

# Supplemental Material: Efficient Unsupervised Temporal Segmentation of Human Motion

Anna Vögele<sup>†</sup>, Björn Krüger<sup>‡</sup>, Reinhard Klein<sup>§</sup>

Bonn University  
Institute for Computer Science II

## Abstract

*In this document we provide some additional information on the details of the original paper. Supplementary examples of segmentation results of both CMU and HDM05 are given as well as two additional sections on parameter evaluation and computation timings for the different steps of our algorithm. Key parameters such as the search radius and number of nearest neighbors are evaluated with respect to the achieved accuracy. The evaluation of timings breaks down the total timings of each part of the segmentation, thus illustrating the statements about the complexity in the respective sections of the paper.*

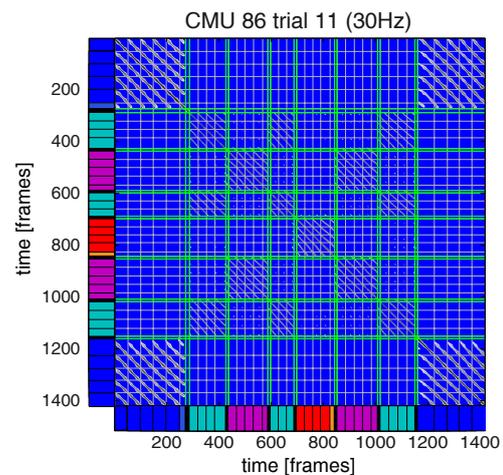
## 1. Extended Results

### 1.1. CMU

Figure 1 displays one exemplary result of segmenting and clustering a take from the CMU data base (subject 86) [CMU13]. The structure of the SSM and the resulting cuts indicate where the activities and the primitives can be found in the sequence. The green lines indicate the start and end frames of activities (the smaller blocks correspond to transitions), the gray lines indicate the primitives detected in the structure. The color bars left and below show how the primitives were assigned to different clusters.

### 1.2. HDM05

Figure 2 shows the segmentation and clustering results for data from the HDM data base [MRC\*07]: a sequence of walking with intermissions of sitting down on and getting up from various furniture. As can be seen, the recurring primitives of sitting (chair - blue, table - pink, floor - orange, floor (lying down) - purple) and the double steps (light blue) are clustered together. Also, most of the transition phases the



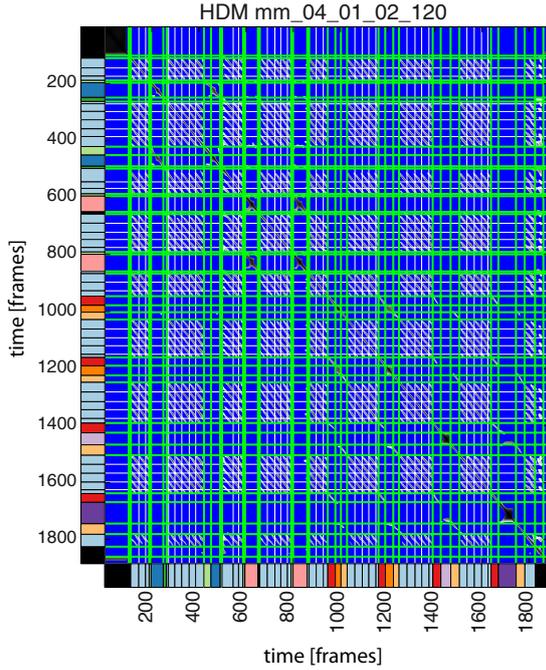
**Figure 1:** SSM computed for subject 86, trial 11 in the CMU data base. Motion primitives associated to coinciding classes of walking (blue) and different arm rotations (both arms - cyan, left arm - magenta, right arm - red) are color coded accordingly.

<sup>†</sup> voegele@cs.uni-bonn.de

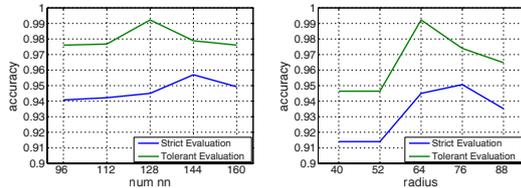
<sup>‡</sup> kruegerb@cs.uni-bonn.de

<sup>§</sup> rk@cs.uni-bonn.de

method has found occur repeatedly and are clustered accordingly (e.g. sitting down on chair - light green, getting up from chair - green).



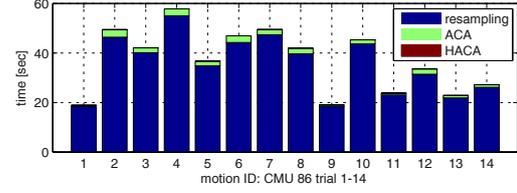
**Figure 2:** Segmentation and clustering of HDM take containing walking and sitting down. Recurring primitives are sitting (chair - blue, table - pink, floor -orange, floor (lying down) - purple) and the double steps (light blue, t-pose - black). Some transition phases such as sitting down on chair (light green), getting up from chair (green) also occur repeatedly and are clustered accordingly.



**Figure 3:** Evaluation of effect of search radius  $r$  and number of nearest neighbors  $k$  which are optimized for accuracy.

## 2. Parameter Evaluation

Our method is parameterized by the chosen feature set  $\mathcal{F}_{\text{stacked}}$  as well as the number of nearest neighbors  $k$  and the search radius  $r$  chosen for the construction of our neighborhood graph  $G_M$ . Figure 3 demonstrates how the segmentation results depend on the choice of parameters in practice and how far the choice of these items is reasonable for the accuracy values achieved in our evaluation.



**Figure 5:** Timings of HACA and ACA for all 14 sequences of subject 86. As can be observed, the preprocessing phase contributes substantially to the computation times. Note that the HACA computation is fast to the point that the red areas at the top of the color bars are hard to detect and may even vanish behind their black outlines.

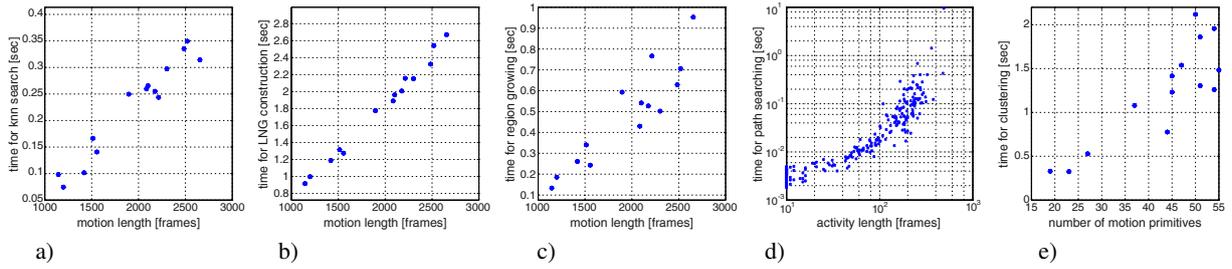
## 3. Comparison of Timings

On an Intel Core i7 4930K at 3.40GHz we were able to segment and cluster each motion in less than 15 seconds using our single threaded Matlab implementation. A Matlab implementation of (H)ACA running on the same machine is observed to take longer processing the same data set (for timings for CMU subject 86 see Figure 5). The most costly factor in the processing is obviously the frame reduction which Zhou et al. [ZITH13,ZITH08] introduce to save computation time throughout the later processing phases.

Figure 4 breaks down our total timings for the CMU examples to the timings of each part of the segmentation and clustering. The relationship of the first three segmentation steps, knn search, graph construction and activity separation are, in practice, approximately linear in the number of frames (Figure 4 a) - c). In theory, the worst case complexity for region growing is  $O(k \frac{n \cdot (n-1)}{2})$ , the worst case being that the first region grows from the first to the last frame of the input motion sequence. But this case has not been observed in practice. As discussed in the paper, the computation of path searches for primitive detection (part d) is a quadratic function in the number of frames per activity. Finally, the clustering step also computes in linear time (in the number of motion primitives).

## References

- [CMU13] CMU: Carnegie Mellon University Graphics Lab: Motion Capture Database, 2013. [mocap.cs.cmu.edu](http://mocap.cs.cmu.edu). 1
- [MRC\*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation Mocap Database HDM05*. Tech. Rep. CG-2007-2, Universität Bonn, June 2007. 1
- [ZITH08] ZHOU F., LA TORRE F. D., HODGINS J. K.: Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conference on Automatic Face and Gestures Recognition* (September 2008). 2
- [ZITH13] ZHOU F., LA TORRE F. D., HODGINS J. K.: Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35, 3 (2013), 582–596. 2



**Figure 4:** Scatter Plots for timings of the steps of our method. a) knn search, b) LNG construction, c) region growing, 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. Note that part d) is plotted on a log – log – scale. In practice, the clustering (e) also computes linear in the number of motion primitives.