Automatic Temporal Segmentation of Articulated Hand Motion

Katharina Stollenwerk[1], Anna Vögele[2], Björn Krüger[3], André Hinkenjann[1], and Reinhard Klein[2]

[1] Bonn-Rhein-Sieg University of Applied Sciences, Institute of Visual Computing, Sankt Augustin, Germany
[2] Insitute for Computer Science II, Bonn University, Germany,
[3] Gokhale Method Institute, Stanford CA, United States

Abstract. This paper introduces a novel and efficient segmentation method designed for articulated hand motion. The method is based on a graph representation of temporal structures in human hand-object interaction. Along with the method for temporal segmentation we provide an extensive new database of hand motions. The experiments performed on this dataset show that our method is capable of a fully automatic hand motion segmentation which largely coincides with human user annotations.

1 Introduction

Motion Capture has become a standard technique for motion data recording in the past decades. Easy access to improved and relatively low-cost systems makes motion capture possible to a wider community and for numerous applications.

There is an increased focus on recording facial movement and hand gestures, since both are significant parts of communication and daily life. Recent work ([1, 2]) has brought to attention the importance of correctly-timed hand motions and details in face and hand movement.

There is a need for high quality data in order to enable both motion analysis and synthesis. Recent technologies allow for fast capturing of highly resolved motion data [3–5]. However, in order to make use of the recordings, appropriate tools for data processing are needed. The segmentation of motions into simple data units is a crucial step in processing [6]. Recent developments of segmentation methods produce good results for full-body motion [6, 7]. Our goal is presenting techniques which work for gestures and grasping and are similarly efficient as the above-mentioned. We introduce a method for temporal segmentation of hand motions which enables the isolation of primitive data units. We show that these units coincide with perceptive motor primitives by comparing the results to those which have been manually segmented by different users. Moreover, we discuss a method for clustering the motion segments achieved by the segmentation, thus resulting in a compact representation of hand motion.

The main contributions of this paper are:

– A database of hand motions
– A technique for fully automatic and accurate segmentation
– A method to cluster segments

The remainder of this paper is organised as follows. Section 2 gives an overview of the work related to temporal segmentation and processing of hand motion. Our recording setup and the database that our experiments are based on are described in Section 3. Section 4 introduces the segmentation technique. Section 5 discusses our clustering approach. Finally, an evaluation as well as a comparison to other segmentation approaches are presented in Section 6.

2 Related Work

Hand Motion Capturing Possibilities for capturing finger motion data include marker-based optical and image-based video tracking with or without depth information as well as glove-based systems with or without tactile sensors. An overview of the main approaches also covering advantages and drawbacks of the respective techniques was surveyed by Wheatland et al. [8].

Current approaches combine multiple capturing methods to overcome limitations of individual techniques. Zhao et al. [3] describe how to record high-fidelity 3D hand articulation data with a combination of an optical marker-based motion capture system and a kinect camera. Arkenbout et al. [9] integrate a 5DT Data Glove into the kinect-based Nimble VR system using Kalman filter. This resolved visual self-occlusion of the hand and fingers and improved precision and accuracy of the joint angles' estimates. Ju and Liu [10] capture joint angle data, finger and hand (contact) force data and sEMG data of forearm muscles in order to study correlations of different sensory information.

This variety of sensor fusion for capturing hand motions indicates that the acquisition of high-quality hand motion data has not yet been satisfactorily resolved. All of the above mentioned approaches point out that they are increasing the quality of the recorded data. Nevertheless, we have decided to only use a CyberGlove data glove as, on the one hand, it was not important that the user's hand is unencumbered and on the other hand we believe that the main challenge in recording hand motion and manipulation data is occlusion from an object or the hand itself, both of which are handled effortlessly by a data glove.

Databases While there are a number of high-quality, full-body motion capture databases that can be used for academic purposes (e.g. the CMU and HDM05 motion capture databases [11, 12]), only a few such data collections exist for articulated hand motions.

In the field of robotics, Goldfeder et al. [13] presented algorithms for automatic generation of a database of precomputed stable grasps, i.e. a single pose, for robotic grasping. This resulted in *The Columbia grasp database* which contains computed grasp configurations of different (robotic) hands along with a set of graspable objects. Thus, the database only contains single-hand poses and no finger movements. Feix et al. [14] provide a small dataset of human grasping

used as a basis for evaluation of the motion capabilities of artificial hands. The dataset contains 31 motions, each performed by five subjects twice. The motion data contains only the 3D position and orientation of each fingertip with no information on the specific underlying hand model. Notable also due to its size, the *NinaPro database* [15] contains 52 full-hand and wrist motions collected from 27 subjects. The data recorded consists of surface electromyography (sEMG) data together with the 8-bit valued raw output of a 22-sensor CyberGlove (kinematic data). The 8-bit valued raw kinematic recordings of each sensor only roughly represent joint angles and fail to account for cross coupled sensors in the CyberGlove.

Segmentation and Re-Use of Motion Capture Data Manual segmentation and annotation of motion data into meaningful phases is a tedious and daunting task. But it is segmentation and annotation that makes the data re-usable.

For full-body motion data, unsupervised, temporal segmentation techniques have been developed. Among them is the work of Beaudoin et al.'s [16] who focus on visualising the structure of motion datasets. They propose to partition motion data streams into motion-motifs organised in a graph structure useful for motion blending and motion data compression. Zhou et al. [17, 7] segment human motion data based on (hierarchical) aligned cluster analysis (H(ACA)). They frame the task of motion segmentation as temporal clustering of (motion capture) data into classes of semantically-similar motion primitives. Min and Chai's Motion Graph++ [18] is not only capable of segmenting motions into basic motion units, but also of motion recognition and synthesis. Their segmentation automatically extracts keyframes of contact transitions and semi-automatically extracts keyframes exhibiting salient visual content changes. Vögele et al. [6] employ (backward and forward) region growing in order to identify start and end frames of activities (groups of motion primitives). These activities are split into motion primitives by taking advantage of how repetitive activity patterns manifest in self-similarity matrices (SSSM). Recently, Krüger et al. [19] further improved the outcomes of [6] by aligning and grouping segregated feature trajectories and exploiting the symmetric nature of motion data.

For temporal segmentation of captured hand motion data, we have mainly found motion streams of conversational hand gestures to have been automatically segmented into different phases and synthesised into new motions, often based on an accompanying audio stream. Examples thereof include Levine et al. [20], Jörg et al. [2] and Mousas et al. [21] who solely use features derived from the wrist joint's position over time for segmentation into gesture phases. While the first two works synthesise new gestures based on the whole hand at once, the authors of the last paper estimate finger motion separately for each finger. The estimation is limited to adjusting frame times for creating optimally timed transitions.

There is little published on temporal segmentation of hand motion data; no research has been found that analysed automatic or unsupervised segmentation of such data. For robot programming and teaching from example in, e.g. pick and place scenarios, researchers have looked into temporal segmentation of recorded human hand and finger motions. This is often needed to characterise

grasp phases (e.g. pre-grasp, grasp, manipulation). However, usually this information is included only implicitly in models used for grasp classification: Ekvall and Kragic [22], for instance, classify grasp movements using 5-state HMMs for grouping fingertip positions and transitions. A noteworthy exception is the work of Kang and Ikeuchi [23]. They segment prehensile movements of a human demonstrator using motion profiles and volume sweep rates. Their presented results, however, are limited to two segmented exemplary motion sequences without specification of reference data or ground truth data. In the area of computer animation, Zhao et al. [24] combined recorded 3D hand motion capture data of ten different grip modes and physics-based simulation with the aim of achieving physically-plausible interaction between a hand and a grasped object. Each motion was then manually segmented into three phases: reaching, closing and manipulation, the last of which is assumed to be a static pose. From their paper, it is unclear whether the grasping motion data was recorded from interacting with a real or a virtual object.

3 Database

To the best of our knowledge there still is no database of articulated human hand motions covering a wide variety of actions and actors usable for motion analysis and data-driven synthesis. We therefore decided to create one.

We chose to include two main setups in the database: *Uncontrolled transport* and *controlled transport*, both of which will be described below. To ensure inter-person consistency in the recording setups and for later reproducibility, a protocol was written detailing each step of the recording. The setups can be summarised as follows:

In *uncontrolled transport*, each person is presented with a variety of objects in random order with no object being presented twice in a row. The task is to pick up the object, move it to a different location, and put it down. The setup was designed in order to obtain a high diversity of possible grasps per object. Each object had to be moved five times. In *controlled transport* each person is again presented with a variety of objects. The task is to pick up the object, move it to a different location, and put it down. Only here, a picture illustrates how to hold the object during transport and the task has to be executed five consecutive times on each object. Contrary to the first setup, the focus here lies on reproducing consistent, predefined grasp motions.

In all setups the hand was placed flat on a table before and after task execution. The data were recorded at an acquisition rate of 60[Hz] using an 18-sensor right-hand ImmersionSquare CyberGlove as depicted in Figure 1. Each hand pose (a set of 18 data points per sample) is represented by a joint angle configuration of the joints in each finger, wrist and palm.

Choice of Grasp Types in Controlled Transport Several fields of research (e.g. biomechanics, robotics, medicine) have introduced grasp taxonomies for grouping different grasps by common criteria seeking simplification of the hand's complex prehensile capabilities. Due to the wide range of application and a resulting

(a) Objects in the database (b) CyberGlove (c) Grasping a tennis ball

Fig. 1. (a) Objects used for grasping: classic notebook, oval jar, tennis ball, mug, cube, bottle crate, small cylinder, pen, bottle, glass, business card, bowl, cylinder large, carton ($7.6 \times 26 \times 37.5[\text{cm}^3]$). Objects in red were used in *controlled transport*.

lack of consensus in naming and classifying grasp types Feix et al. [25, 26] collected and consolidated the vast amount of grasp examples found in literature. Their taxonomy comprises 33 grasp types grouped into 17 basic types by merging equivalent grasps. The chosen taxonomy defines a grasp as *a static configuration of one hand able to securely hold an object* (in that hand). This explicitly rules out intrinsic movements in the hand during grasp, bi-manual interaction and gravity-dependent grasps such as an object lying in equilibrium on a flat hand.

Out of the list of 17 basic grasp types we have chosen to cover 13, omitting speciality configurations such as holding chopsticks (tripod variation), a pair of scissors (distal type), a cigarette between index finger and middle finger (abduction grip), and holding a lid of a bottle after unscrewing it (lateral tripod). Some of the basic grasp types were covered more often, hence extending the number of grasp types to 25 also using different sized objects. For a complete list of grasps and objects used for controlled transport see Table 1 and Figure 1.

Table 1. List of grasps in controlled transport grouped by basic grasp type. It includes the grasp type's number and name based on Feix et al. [25] and the objects grasped.

Nº Name	Object grasped
1 large diameter	bottle (8cm diameter)
2 small diameter	cylinder 25mm diameter
3 medium wrap	bottle crate
10 power disk	lid of jar (7cm diameter)
11 power sphere	tennis ball (64mm diameter)
31 ring	glass (6cm diameter)
28 sphere 3 finger	tennis ball (64mm diameter)
18 extension type	classic notebook (15mm thick)
26 sphere 4 finger	tennis ball (64mm diameter)
9 palmar pinch	business card
33 inferior pincer	tennis ball (64mm diameter)
8 prismatic 2 finger	cylinder 10mm diameter
14 tripod	bottle at cap (3cm diameter)

Nº Name	Object grasped
7 prismatic 3 finger	cylinder 10mm diameter
27 quadpod	tennis ball (64mm diameter)
6 prismatic 4 finger	cylinder 10mm diameter
13 precision sphere	tennis ball (64mm diameter)
20 writing tripod	pen (2cm diameter)
17 index finger extension	cylinder 25mm diameter
4 abducted thumb	cylinder 25mm diameter
15 fixed hook	bottle crate
30 palmar	classic notebook (15mm thick)
16 lateral	business card
32 ventral	cylinder 25mm diameter
22 parallel extension	classic notebook (15mm thick)

The final database contains approximately 2000 grasp motions of ten different persons interacting with 15 different objects represented by 25 grasp types (13 basic grasp types). Each motion is annotated with the object that was grasped and – for controlled transport – the grasp type that was used.

4 Segmentation Approach

We present a novel method for temporal segmentation of articulated hand motion. Since our method is related to techniques in image processing on self-similarity adjacency matrices, it belongs to the same category of methods as the technique introduced in [6] on segmentation of full-body motion. However, there is a specific focus on the demands of hand motion segmentation.

In the following subsections, the process is outlined by discussing pre-processing and feature computation, segmentation, and merging.

4.1 Pre-processing

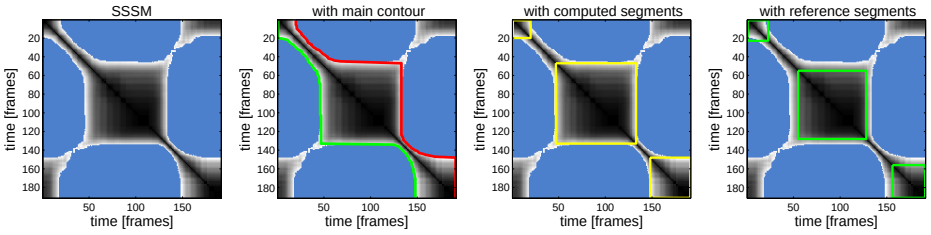
A motion consists of a sequence of n frames each containing a hand pose p_i, $i = 1, \ldots, n$ which itself is encoded in a feature vector $f_i = (f_{i_j})_{j=1..N}$, of dimension N. The feature vector used for our segmentation method consists of the frame-based positions in $\mathbb{R}^3$ inferred from the outermost recorded joint in each finger and thumb (i.e. the thumb's tip and other fingers' distal interphalangeal joint) with respect to the position of the wrist joint. These positions are derived from the recorded angle data by defining a schematic hand model as depicted in Figure 4. For each frame the recorded joint angles are mapped onto the model hand yielding 3D positions for the joints.

In order to convey temporal information in a single feature vector and to emphasise motion consistency over a certain period of time, features are stacked in the time domain. This leads to a vector $[f_{i-t_1}, f_i, f_{i+t_2}]$ of features where f_{i-t_1} is temporally located t_1 frames before f_i and f_{i+t_2} is t_2 frames after f_i.

4.2 Segmentation

The segmentation process can be broken down into two main stages: *construction of the local neighbourhood* and *identification of primitive data units*.

The *local neighbourhood* of a pose p_i is the set S_i of nearest neighbours of p_i. We construct the local neighbourhood based on the Euclidean distance d_{ij} between pairs of feature vectors (f_i, f_j) of hand poses p_i and p_j and a predefined *search radius* r. The *search radius* is a constant R depending solely on the dimensionality N of the feature vector, $r = R \cdot \sqrt{N}$. A pose p_j is added to the set of neighbours of p_i if d_{ij} is below r. This can be efficiently achieved by building a kd-tree from all feature vectors and reporting the subset of vectors located within the *search radius* r for each feature vector f_i (representing pose p_i). As a result we obtain a set S_i of nearest neighbours for each pose p_i.



(a) SSSM (b) Main contour (c) Computed segments (d) Reference segments

Fig. 2. Overview of results of the steps in our segmentation method. The main contour in (b) is divided into its upper (red) and lower (green) part.

These sets are subsequently converted into a sparse self-similarity matrix (SSSM). In our case, this matrix holds the pairwise distances d_{ij} for all pairs of poses p_i, p_j of all sets S_i (see Figure 2). The SSSM can thus be divided into populated regions representing pairs of poses, that are within the *search radius* (greyscale regions in Figure 2), and empty regions in which the pairwise pose distance is outside the *search radius* (blue regions in Figure 2).

Hands, during the grasp phase, do not exhibit major intrinsic movements. This is reflected by a large set of local neighbours and (were it ideal data) expressed in the SSSM by a square region along its main diagonal. We exploit this fact for *identification of primitive data units* and search for square-like shapes along the main diagonal of the matrix. To this end, we first extract start and end indices, t_{s_i} and t_{e_i} $i = 1, \ldots, n$, of populated regions along the diagonal of the SSSM in a row-wise fashion. For each new index t_j we ensure that the sequence found up to t_j is monotonically increasing, i.e. $t_j \geq t_{j-1}\ \forall\ j \leq i$. That way the two index sequences each form a path contouring the upper and lower (list of end and start indices) populated region along the diagonal of the SSSM (Figure 2 (b)). This contour does not contain any neighbour outside of the defined *search radius*.

In the following step, the sequence of end indices is inspected for significant increases, noting the end index t_{e_i} and the row i in which this increase occurrs as interval boundaries $[i, t_{e_i}]$. An increase is considered significant if it is larger than a fixed parameter B, the *ignoreband*. The list of start indices is processed similarly, traversing it from back to front and seeking significant decreases. Eventually, this will result in two preliminary lists of intervals marking candidate primitive data units (rests and grasps) in the motion.

4.3 Merge Step

As we have posed only few constraints on finding square-like shaped segments in the SSSM there is sometimes a significant overlap of the preliminary segments which were identified. While minor overlaps merely illustrate that the fixed *search radius* is too high to separate two distinct structures blending into each other, a large overlap may show that the *search radius* is too low, causing a single structure to split into two. To remedy the latter, we merge segment inter-

vals in each list of preliminary segments if they overlap one another by more than half their widths. Lastly, the two lists of merged preliminary segment intervals are merged into one list under the previously mentioned condition. Here, we will keep every interval that existed in both lists. This may also have resulted from merging intervals from both lists into one bigger segment.

The final list of segments contains intervals representing phases of rests and grasps. The phases inbetween two consecutive segment intervals contain the motion transitioning from the one segment into the next.

In our extended version of the merge step, we additionally reproject candidate intervals from a merge back into the SSSM and check if the region covered by the corresponding square is populated by at least 95%. We only perform the merge if it does. An exemplary result from the segmentation and merge step is displayed in Figure 2 (c).

5 Clustering Approach

Once a set of motion trials has been segmented into primitive data units by extracting square-like regions from the main diagonal of a SSSM, we can group similar segments and moreover group similar motions within the set of trials.

5.1 Clustering of Primitives

Primitive data units are represented by squares along the main diagonal of a SSSM. Looking more closely at the populated areas in the SSSM, we find an interesting structure mainly consisting of off-diagonal blobs. These blobs indicate similarity between motion segments (see Figure 2). We will avail ourselves of this structure for clustering motion segments based on their similarity with respect to *segment area coverage* and *path coverage*, respectively, which are described below.

Consider a set of motion trials segmented into primitive data units $\mathcal{I} = I_1, \ldots, I_K$. In order to keep track of pairwise similarities within $\mathcal{I}$, we build a similarity graph $G_{\mathcal{I}}$. Each primitive $I_k, k = 1, \ldots, K$, is associated with one node in a similarity graph $G_{\mathcal{I}}$. Two nodes I_v, I_w will be connected by an edge if they are considered similar (based on the similarity measures described below). The final graph will consist of several strongly connected components representing clusters of similar types of primitive data units.

As the segmentation is performed per motion trial, but clustering is aimed at inter-trial comparison of primitive data units, we have to construct the local neighbourhood for each pair of primitives. The comparison of the primitives is based on the resulting sets of local neighbours which, for clustering, are not further converted into a SSSM.

Segment Clustering by Segment Area Coverage A simple straightforward approach for comparing primitive data units uses the area covered by each off-diagonal blob restricted to the range of the compared primitives in their SSSM.

If this area is sufficiently covered/populated, we add an edge between the nodes representing the compared primitives. This approach, however, is incapable of representing a temporal alignment of the segments needed for re-use of the data in, e.g. motion synthesis. Also note that while the SSSM here is immensely useful for explaining the underlying concept of this approach, we in fact count the number of nearest neighbours in each of the relevant sets.

Segment Clustering by Path Coverage A classical way of searching for the best temporal alignment of two time series $Q = \{q_1, \ldots, q_N\}$ and $V = \{v_1, \ldots, v_M\}$ is dynamic time warping (DTW). The alignment is given as an optimum cost warping path $P_{Q,V}$ between Q and V. A path $P_{Q,V}$ of length L is constituted by a sequence $\pi = \{\pi_1, \ldots, \pi_L\}$ of index pairs $\pi_l = (n_l, m_l) \in [1, N] \times [1, M] \subset \mathbb{N} \times \mathbb{N}$ for $l \in [1, L] \subset \mathbb{N}$ into Q and V subject to constraints such as $q_{n_l} \leq C \cdot q_{n_{l+1}}$ and $v_{m_l} \leq C \cdot v_{m_{l+1}}$ thus limiting the path's *slope*.

This kind of motion matching has been elegantly solved by Krüger et al. [27] using the sets of local neighbours in a neighbourhood graph. They represent subsequence DTW as kd-tree-based fixed-radius nearest neighbour search in the motions' feature set. For details see [27].

We compute the warping path for each pair of motion segments. Such a path is considered to be valid if it sufficiently covers the segments and does not fall below a certain length. If we find a valid warping path between two segments we record this information in the graph $G_{\mathcal{I}}$ by adding an edge between the two nodes representing the compared primitives.

This algorithm was adapted from Krüger et al. [19] and Vögele et al. [6].

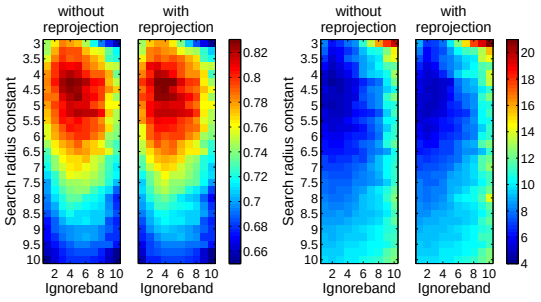
5.2 Clustering of Motion Sequences

After clustering of the segments, we are able to represent each trial by a sequence of IDs. Each ID represents a specific cluster of primitive data units. These sequences of IDs are subsequently used to group similar motion trials: First, each motion trial's sequence of IDs is processed such that it is free from successive identical IDs, e.g. 11123331 becomes 1231. For all pairs of sequences we compute their weighted longest common subsequence (LCS), i.e. we divide the pairwise LCS by the minimum length of the compared sequences. Based on this similarity measure, motion trials are grouped into final motion trial clusters.

Finally, each cluster is assigned to the class of the motion trial occurring most frequently in the cluster. Clusters containing exactly one motion trial will be disregarded and later counted as being unidentified.

6 Results

This section summarises achieved results for segmentation and clustering. We will start with an evaluation of different values for the two important segmentation parameters. Subsequently, we will compare our segmentation approach with the method proposed by Vögele et al. [6]. Concluding this section we will present results from clustering found motion segments.



(a) Overlap union ratio (b) Cut localisation

Fig. 3. Results of the parameter evaluation with respect to the presented segmentation quality measures. *Search radius constant* was plotted against *ignoreband* without (each left) and with (each right) reprojecting candidate intervals from merges. The optimum value for *overlap union ration* is 1.0 and for *cut localisation* 0.0.

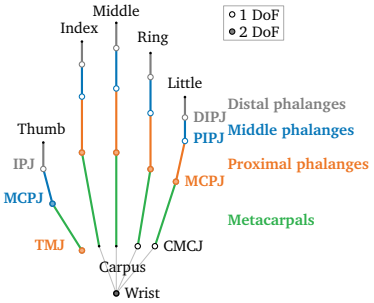


Fig. 4. Schematic kinematic chain of the hand model used in this work. Abbreviated joints spell out (proximal, distal) interphalangeal joints (PIPJ, DIPJ, IPJ), metacarpophalangeal joint (MCPJ), carpometacarpal joint (CMCJ) and trapeziometacarpal joint (TMCJ).

6.1 Parameter Evaluation

The parameter space of the discussed method is determined by the *search radius constant* R and the *ignoreband* B. In our evaluation, we have iterated over both values with respect to the two different quality criteria, *overlap union ration* and *cut localisation*, which our segmentation evaluation is based on. The results are given in Figure 3. Each parameter has a separate contribution in the overall segmentation results as the *search radius* is responsible for the structure of the sets of local neighbours, and hence determines the structure of populated and empty regions in the SSSM. The *ignoreband* has a major influence on the minimum detectable segment interval width.

The results demonstrate that the method performs well for a range of parameters, i. e. for $R \in [3.5, 5.5]$ and $B \in \{2, 3, 4, 5, 6\}$. Based on optimisation in this parameter space we have chosen $R = 4.25$ and $B = 4$ for all the experiments presented below.

6.2 Segmentation

To evaluate our segmentation algorithm, 50 prehensile motions from our database of controlled grasps were randomly chosen and annotated as belonging to one of five phases: rest, reaching, grasp, retraction, and rest. Reaching and retraction are transitional phases. These manually segmented motions will be referred to as *reference (segmentation)* as opposed to *automatic* or *computed segmentation*. In terms of features, we have chosen to stack five frames $[f_{i-6}, f_{i-1}, f_i, f_{i+1}, f_{i+6}]$ leading to a 75-dimensional feature vector and a *search radius* of $r = 4.25 \cdot \sqrt{75}$ [cm] ≈ 37 [cm]. For better comparability with [6] we have converted the intervals found by our segmentation into a sequence of cuts and regard each two consecutive cuts as an interval. In order to assess the quality of a computed

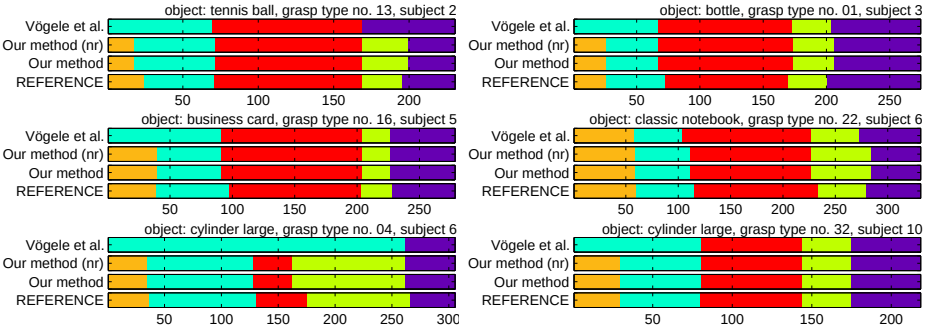


Fig. 5. Segmentation results for a number of annotated motion trials. For each trial, the bottom row displays the reference annotation. The two middle rows compare the reference annotation to our results and the top row depicts results of Vögele et al. [6]. We use '(nr)' throughout figures and tables to abbreviate 'no reprojection' (of candidate segment intervals).

segment interval I_c with respect to a reference segment interval I_r we use the following measures:

Quality Measures The *overlap union ratio* relates the overlap of two segment intervals I_r, I_c to the width of their union, $\frac{|I_r \cap I_c|}{|I_r \cup I_c|}$. In case of multiple computed segment intervals overlapping a reference interval we use the largest computed segment and disregard the others. Overlap union ratio is invariant to the position within a complete overlap. It ranges from 0 (no overlap) to 1 (exact overlap).

The second measure (*cut localisation*) considers detected segments as a linear list of segment cuts. It computes the distance of start/end frames in a reference segment to the closest computed cut. The conversion from segment intervals to cuts is straightforward and is computed by simply concatenating the interval boundaries and dropping the motion's first and last frame. Additionally, and to avoid favouring over-segmentation, the *total number of cuts* in the complete motion trial for both the computed and reference segmentation is reported.

Discussion of Results For a visual comparison of segmentation results refer to Figure 5. Segments are colour-coded according to the best matched reference segment. Segments found by our methods highly coincide with the reference segments in both, number of found segments and position of these segments within each motion trial. Our approach outperforms that of Vögele et al. [6] which often misses segments with respect to the reference segmentation.

This tendency is confirmed in the evaluation of the segmentation quality based on the aforementioned measures. Figure 6 shows histograms of absolute cut location offsets and number of cuts identified in the motions from the three methods. Our methods not only mainly find the correct number of (four) cuts for each motion but also with little frame offset with respect to the reference segmentation. Mean values and standard deviations of the quality measures used

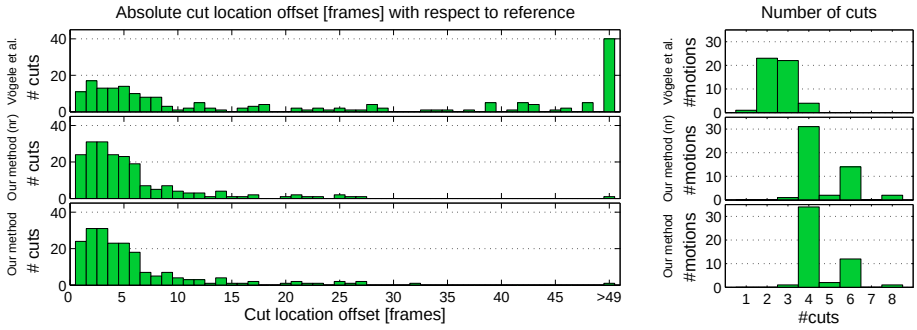


Fig. 6. Cut localisation (absolute number of frame offsets) and number of cuts with respect to to the reference segmentation. Reference segmentations consist of four cuts.Top row depicts results from the approach of Vögele et. al., the second row depicts our approach without reprojection and the last row is our approach with reprojection.

are listed in Table 2. Here, our methods reach an *overlap union ratio* of almost 0.84 with a very low standard deviation of 0.15. The combination of mean *cut localisation* of our method with and without reprojection (5.09 frames and 4.85 frames with a standard deviation of 6.85 and 6.43 frames) and mean *number of found cuts* (4.58 and 4.74 cuts with a standard deviation of 1.01 and 1.14 cuts) further confirm that our segmentation closely matches the reference.

6.3 Clustering

For clustering motion trials we used an excerpt of the controlled transport setup in our database. This excerpt covers all objects featured in the experiment and was divided into ten sets based on the test person. Hence, each class in this section represents an object. Unlike Krüger et al. [27], we chose to allow $\{(1,1),(1,2),(2,1),(1,4),(4,1)\}$ as warping steps to account for the mainly static nature of our primitive data units. A segment I_a is considered to sufficiently cover a second segment I_b if the off-diagonal area of I_a and I_b in their SSSM is populated by at least 0.66%. For path coverage this refers to coverage in the horizontal and vertical extent of a computed warping path.

As a side effect of how clusters are affiliated with motion trial classes, each cluster represents one class, but classes may spread multiple clusters. In this sec-

Table 2. Mean and standard deviations for different methods and different evaluation measures. *Overlap union ratio* ranges from 0 (no overlap) to 1 (exact overlap), *cut localisation* is measured in frames and the *number of cuts* in cuts.

Overlap union ratio			Cut localisation			Number of cuts		
Method	Mean	Std.	Method	Mean	Std.	Method	Mean	Std.
Vögele et al.	0.525	0.369	Vögele et al.	25.305	32.229	Vögele et al.	2.580	0.673
Our method (nr)	0.837	0.153	Our method (nr)	4.845	6.429	Our method (nr)	4.740	1.139
Our method	0.838	0.151	Our method	5.090	6.851	Our method	4.580	1.012

tion, we will present measures for assessing the quality of our clustering process and discuss the results.

Quality Measures In order to evaluate the accuracy of this assignment we use *cluster purity* as well as *precision*, *recall*, and F_1*-score* of the clustering. *Purity* measures the quality of a cluster by putting the number of correctly assigned motion trials in relation to the total number of motion trials. This does not take into account the number of clusters with respect to the number of actual classes, so we also give the total number of clusters and the number of classes (objects) to be represented by them.

Two trials should only be in the same cluster if they are similar and should be in different clusters if they are dissimilar. Based on this, we can derive the *precision* of the clustering as the number of (pairs of) trials correctly-grouped into the same cluster with regard to the total number of (pairs of) trials in these clusters. This quantifies the correctness of the separation of dissimilar trials into different clusters, or, put differently, the amount of correct predictions. Conversely, *recall* puts the number of trials correctly grouped into the same cluster in proportion to the number of trials that should have been grouped into the same cluster. This measures the success of avoiding separation of similar trials into different clusters or the ability to group trials by similarity. Finally, the F_1*-score* combines precision and recall through their harmonic mean, thus conveying the balance between the both.

Discussion of Results Figure 7 illustrates results obtained from clustering motion trials while Table 3 provides quantitative results obtained from evaluating the motion trial clustering. As can be seen from the table all test sets reached high cluster purity while the number of clusters nearly match the number of classes/objects. It should be noted that because we are basing computation of precision (and recall) on pairs of motion trials, incorrectly clustered trials strongly influence precision. This effect is less pronounced for recall.

The bottom half of Table 3 illustrates that we can reach high precision and recall values in many cases for clustering by path coverage. The minimum value for precision (recall) amounts to 0.71 (0.832). The top half of the table summarises results for clustering by segment area coverage. Overall, values are slightly lower than for clustering by path coverage. Minimum precision drops to 0.56, which is due to the fact that three different classes are identified as equivalent, and hence share the same cluster and heavily influence precision (compare Figure 7 (c)).

7 Limitations

Discussion of Segmentation Limitations While the approach of Vögele et al. [6] tends to miss short segments in particular, both our proposed methods are more likely to over-segment the motions (see Figure 8). This is due to the fact that our algorithm has posed a relatively strict condition on how to find the segments

Table 3. Results of the clustering for all sets. The last column contains the mean (or, where appropriate, the total) of the quality measures. We use # to abbreviate 'number of'. In the listing *found* counts the number of motion trials grouped into clusters with other trials, *correct* is the number of correctly grouped similar trials, *incorrect* denotes the number of incorrectly grouped dissimilar trials, and *unidentified* lists the number of trials that could not be grouped with other trials. Because we do not measure precision and recall based on single class division but based on pairs of motion trials, these values cannot be directly derived from the number of found, correct, etc. trials.

segment area coverage											
criterion	set 01	set 02	set 03	set 04	set 05	set 06	set 07	set 08	set 09	set 10	mean/total
# clusters	11	10	14	12	9	11	10	9	12	11	10.9
# classes	11	11	11	11	9	11	11	11	11	11	10.8
purity	1.000	0.811	1.000	1.000	0.860	0.926	1.000	0.796	0.944	1.000	0.934
# trials	55	55	55	55	44	56	55	57	55	55	542
found	53	53	48	52	43	54	42	54	54	50	503
correct	53	43	48	52	37	50	42	43	51	50	469
incorrect	0	10	0	0	6	4	0	11	3	0	34
unidentified	2	2	7	3	1	2	13	3	1	5	39
precision	1.000	0.662	1.000	1.000	0.709	0.835	1.000	0.560	0.862	1.000	0.863
recall	1.000	0.961	0.837	0.939	0.890	0.944	1.000	0.953	0.887	1.000	0.941
F_1-score	1.000	0.784	0.911	0.969	0.789	0.886	1.000	0.706	0.874	1.000	0.892

path coverage											
criterion	set 01	set 02	set 03	set 04	set 05	set 06	set 07	set 08	set 09	set 10	mean/total
# clusters	11	11	14	11	10	13	12	10	11	11	11.4
# classes	11	11	11	11	9	11	11	11	11	11	10.8
purity	1.000	0.906	1.000	1.000	0.976	0.963	1.000	0.889	0.855	0.980	0.957
# trials	55	55	55	55	44	56	55	57	55	55	542
found	54	53	48	50	42	54	48	54	55	51	509
correct	54	48	48	50	41	52	48	48	47	50	486
incorrect	0	5	0	0	1	2	0	6	8	1	23
unidentified	1	2	7	5	2	2	7	3	0	4	33
precision	1.000	0.797	1.000	1.000	0.934	0.918	1.000	0.766	0.710	0.949	0.907
recall	1.000	0.961	0.837	1.000	0.899	0.832	0.929	0.916	0.891	0.959	0.922
F_1-score	1.000	0.871	0.911	1.000	0.916	0.873	0.963	0.834	0.790	0.954	0.911

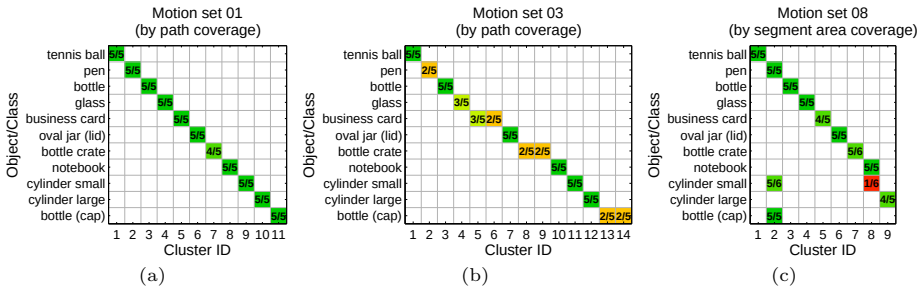


Fig. 7. Results of clustering motion sequences. Colouring is based on the number of motions in a class grouped together and ranges from green (similar trials grouped together) over yellow to red (trial grouped with dissimilar trials). Non-zero entries contain the number of motions of a class in a cluster (left) and the number of motions in that class (right). Multiple entries in a column represent a cluster covering multiple classes, multiple entries in a row indicate a class split into multiple clusters.

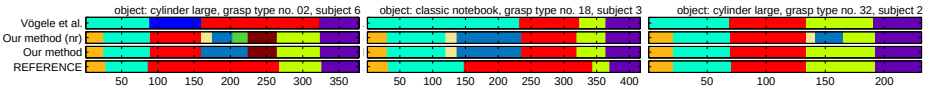


Fig. 8. Segmentation results illustrating over-segmentations by our algorithm. Reprojecting candidate intervals can help alleviating this issue (left and right).

during extraction of the diagonal contour. This leads to every gap in the SSSM within a square-like region along the diagonal causing our algorithm to start a new segment (segments stretch from the interior of the diagonal outwards). By contrast, the approach by Vögele et al. [6] introduce cuts whenever the main diagonal band is interrupted (segments wrap the around the diagonal structure from the exterior).

Discussion of Clustering Limitations By basing our clustering essentially on feature similarity we implicitly assume that these features are able to discriminate well between classes. For grasping this is not entirely true as the configuration of the hand strongly depends on the size and shape of the object as well as on the grasp applied to hold the object. This, on the one hand, can lead to our clustering separating single classes into multiple clusters (Figure 7 (b)) and, on the other hand, to grouping multiple classes of similar objects into the same cluster (Figure 7 (c)).

8 Conclusion and Future Work

In this paper, we presented a database of prehensile movements and a novel method for temporal segmentation of articulated hand motion. One of our goals was to present an effective method for segmentation and clustering of hand data. Our experiments confirm a high coincidence of our results with manual segmentation (cf. Section 6.2). Also, comparison to the clustered results of Vögele et al. [6] shows that both our evaluation methods (path coverage and segment coverage) yield higher accuracy scores (refer to Table 3). Particularly, the recall values are convincing compared to the relatively poor results by Vögele at al.

Acknowledgement. We would like to thank Fraunhofer IAO for providing us with the CyberGlove used to record the motion data. We also thank the authors of [6] for providing source code of their method for comparison.

References

1. Jörg, S., Hodgins, J., O'Sullivan, C.: The perception of finger motions. In: Proc. APGV. (2010) 129–133
2. Jörg, S., Hodgins, J.K., Safonova, A.: Data-driven finger motion synthesis for gesturing characters. ACM Trans. on Graphics **31**(6) (2012) 189:1–189:7
3. Zhao, W., Chai, J., Xu, Y.Q.: Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In: Proc. ACM SCA. (2012) 33–42

4. Tognetti, A., Carbonaro, N., Zupone, G., Rossi, D.D.: Characterization of a novel data glove based on textile integrated sensors. In: IEEE EMBS. (2006) 2510–2513
5. Dipietro, L., Sabatini, A.M., Dario, P.: A survey of glove-based systems and their applications. IEEE Trans. on SMC-C **38**(4) (2008) 461–482
6. Vögele, A., Krüger, B., Klein, R.: Efficient unsupervised temporal segmentation of human motion. In: Proc. ACM SCA. (2014)
7. Zhou, F., la Torre, F.D., Hodgins, J.K.: Hierarchical aligned cluster analysis for temporal clustering of human motion. IEEE Trans. on PAMI (2013)
8. Wheatland, N., Wang, Y., Song, H., Neff, M., Zordan, V., Jörg, S.: State of the art in hand and finger modeling and animation. Computer Graphics Forum **34**(2) (2015) 735–760
9. Arkenbout, E.A., de Winter, J.C.F., Breedveld, P.: Robust hand motion tracking through data fusion of 5DT data glove and Nimble VR kinect camera measurements. Sensors **15**(12) (2015) 31644–31671
10. Ju, Z., Liu, H.: Human hand motion analysis with multisensory information. IEEE/ASME Trans. on Mechatronics **19**(2) (2014) 456–466
11. CMU: Carnegie Mellon University Graphics Lab: Motion Capture Database (2013)
12. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation Mocap database HDM05. Technical Report CG-2007-2, Universität Bonn (2007)
13. Goldfeder, C., Ciocarlie, M.T., Dang, H., Allen, P.K.: The columbia grasp database. In: IEEE ICRA. (2009) 1710–1716
14. Feix, T., Romero, J., Ek, C.H., Schmiedmayer, H.B., Kragic, D.: A metric for comparing the anthropomorphic motion capability of artificial hands. IEEE Trans. on Robotics **29**(1) (2013) 82–93
15. Atzori, M., Gijsberts, A., Heynen, S., Hager, A.G.M., Deriaz, O., van der Smagt, P., Castellini, C., Caputo, B., Müller, H.: Building the ninapro database: A resource for the biorobotics community. In: Proc. IEEE/RAS-EMBS BioRob. (2012) 1258–1265
16. Beaudoin, P., Coros, S., van de Panne, M., Poulin, P.: Motion-motif graphs. In: Proc. ACM SCA. (2008) 117–126
17. Zhou, F., la Torre, F.D., Hodgins, J.K.: Aligned cluster analysis for temporal segmentation of human motion. In: Proc. IEEE CAFGR. (2008)
18. Min, J., Chai, J.: Motion graphs++: A compact generative model for semantic motion analysis and synthesis. ACM Trans. on Graphics **31**(6) (2012) 153:1–153:12
19. Krüger, B., Vögele, A., Willig, T., Yao, A., Klein, R., Weber, A.: Efficient unsupervised temporal segmentation of motion data. CoRR **abs/1510.06595** (2015)
20. Levine, S., Theobalt, C., Koltun, V.: Real-time prosody-driven synthesis of body language. ACM Trans. on Graphics **28**(5) (2009) 172:1–172:10
21. Mousas, C., Anagnostopoulos, C.N., Newbury, P.: Finger motion estimation and synthesis for gesturing characters. In: Proc. SCCG. (2015) 97–104
22. Ekvall, S., Kragic, D.: Grasp recognition for programming by demonstration. In: Proc. IEEE ICRA. (2005) 748–753
23. Kang, S.B., Ikeuchi, K.: Determination of motion breakpoints in a task sequence from human hand motion. In: Proc. IEEE ICRA. (1994) 551–556 vol.1
24. Zhao, W., Zhang, J., Min, J., Chai, J.: Robust realtime physics-based motion control for human grasping. ACM Trans. on Graphics **32**(6) (2013) 207:1–207:12
25. Feix, T., Pawlik, R., Schmiedmayer, H.B., Romero, J., Kragic, D.: A comprehensive grasp taxonomy. In: Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation. (2009)

26. Feix, T., Romero, J., Schmiedmayer, H.B., Dollar, A.M., Kragic, D.: The grasp taxonomy of human grasp types. IEEE Trans. on HMS **46**(1) (2016) 66–77
27. Krüger, B., Tautges, J., Weber, A., Zinke, A.: Fast local and global similarity searches in large motion capture databases. In: Proc. ACM SCA. (2010) 1–10