\frac{\mathrm{DTW}_\alpha(s_i, s_j)}{\|s_i\|} \right) \tag{5}$$

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. [22]. 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. Figure 8 a) compares the clustered results by our method, the method of Vögele et al., the HACA [51] and the ACA [52] method. Figure 8 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 6, Subject 86 trial 03, frames 1600-1800) and we segment this into exactly five primitives. Zhou et al.'s methods [51], [52] does not account for the inherent repetition and segments the action into three primitives, thereby yielding much

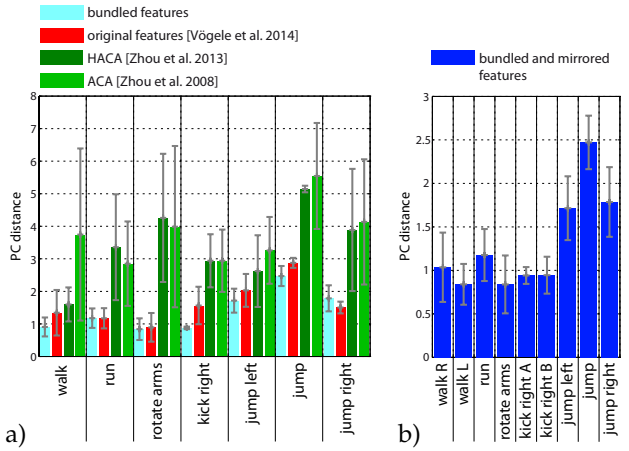


Fig. 8: a) Evaluation of clusters produced by our method, the method of Vögele et al., the HACA and the ACA method. The example at hand is trial 3 of CMU subject 86. The respective means are given by the blue color bars for our method, red for Vögele et al., (dark) light green color bars for (H)ACA, 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.

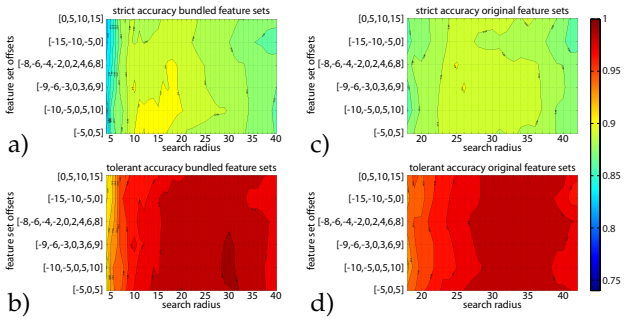


Fig. 9: Maps displaying the accuracy values for various combinations of the parameters: The generalized radius R is plotted on the horizontal axis, choices of stacking offsets are plotted along the vertical axis. A stacking offset of $[-5, 0, 5]$ is a concatenation of frames at times $i-5$, i and $i+5$. 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.

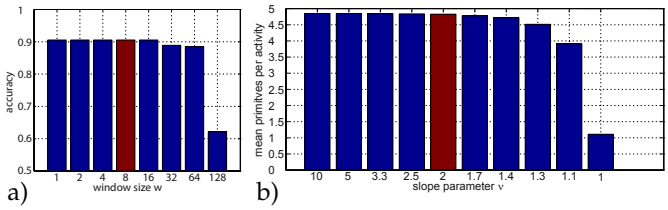


Fig. 10: a) Accuracy (strict evaluation) for varying window sizes w of the region growing step. $w = 8$ (red bar) was chosen for our experiments. b) Average number of motion primitives per activity based on the minimal and maximal slope of the warping paths. $\nu = 2$ (red bar) was chosen for all other experiments in this work.

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 4.2) and the temporal window for feature stacking (Section 4.1). We found that the method is insensitive to either parameter and show the segmentation accuracy in Figure 9 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 10 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 4.3). Following conventions in the literature [21], [23] 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 10 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 11 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 11 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.

7.2 Combined Motion Sensors

To demonstrate our algorithm's effectiveness on different sensor modalities, we recorded a set of electromyography

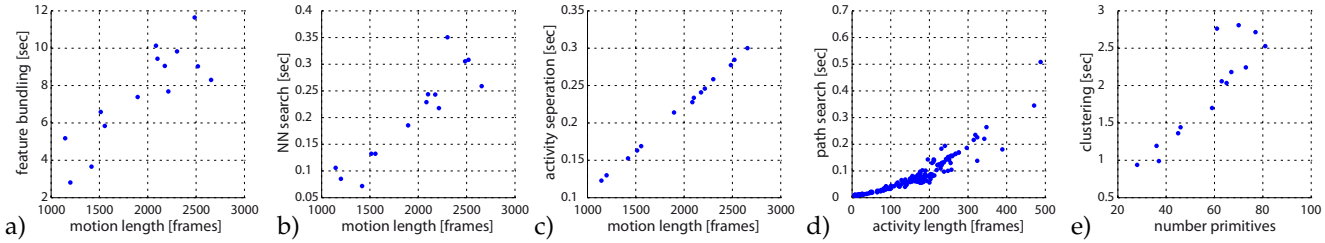


Fig. 11: Scatter Plots for timings of the steps of our method. a) feature bundling, b) knn search, c) region growing (scan lines), d) path searches, e) clustering. Note that the complexity of the first three steps (a-c) is linear in the number of frames, whereas path searches (d) are quadratic in the lengths of the activities. In practice, the clustering (e) also computes linear in the number of motion primitives.

(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 12). 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 A 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 [6] 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 12 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' (grey 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.

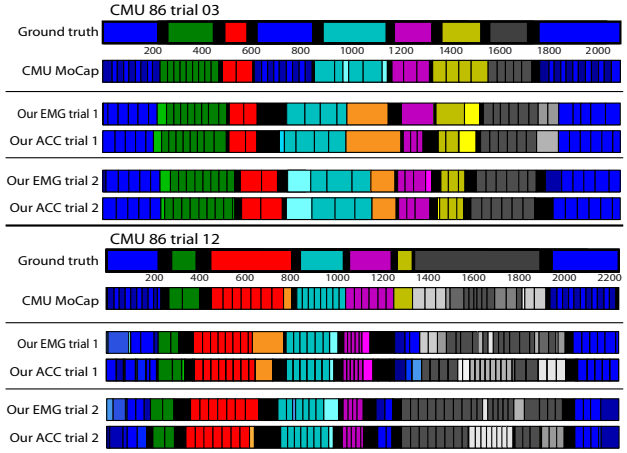


Fig. 12: Exemplary results of two trial As can be seen from the key point visualization, this third example is also an interesting situation for the consistency measure.s of subject 86 of the CMU. The color codes correspond to the different motion clusters the primitives were assigned to by our method. The results are very similar between the data sets, even between the original and the re-enacted recordings.

7.3 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. [10], 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 14.

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 [35] 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 14(a) and (b)). Figure 14(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 14(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 14(a), three repetitions were broken up into two individual parts (yellow and turquoise). In Figure 14(b), the longer breaks (brown) between the repetitions were found but not all repetitions (dark blue) could be cleanly separated. Finally, in Figure 14(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 13. 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 13(a)), with a relatively low standard deviation of 15% around the mean values (see Figure 13(b)) when feature bundling is enabled. Figure 13(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 14 (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.

7.4 Non-cyclic Kinect data

We tested our approach on the more challenging MSR 3D Online Action Dataset [49]. 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 15 (a) along with the corresponding SSSM (b). Since the annotations are given as intervals in this dataset we use

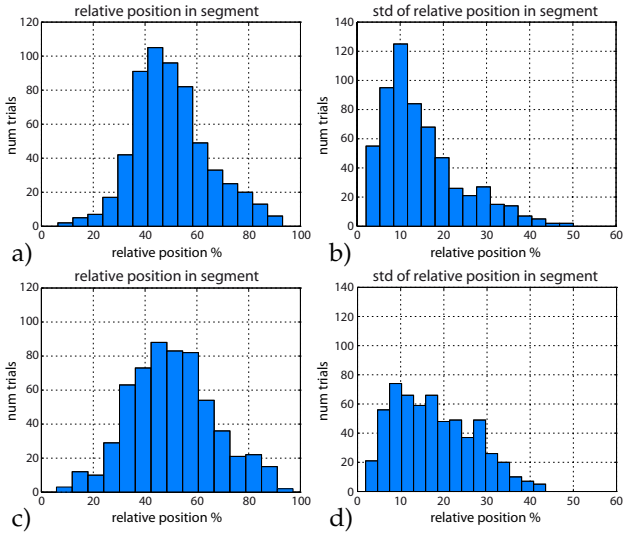


Fig. 13: The histograms show the average location of given key point annotations (left hand side) in the motion primitive and their standard deviations (right hand side). In the majority of cases, the location of key points is approximately at the center of the corresponding primitive with a deviation 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).

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 15 (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.

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

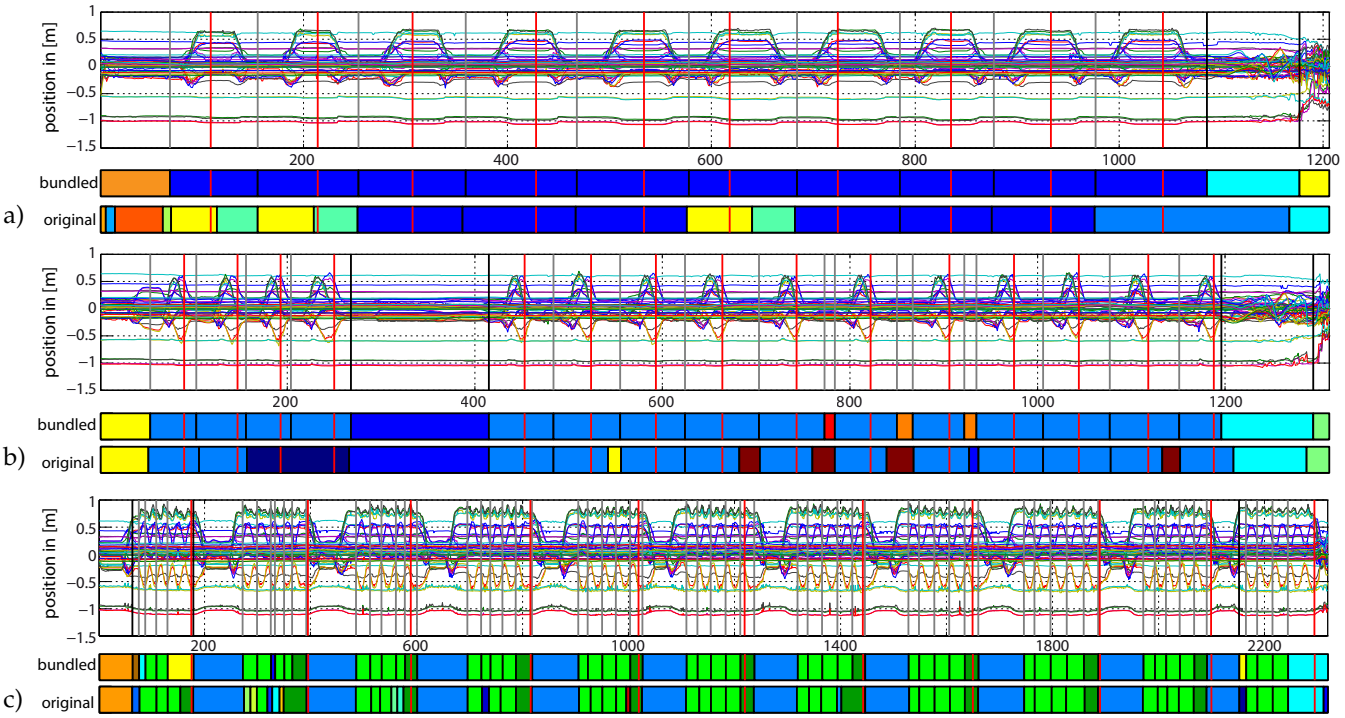


Fig. 14: Three different results produced on Kinect recordings. The input time series are plotted for an overview of the general structures of the underlying motions, showing motions with (a) regular primitives segments [P1_2_g09_s08] (b) a sequence with a longer break and also some speed variation [P1_2_g05_s08] 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.

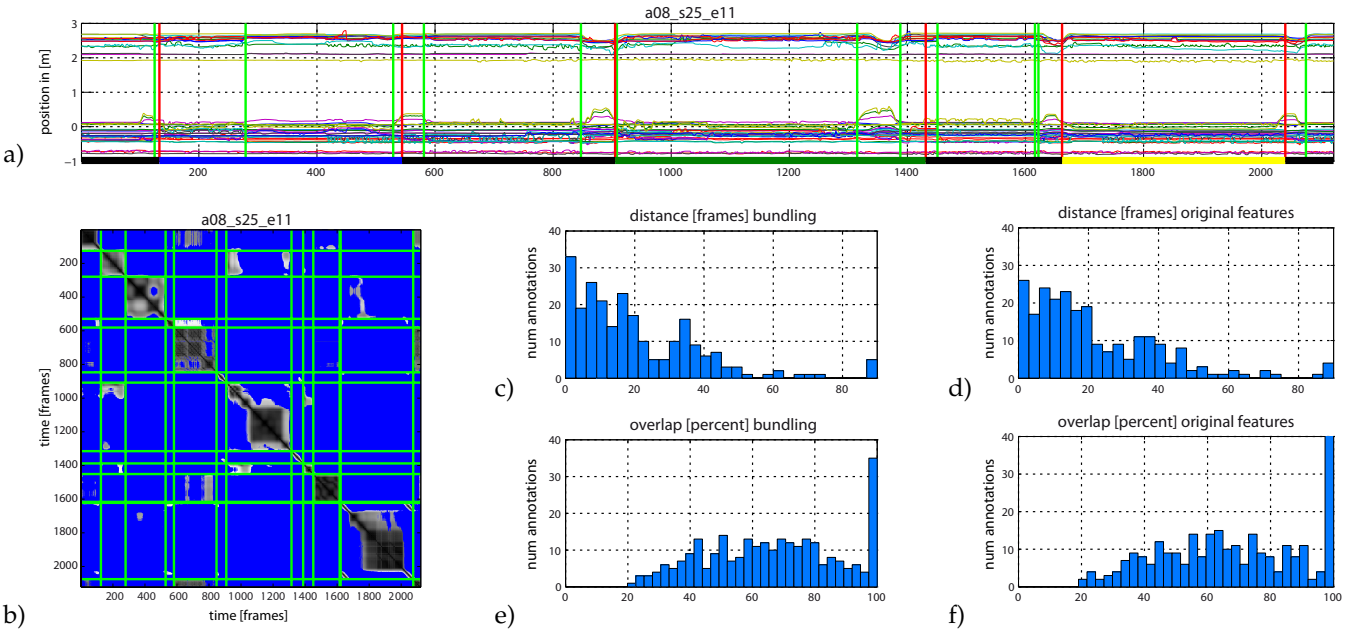


Fig. 15: Results produced on Kinect recordings from the MSR 3D Action Dataset. Segmentation results (green lines) and ground truth annotations (red lines) for trial a08_s25_e11 (a). The bars at the bottom indicate the ground truth labels: no action (black), drinking (blue), eating (green), reading book (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).

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 [41].

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 [7]. For other sensor modalities we found a similar behaviour; 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 [48] 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.

We offer source code for our feature bundling, segmentation and clustering approach online[1]. We believe that it will be of use for many different research applications and hope that it will encourage others in the community to work on generalized segmentation.

REFERENCES

[1] R. Araujo and M. Kamel, "Semi-supervised kernel-based temporal clustering," in Machine Learning and Applications (ICMLA), 2014 13th International Conference on, Dec 2014, pp. 123–128.

[2] J. Barbič, , J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. Pollard, "Segmenting motion capture data into distinct behaviors," in In Proc. of Graphics Interface 2004, May 2004, pp. 185 – 194.

[3] P. Beaudoin, S. Coros, M. van de Panne, and P. Poulin, "Motion-motif graphs," in Proc. of the 2008 ACM SIGGRAPH, ser. SCA '08, 2008, pp. 117–126.

[4] J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer, "Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation," IEEE TVCG (Proc. VAST), 2013.

[5] J. Bottger, A. Schafer, G. Lohmann, A. Villringer, and D. S. Margulies, "Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain," IEEE TVCG, vol. 20, no. 3, pp. 471–480, 2014.

[6] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," Sensors, vol. 13, no. 9, pp. 12 431–12 466, 2013.

[7] CMU, "Carnegie Mellon University Graphics Lab: Motion Capture Database," 2013. [Online]. Available: http://mocap.cs.cmu.edu

[8] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," Signal Processing, IEEE Transactions on, vol. 53, no. 8, pp. 2961–2974, Aug 2005.

[9] P. Fearnhead, "Exact and efficient bayesian inference for multiple changepoint problems," Statistics and Computing, vol. 16, no. 2, pp. 203–213, 2006.

[10] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in CHI. ACM, 2012, pp. 1737–1746.

1. http://cg.cs.uni-bonn.de/en/projects/gemmquad/motionsegmentation/

[11] D. Gong, G. Medioni, and X. Zhao, "Structured time series analysis for human action segmentation and recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 36, no. 7, pp. 1414–1427, July 2014.

[12] G. Guerra-Filho and Y. Aloimonos, "A language for human action," Computer, vol. 40, no. 5, pp. 42–51, May 2007.

[13] Z. Harchaoui, E. Moulines, and F. R. Bach, "Kernel change-point analysis," in Advances in Neural Information Processing Systems 21. Curran Associates, Inc., 2009, pp. 609–616.

[14] R. Heck and M. Gleicher, "Parametric motion graphs," in Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games, ser. I3D '07. New York, NY, USA: ACM, 2007, pp. 129–136.

[15] M. Hoai, Z.-Z. 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, June 2011, pp. 3265–3272.

[16] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," IEEE TVCG, vol. 12, no. 5, pp. 741–748, Sept 2006.

[17] J. Hou, L.-P. Chau, Y. He, and N. Magnenat-Thalmann, "Human motion capture data tailored transform coding," IEEE TVCG, 2015.

[18] O. C. Jenkins and M. J. Matarić, "A spatio-temporal extension to isomap nonlinear dimension reduction," in Proceedings of the Twenty-first International Conference on Machine Learning, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 56–.

[19] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 1, pp. 172–185, Jan 2011.

[20] 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, 2001, pp. 289–296.

[21] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," ACM Trans. Graph., vol. 23, no. 3, pp. 559–568, Aug. 2004.

[22] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," ACM Trans. Graph., vol. 21, no. 3, pp. 473–482, Jul. 2002.

[23] B. Krüger, J. Tautges, A. Weber, and A. Zinke, "Fast local and global similarity searches in large motion capture databases," ser. SCA 2010, 2010, pp. 1–10.

[24] R. Lan and H. Sun, "Automated human motion segmentation via motion regularities," The Vis. Comp., pp. 1–19, 2013.

[25] G. Liu and L. McMillan, "Segment-based human motion compression," in Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ser. SCA '06, 2006, pp. 127–135.

[26] A. López-Mendez, J. Gall, J. R. Casas, and L. J. V. Gool, "Metric learning from poses for temporal clustering of human motion," in BMVC, 2012, pp. 1–12.

[27] C. Lu and N. J. Ferrier, "Repetitive motion analysis: segmentation and event classification," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 2, pp. 258–263, Feb 2004.

[28] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in Computer Vision ECCV 2006, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 3954, pp. 359–372.

[29] J. Min and J. Chai, "Motion graphs++: A compact generative model for semantic motion analysis and synthesis," ACM Trans. Graph., vol. 31, no. 6, pp. 153:1–153:12, Nov. 2012.

[30] M. Müller, A. Baak, and H.-P. Seidel, "Efficient and robust annotation of motion capture data," ser. SCA 2009, 2009, pp. 17–26.

[31] M. Müller and P. Grosche, "Automated segmentation of folk song field recordings," in Proceedings of the ITG Conference on Speech Communication, Braunschweig, Germany, 2012.

[32] M. Müller, P. Grosche, and F. Wiering, "Robust segmentation and annotation of folk song recordings," in Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR), Kobe, Japan, Oct. 2009, pp. 735–740.

[33] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," ser. SCA 2006, 2006.

[34] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," ACM Trans. Graph., vol. 24, no. 3, pp. 677–685, Jul. 2005.

[35] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," 7 J J Thomson Ave, CB30FB Cambridge, UK, Tech. Rep. MSR-TR-2012-68, July 2012.
[36] M. Ostendorf, V. Digalakis, and O. Kimball, "From hmm's to segment models: a unified view of stochastic modeling for speech recognition," Speech and Audio Processing, IEEE Transactions on, vol. 4, no. 5, pp. 360–378, Sep 1996.
[37] T. Prätzlich and M. Müller, "Frame-level audio segmentation for abridged musical works," in Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 2014.
[38] Y. Rui and P. Anandan, "Segmenting visual actions based on spatio-temporal motion patterns," in Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 1, 2000, pp. 111–118 vol.1.
[39] A. Safonova and J. K. Hodgins, "Construction and optimal search of interpolated motion graphs," ACM Trans. Graph., vol. 26, no. 3, Jul. 2007.
[40] S. Salamah, L. Zhang, and G. Brunnett, "Hierarchical method for segmentation by classification of motion capture data," in Virtual Realities, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, vol. 8844, pp. 169–186.
[41] M. Sattler, R. Sarlette, and R. Klein, "Simple and efficient compression of animation sequences," in Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005), Jul. 2005.
[42] D. W. Scott, Multivariate density estimation : theory, practice, and visualization, ser. Wiley series in probability and mathematical statistics : Applied probability and statistics section. New York, Chichester, Brisbane: Wiley-Interscience, 1992.
[43] D. D. Vecchio, R. M. Murray, and P. Perona, "Decomposition of human motion into dynamics-based primitives with application to drawing tasks," Automatica, vol. 39, no. 12, pp. 2085 – 2098, 2003.
[44] M. Vejdemo-Johansson, F. T. Pokorny, P. Skraba, and D. Kragic, "Cohomological learning of periodic motions," 2014.
[45] A. Vögele, B. Krüger, and R. Klein, "Efficient unsupervised temporal segmentation of human motion," in 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Jul. 2014.
[46] A. S. Willsky, E. B. Sudderth, M. I. Jordan, and E. B. Fox, "Sharing features among dynamical systems with beta processes," in Advances in Neural Information Processing Systems 22. Curran Associates, Inc., 2009, pp. 549–557.
[47] X. Xuan and K. Murphy, "Modeling changing dependency structure in multivariate time series," in Proceedings of the 24th International Conference on Machine Learning, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 1055–1062.
[48] A. Yao, J. Gall, L. V. Gool, and R. Urtasun, "Learning probabilistic non-linear latent variable models for tracking complex activities," in Advances in Neural Information Processing Systems 24. Curran Associates, Inc., 2011, pp. 1359–1367.
[49] G. Yu, Z. Liu, and J. Yuan, "Discriminative orderlet mining for real-time recognition of human-object interaction," in Computer Vision – ACCV 2014, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, vol. 9007, pp. 50–65.
[50] L. Zelnik-Manor and M. Irani, "Statistical analysis of dynamic actions," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28, no. 9, pp. 1530–1535, Sept 2006.
[51] F. Zhou, F. D. la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 35, no. 3, pp. 582–596, 2013.
[52] ——, "Aligned cluster analysis for temporal segmentation of human motion," in IEEE Conference on Automatic Face and Gestures Recognition, September 2008.

Björn Krüger studied computer science, mathematics and physics at Bonn university. He received his MS in computer science (Dipl.-Inform.) in 2006 and his PhD (Dr. rer. nat.) in computer science in 2012. From 2012 to 2015 he worked as postdoc at Bonn university. Since 2015 he joined the Gokhale Method Institute (Stanford, CA) as senior researcher. His research interests include: computer animation, computer graphics, machine learning, and motion capture.

Anna Vögele is a research assistent with the department of computer science at the University of Bonn. She received her MS in Mathematics (Dipl.-Math.) from the University of Bonn in 2011. Her research interests include computer graphics and animation as well as machine learning.

Tobias Willig received a BASc degree in Computer Science from Bonn-Rhein-Sieg University of Applied Sciences in 2011 with a grade of 1.3, and a MSc degree in Computer Science from University of Bonn with an excellent grade of 1.2. He finished his Master's degree in 2015 with a thesis on temporal segmentation of human motion capture data.

Angela Yao is currently an Assistant Professor at the Institute of Computer Science at the University of Bonn. She received a BASc degree in Engineering Science from the University of Toronto in 2006 and a PhD in Information Technology and Electrical Engineering from ETH Zurich in 2012. Her research interests are in computer vision and machine learning, with special focus on human pose estimation and action recognition.

Reinhard Klein studied Mathematics and Physics at the University of Tübingen, Germany, from where he received his MS in Mathematics (Dipl.-Math.) in 1989 and his PhD in computer science in 1995. In 1999 he received an appointment as lecturer ("Habilitation") in computer science also from the University of Tübingen, with a thesis in computer graphics. In September 1999 he became an Associate Professor at the University of Darmstadt, Germany and head of the research group Animation and Image Communication at the Fraunhofer Institute for Computer Graphics. Since October 2000 he is professor at the University of Bonn and director of the Institute of Computer Science II.

Andreas Weber studied mathematics and computer science at the Universities of Tübingen, Germany and Boulder, Colorado, U.S.A. From the University of Tübingen he received his MS in Mathematics (Dipl.-Math) in 1990 and his PhD (Dr. rer. nat.) in computer science in 1993. From 1995 to 1997 he was working with a scholarship from Deutsche Forschungsgemeinschaft as a postdoctoral fellow at the Computer Science Department of Cornell University. From 1997 to 1999 he was a member of the Symbolic Computation Group at the University of Tübingen, Germany. From 1999 to 2001 he was a member of the research group Animation and Image Communication at the Fraunhofer Institut for Computer Graphics.

Appendix A
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 16 for a list of locations and a visualization of the hardware attachment). Both sensor attachment and recordings were supervised by an accredited expert.

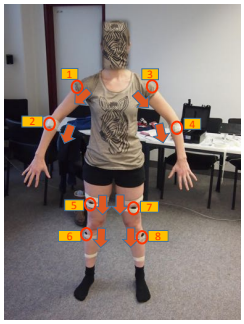
A selection of trials performed originally by subject 86 of the CMU database was re-enacted by another human subject as true to original as possible. This was done to compare the performance of our segmentation method on other data modalities while maintaining control of the activities represented by 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.

Appendix B
Symmetry Types

Schematic representations of self-similarity matrices for each of the possible cases, discussed in Section 5 are given in Table 1. 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.

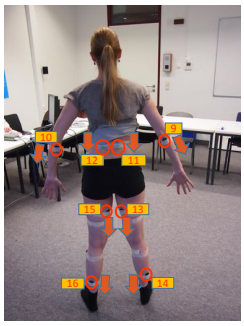
Acknowledgments

The authors would like to thank Kristian Welle (Bonn University Hospital, Clinic for Orthopaedics and Trauma Surgery) for supporting with the EMG measurements. This work was partially supported by Deutsche Forschungsgemeinschaft (DFG) under research grants KR 4309/2-1.



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)

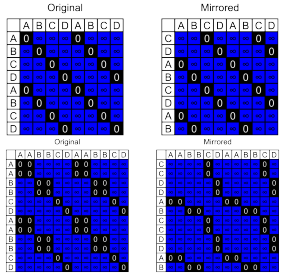
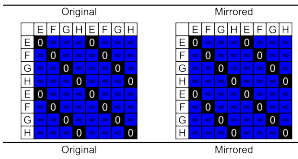
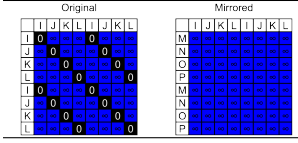
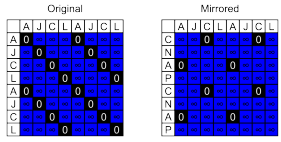


EMG - Back

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)

Fig. 16: Documentation of EMG sensor placement on a human subject seen from the front and back, respectively. Orange arrows point to direction of y-axis.

TABLE 1: Schematic self similarity matrices for various types of symmetries.

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