Human Motion Parsing by Hierarchical Dynamic Clustering

Yan Zhang
yan.zhang@uni-ulm.de
Institute of Neural Information Processing, Ulm University
Ulm, Germany

Siyu Tang
stang@tuebingen.mpg.de
Department of Perceiving Systems, Max Planck Institute for Intelligent Systems
Tübingen, Germany

He Sun
h.sun@ed.ac.uk
School of Informatics, The University of Edinburgh
Edinburgh, UK

Heiko Neumann
heiko.neumann@uni-ulm.de
Institute of Neural Information Processing, Ulm University
Ulm, Germany

Abstract

Parsing continuous human motion into meaningful segments plays an essential role in various applications. In this work, we propose a hierarchical dynamic clustering framework to derive action clusters from a sequence of local features in an unsupervised bottom-up manner. We systematically investigate the modules in this framework and particularly propose diverse temporal pooling schemes, in order to realize accurate temporal action localization. We demonstrate our method on two motion parsing tasks: temporal action segmentation and abnormal behavior detection. The experimental results indicate that the proposed framework is significantly more effective than the other related state-of-the-art methods on several datasets.

1 Introduction

Human motion parsing is the task of partitioning a continuous human motion sequence into several meaningful primitive actions. It has various applications including animation, gait analysis and human-robot interaction, and is of great interest both in research and industry.

In recent years, human motion parsing is mainly investigated by model-based methods [1, 9, 22, 24], which require a large amount of training data and manual annotation. Thus, they are often not applicable when training data is limited or testing scenarios have visual domain shifting. For example, recording daily behaviors of patients (especially in video) are in general forbidden due to privacy issues, hence only few data is available to train the model offline. Moreover, a number of applications require fast response (e.g., ≤ 0.1 second) to the sensory input, such as detecting falling and raising the alarm, yet the supervised methods usually have unsatisfactory lags.

These limitations can be overcome by unsupervised methods of human motion parsing [8, 14, 32, 35]. Despite being applicable to uncontrolled real-world scenarios, an essential difficulty of the unsupervised method is to aggregate temporal local features (e.g., frame-wise body skeleton data) to the patterns of actions spanning in longer time durations, without the support of learned models as in the supervised methods. Thus, it is expected to setup a bottom-up pipeline to generate patterns of different actions, which can be subsequently differentiated by straightforward metrics like the Euclidean distance. Referring to the literature of action recognition [20, 28], local input features are encoded according to a codebook, then fused by a temporal pooling scheme. The outstanding performances with simple classifiers, like linear SVM, can indicate that such bottom-up aggregation is effective. Nevertheless, each video is trimmed to contain only one action, hence the temporal pooling step is simply to compute the average of encoded features of the entire video.

When processing natural untrimmed input streams, the temporal pooling scheme becomes highly nontrivial, since it is required to determine the temporal boundaries of different actions in a video. Without the support of supervised models, in which the temporal durations can be learned from massive data, e.g. [1], solving such problem is extremely challenging. First, human behaviors have a large range of temporal scales and intra-person variations. Second, determining accurate temporal durations requires precise action representations, and vice versa, leading to a "chicken-egg" problem that two inherently difficult tasks, namely action representation and action temporal localization, are coupled.

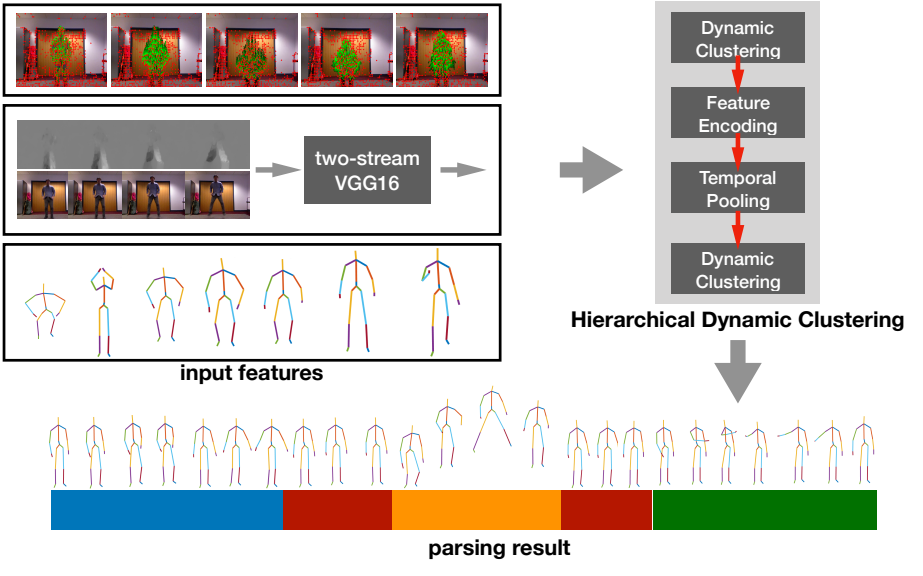

Figure 1: Illustration of our proposed method. The input features are generic and can be IDT+FV [28], outputs of convolutional nets [25] and motion capture data. The HDC pipeline comprises four components and produces a 1-D piece-wise constant sequence indicating the human parsing result. In this figure, colors denote different types of actions.

To address these challenges, in this paper we propose a hierarchical dynamic clustering framework, which is illustrated in Figure 1. Referring to the Bag-of-Words method for action analysis [20], in our framework the codebook is learned via dynamic clustering [32]

and the feature aggregation step comprises feature encoding and temporal pooling. With the merits of dynamic clustering, the number of the clusters is not necessary to be known in advance, but is estimated from the input data stream and is determined by the nature of human motion continuity. As it is unsupervised, self-evolving and efficient to process the input data stream (faster than standard k-means and spectral clustering as discussed in [32]), it could be employed in a large number of applications both in research and industry.

Another key contribution of our work is the study of temporal pooling methods for unsupervised human motion parsing. Besides validating the standard dense sliding window scheme [5], we propose two effective pooling schemes: kernelized cut-based pooling which is derived from the work [8, 23], and motion energy-based pooling which is inspired by the work [13, 31], in which the motion energy measure is proposed based on the concept of cognitive attention. We demonstrate our method on two motion parsing tasks: temporal action segmentation and abnormal action detection. The experimental results indicate that the proposed framework is significantly more effective than previous state-of-the-art in terms of accuracy and efficiency. For instance, for the temporal action segmentation task, our method achieves 0.87/0.91 of precision/recall values in the **CMUMAD** dataset. For the abnormality detection test, our method achieves 0.92 of accuracy in 0.07 second to recognize fainting as a novel behavior in the **BOMNI** dataset [3].

2 Related Work

Here we discuss several investigations in the past relevant to our methodology, as well as some studies on temporal action segmentation and abnormal behavior detection, which can be solved by human motion parsing.

Human motion parsing. Human motion parsing is highly related to continuous action understanding [2, 4, 22, 30] and action detection in untrimmed videos [1, 6, 21, 24, 29, 33], which are prevalent in recent years. However, we are motivated to propose alternative methods, which are unsupervised, efficient and not data-hungry. Our hierarchical dynamic clustering framework is motivated by the effective performances of bag-of-words pipelines for action recognition [20, 28] and the dynamic clustering [32] for temporal action segmentation.

Temporal action segmentation. Parsing a continuous human motion can directly yield the segmentation result. In the case of temporal action segmentation, the temporal boundaries are focused while the segmentation labels (generated by clustering methods) can be ignored. Zhou *et al*. [35] proposes aligned cluster analysis (ACA) that uses a kernel for time series alignment and obtain optimized temporal boundaries via dynamic clustering. Krüger *et al*. [12] proposes an efficient motion segmentation approach (EMS), in which a feature bundling method is used to generate compact and robust motion representations. Li *et al*. [14] addresses motion segmentation via temporal subspace clustering (TSC). Zhang *et al*. [32] proposes a dynamic clustering algorithm (DC) to segment human actions temporally, and systematically compare different clustering methods for codebook learning.

Abnormality detection. In [16], abnormalities are regarded as statistical outliers of the distribution of normal behaviors. In [18], a two-dimensional tree structure is established based on normal behaviors and abnormal behaviors in the test data are detected by matching. In [9], a deep autoencoder is trained to learn the temporal regularity of normal behaviors, and abnormalities are detected by detecting irregular patterns. These methods are unsupervised, but require training data to create a model. In contrast, our human motion parsing method detects novelties only based on previously preprocessed data in the same sequence. After

long-term observation, novel behaviors are normally equivalent to abnormal behaviors.

Comparison with dynamic clustering [32]. Our method is distinguished from [32] according to the following two aspects: (1) our method uses dynamic clustering to process high-level action patterns, while [32] uses k-means. (2) our method uses novel temporal pooling methods, i.e., kernelized cut-based and motion energy-based pooling, while [32] uses sliding window or dataset-dependent approaches.

3 Methods

3.1 Hierarchical dynamic clustering framework

Our hierarchical framework is proposed based on the *bag-of-words* pipeline, as illustrated in the block of **Hierarchical Dynamic Clustering** in Figure 1. In our work, we use dynamic clustering for codebook learning and soft-assignment for feature encoding due to validated performances on action analysis [20, 32]. Therefore, we can create a bottom-up pipeline to convert low-level input features to action patterns in an unsupervised manner.

While it is known that dynamic clustering outperforms k-means in terms of codebook learning (grouping low-level input features) [32], there have not been extensive studies on how well dynamic clustering is able to group high-level action patterns. Because of this, we replace the k-means module in a previous studies [32] by other methods, and evaluate their performances. We first conduct a qualitative study here, while the quantitative results are discussed in Section 4. We find that dynamic clustering outperforms k-means and spectral clustering [19, 27] for action pattern grouping, and hence use dynamic clustering after temporal pooling, as shown in Figure 1.

We estimate the improved dense trajectories and derive Fisher vectors [28] of the *drinkwaterS1R1* video in the **RADL** dataset [17], which consists of three actions (i.e., *fetching water*, *pouring water* and *drinking water*). The Gaussian mixture model has 32 components and is trained from the videos *drinkwaterS1R2* and *drinkwaterS1R3*. The Fisher vectors have 12,288 dimensions, and are extracted using a sliding window of 50-frame length and 1-frame stride. We apply PCA to reduce the dimension of vectors to 50 before performing the clustering algorithm.

The results are shown in Figure 2, in which the sequence of the high-dimensional feature vectors are visualized by t-SNE [15]. One can see that the dynamic clustering algorithm is able to generate and update cluster structures in an online fashion, leading to comparable results with the human annotation. The k-means and the spectral clustering algorithms falsely parses *pouring water* and *drinking water* actions into non-consecutive segments.

3.2 Temporal Pooling

A standard temporal pooling method is to use a time sliding window on the input sequence [6]. However, due to large variations in the temporal scales of different motions, a unique pre-defined window size leads to fusing information of different actions into one pattern and hence often produces poor motion features, which is illustrated in Figure 3. When the window size is too large, many different short-term actions are merged, leading to imprecise temporal localization. When the time window is too small, long-term actions are split and also the temporal structure represented by each action pattern will be limited.

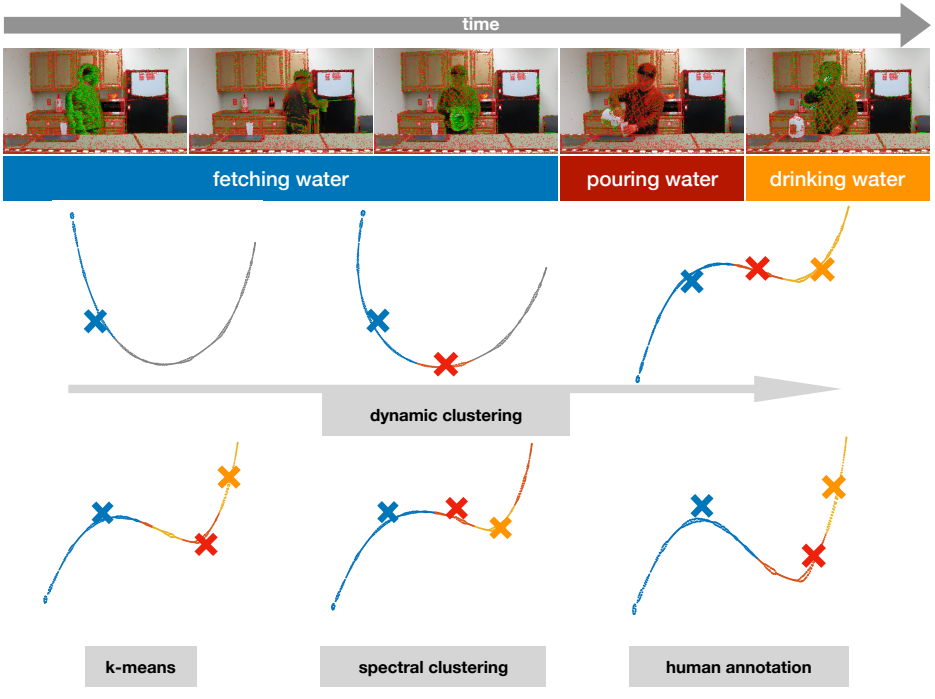

Figure 2: Qualitative comparison of different clustering methods. The first row shows 5 frames of a video with rendered dense trajectories. In the second row, the trajectories denote the 2D projections of the sequence of dimensionality-reduced Fisher vectors. The crosses denote the 2D projections of the cluster centroids. The gray color denotes the unprocessed Fisher vectors and other colors denote action labels.

To overcome this, we propose two alternative pooling methods: the kernelized cut-based pooling that creates an embedding space making action patterns more distinguishable, and the motion energy-based pooling that utilizes attention to categorize human motions into *still poses* and *moving actions*. In particular, the motion energy-based pooling is simple yet effective for encoding representative motion features, because the temporal window is retrieved from the energy measures that could be considered as another information source, avoiding the classic "chicken-egg" problem between the temporal pooling window and encoded motion features.

Kernelized cut-based pooling. Our kernelized cut method is developed on top of the dense sliding window scheme: We first run the sliding window aggregation with 1-frame stride, then use the kernelized cut method to determine disjoint time windows, and at last aggregate the encoded input features within each individual time window to derive the parsing result.

Referring to [8, 14, 23], one can create a fully connected graph, design a kernel function to specify the edge weights and perform graph cut. Here an essential question is to design the kernel function. Inspired by the work of [8], we propose a kernel function to measure feature similarity with consideration of local temporal structures. Given a set of features $\{\boldsymbol{x}_i\}$, the

kernel function $k(\cdot,\cdot)$ is given by

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = k_s(\boldsymbol{x}_i,\boldsymbol{x}_j)\cdot k_t(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp(-\alpha\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2)\cdot\frac{\langle\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{x}}_j\rangle}{\|\bar{\boldsymbol{x}}_i\|_2\cdot\|\bar{\boldsymbol{x}}_j\|_2} \tag{1}$$

where $\bar{\boldsymbol{x}}_i = \frac{1}{|\mathcal{T}_i|}\cdot\sum_{k\in\mathcal{T}_i}\boldsymbol{x}_k$, $\mathcal{T}_i$ denotes a temporal range centered at $\boldsymbol{x}_i$ and α is a positive constant. In the kernel function, the first component k_s measures the spatial similarity straightforwardly; the second component k_t incorporates temporal information and hence can differentiate two identical features with different temporally local statistics. In addition, in our case the set $\{\boldsymbol{x}_i\}$ is obtained by the soft-assignment encoding [20] and hence is located within the domain of positive real numbers, leading to the fact that $k(\cdot,\cdot)$ is positive-definite and its codomain is non-negative. In our empirical trials, our proposed kernel performs better than the one in [8]. Figure 3 illustrates the benefit of a kernelized cut.

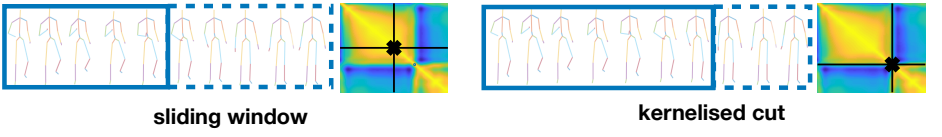

Figure 3: Comparison between sliding window pooling (left) and kernelized pooling (right) with the kernel function (1). One can see that the kernelized cut can find the optimal segment with small inter-class and large intra-class similarities.

Given the kernel function, one can perform two types of cuts: sequential cut as in [8] and batch cut as in [14, 23]. In the sequential cut, the number of cuts are highly influenced by the pre-fixed time range: the shorter the time range is, the more cuts are produced. One can note that a high recall value can be achieved in the result of human motion parsing, if the number of cuts is large. In the batch cut, one is required to specify the number of clusters in advance. Similar to specifying the time range in the sequential cut, a larger number of clusters will result in a larger number of cuts. Nevertheless, a smaller number of clusters can also lead to large number of cuts in certain cases.

Motion energy-based pooling. Our proposed motion energy-based pooling scheme is inspired by the observation that most human motions can be categorized into two meta-classes, i.e., *moving actions* and *still poses*, and *moving actions* can easily gain cognitive attention. In our work, we compute a motion energy measure based on which time windows are determined and categorized to the two meta-classes. Then, in each time window the encoded features are fused and dynamic clustering is applied for each meta-classes separately.

The motion energy is calculated based on the indices of clusters that the input frames belong to, generated by the first dynamic clustering module. Afterwards, we employ a moving window to sequentially compute the motion energy, which is the number of transitions between clusters divided by the window length. To remove noise a Gaussian smoothing is subsequently performed. Then we detect peaks on the smoothed motion energy curve, retrieve windows around such peaks as the time periods of *moving actions*, and regard the remaining regions as *still poses*. See Figure 4 for illustration.

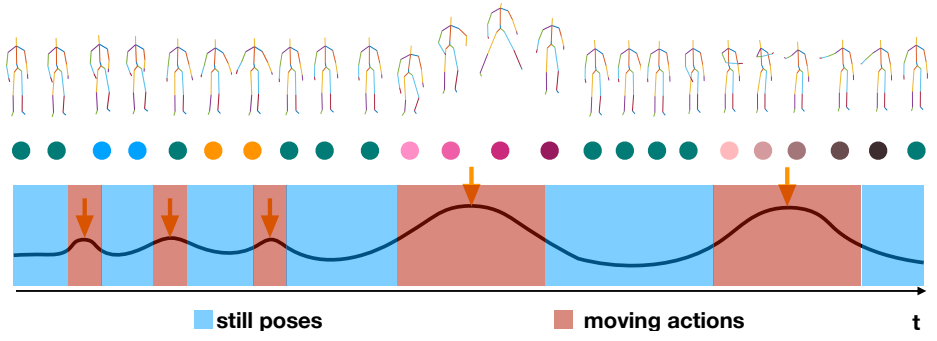

Figure 4: Illustration of the motion energy-based pooling. The color of dots denotes the cluster index, the curve denotes the motion energy measure and the arrows denote the detected peaks. One can see that the moving actions and still poses are differentiated.

4 Experiments

4.1 Temporal action segmentation

Datasets. We use the **CMUMAD** [11] and the **TUMKitchen** [26] datasets to validate the performance of our methods for temporal action segmentation. Specifically, **CMUMAD** contains 40 recordings (video and motion capture data) from 20 subjects, each of which comprises 35 different actions and null actions in between. **TUMKitchen** consists of 20 recordings from multiple types of modalities such as videos, motion captures and on-off sensors on the door. The action labels are annotated for the torso and the arms separately.

Evaluation metric & Input features. On the **CMUMAD** dataset, the true positive is defined as a segment having at least 50% overlaps with the ground truth yet the label has not been detected before. On the **TUMKitchen** dataset, we evaluate how effective our algorithm can locate the segment boundaries, so a true positive is the boundary detected within ± 7 frames (approx. 0.25 second) of a ground truth boundary.

Our algorithm is generic and able to process diverse types of features from different modalities. In our experiment, we use the following features: (1) the *fc8* layer output of the two stream deep neural model VGG-16 [7, 25], which is denoted as VGG16. (2) The concatenation of all the 3D coordinates of the joints in each individual frame, which is denoted as JointLocation. (3) The relative 2D angles between each two adjacent 3D body parts, which is denoted as RelativeAngle. (4) The quaternion representation of the 3D rotations (e.g., yaw, pitch and row) of the joints, which is denoted as Quaternions. The evaluation metric and the input features are identical to the one used in [32] to enable a fair and direct comparison with the state-of-the-art.

Comparing with the state-of-the-art. We first compare our methods with the following existing methods: TSC [14], ACA [35], EMS [12] and DC [32], as discussed in Section 2. The results on the **CMUMAD** dataset are shown in Table 1. HDC_{ME}, HDC_{KC-S} and HDC_{KC-B} denote our proposed hierarchical dynamic clustering with the motion-energy based pooling, the kernelized sequential cut-based pooling and the kernelized batch cut-based pooling method, respectively. Note that, DC [32] compared in Table 1 utilizes the prior knowledge to dedicatedly design the temporal pooling step for **CMUMAD**, whereas our proposed temporal pooling methods are generic and do not use any dataset specific in-

formation. Nevertheless, HDC_{ME} outperforms other work for most of the input features by large margin. The performance of the kernelized cut pooling methods is inferior to the motion energy pooling. The reason is that the temporal pooling is conducted based on a pre-defined sliding window size. Without utilizing the prior knowledge of the dataset, which is performed in most of the previous works, it is hard to derive the representative action patterns for various time durations.

Method	VGG16 [7]	JointLocation	RelativeAngle	Quaternions
TSC [14]	0.01/0.20/0.02	0.10/0.30/0.15	0.05/0.29/0.09	0.05/0.29/0.09
ACA [35]	0.56/0.66/0.61	0.55/0.68/0.61	0.51/0.65/0.57	0.55/**0.66**/0.60
EMS [12]	0.67/0.73/0.70	0.34/0.78/0.47	0.47/0.89/0.62	0.60/0.51/0.55
DC [32]	0.44/0.60/0.51	0.82/0.86/0.84	0.63/0.64/0.63	0.63/0.52/0.57
HDC_{KC-S}	0.23/0.41/0.29	0.50/0.82/0.62	0.52/0.82/0.64	0.31/0.58/0.40
HDC_{KC-B}	0.14/0.40/0.21	0.39/0.86/0.54	0.37/0.85/0.52	0.18/0.61/0.28
HDC_{ME}	**0.72/0.82/0.77**	**0.86/0.88/0.87**	**0.87/0.91/0.89**	**0.76**/0.57/**0.65**

Table 1: Comparison with the state-of-the-art on the **CMUMAD** dataset. The results are shown in the format of *precision/recall/f-score*. The best results are in boldface.

The results on the **TUMKitchen** dataset are shown in Table 2. In terms of *f-score*, the proposed HDC_{KC-B} method shows superior performance to others. Interestingly, the performance of the HDC_{ME} method is not as outstanding as it is on the **CMUMAD** dataset. Our observation is that in the **TUMKitchen** dataset, the recorded persons perform certain actions most of the time, the variation for the motion energy is not significant, therefore it does not provide strong signals on how to segment the continuous motion. Nevertheless, the performance of HDC_{ME} is still rather competitive comparing to the previous methods, especially for the JointLocation of torso, suggesting that the combination of the hierarchical dynamic clustering and the motion energy based pooling is able to produce reliable human motion segmentation results regardless the type of motions in the test sequence.

Method	body part	JointLocation	RelativeAngle	Quaternions
TSC [14]	Torso	0.12/0.04/0.06	0.28/0.10/0.15	0.31/0.19/0.24
	Arms	0.42/0.28/0.34	0.29/0.38/0.33	0.28/0.56/0.37
ACA [35]	Torso	0.19/0.01/0.02	0.30/0.02/0.04	**0.40**/0.03/0.06
	Arms	0.36/0.08/0.13	0.36/0.10/0.16	0.34/0.09/0.14
EMS [12]	Torso	0.13/0.06/0.08	0.28/0.15/0.20	0.37/0.12/0.18
	Arms	0.26/0.12/0.16	**0.38**/0.27/0.32	0.34/0.09/0.14
DC [32]	Torso	**0.46**/0.15/0.21	**0.34**/0.12/0.18	**0.40**/0.26/0.32
	Arms	**0.49**/0.30/0.37	0.27/0.64/**0.38**	0.33/**0.68**/0.44
HDC_{KC-S}	Torso	0.24/**0.67**/0.35	0.23/0.44/0.30	0.27/0.49/0.35
	Arms	0.26/**0.52**/0.35	0.25/0.50/0.33	0.38/0.35/0.36
HDC_{KC-B}	Torso	0.32/0.54/0.40	0.23/**0.52/0.32**	0.29/0.64/**0.40**
	Arms	0.44/0.45/**0.44**	0.26/**0.72/0.38**	**0.44/0.46/0.45**
HDC_{ME}	Torso	0.42/0.54/**0.47**	0.24/0.45/0.31	0.23/0.39/0.29
	Arms	0.37/0.41/0.39	0.30/0.32/0.31	0.31/0.35/0.37

Table 2: Comparison with the state-of-the-art on the **TUMKitchen** dataset. The results are shown in the format of *precision/recall/f-score*. The best results are in boldface.

Analysis of the hierarchical clustering framework. Our approach is motivated by the observation that human motion can be decomposed at different temporal scales. With merits of the robustness of dynamic clustering, it could reliably parse human motions despite large variations of temporal scales of actions. To validate our hypothesis, we replace the last dynamic clustering step in the framework (Figure 1) with either k-means or spectral clustering, while retaining sliding window-based pooling. As shown in Figure 5 (a), the proposed framework improves the baseline methods considerably.

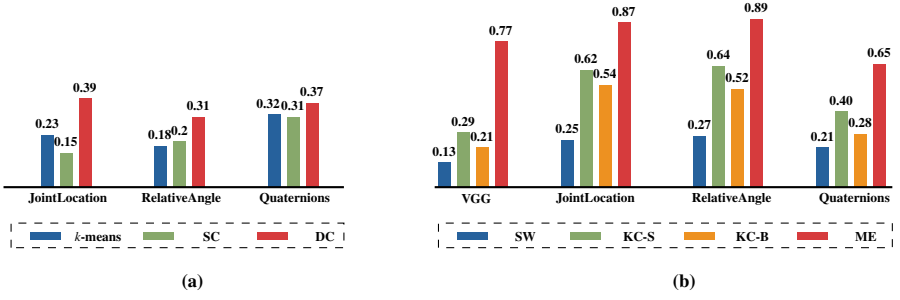

Figure 5: Results are evaluated by the f-score. (a) Comparison of clustering methods to parse torso motions in **TUMKitchen**, where SC and DC are spectral clustering and dynamic clustering respectively. (b) Comparison of the temporal pooling schemes on **CMUMAD**, where SW, KC-S, KC-B and ME denote the time sliding window, the sequential kernelized cut, the batch kernelized cut and motion energy, respectively.

Analysis of the temporal pooling schemes. To validate the effectiveness of the proposed temporal pooling methods, we further compare with the conventional dense sliding window approach (SW) with an empirically optimal window size. As shown in Fig. 5 (b), for various input features the proposed pooling methods consistently outperform the baseline method by noticeable margins. Particularly, the average improvement of *f-score* made by the motion-energy based pooling is 0.55, certifying our assumption that the motion energy term could be informative and utilized for human motion parsing.

4.2 Abnormal behavior detection

Dataset. For abnormal behavior detection, we use the first scenario of **BOMNI** [3]. The annotation of each video consists of the bounding box of the subject in each frame and 6 actions performed by the subject, which are sitting, walking, drinking, washing-hands, opening-closing-door and fainting. Figure 6 shows ten example frames.

Evaluation metric & Input features. We use the accuracy to evaluate the detection quality at the frame level, which is computed as the true positives divided by the number of frames belonging to the *fainting* action in the ground truth. The true positives are defined as the frames, which both belong to "novel" actions and the associated label (cluster index) is the majority in the parsing result within the period of *fainting*. In our setting, a "novel" action is regarded as a segment whose action label first appears along the temporal dimension. For example, if we have a sequence of 6 frames with action labels (or cluster indexes) $\{a,a,b,a,a,c\}$, the frames are annotated as $\{1,1,1,0,0,1\}$ for the novel behavior representation.

Figure 6: Five frames of the first video in the **BOMNI** dataset. In each frame, the annotated bounding box, the action label and the mask detected by Mask-RCNN [10] are presented.

We use the mask-RCNN model provided by [10] to extract the masks of the subjects, since neither pose estimation nor context representation of the entire frame is reliable. Then we resize the mask to patches of 40x40 of pixels, and perform distance transform following [18]. The feature vector is derived via vectorizing the transformed mask.

Results. The results of our methods as well as other state-of-the-art parsing approaches are shown in Table 3. One can see that HDC_{ME} is comparable with ACA and superior to others. However, HDC_{ME} runs significantly faster than ACA. Since the motion-based pooling mainly relies on processing the cluster indices, its computational load is considerably smaller than other pooling methods which require to compute the data similarity matrix. Also, comparing with dynamic programming used in ACA, subspace clustering used in TSC and k-means used in DC, the dynamic clustering modules in HDC lead to faster computation.

Algorithm	TSC [12]	ACA [32]	DC [37]	HDC_{KC-S}	HDC_{KC-B}	HDC_{ME}
Accuracy	0.28	**0.99**	0.33	0.81	0.81	0.92
Runtime (second)	1.63	2.69	0.16	0.18	0.31	**0.07**

Table 3: The results on the **BOMNI** dataset. The best ones are in boldface.

5 Conclusion

In this paper we propose a hierarchical dynamic clustering framework to parse human motions in untrimmed sequences, which is inspired by the unsupervised bag-of-words feature aggregation pipeline. To reliably cluster features with temporal structures, we employ dynamic clustering to create the codebook and group high-level action patterns. To obtain accurate temporal durations of actions, we propose unsupervised temporal pooling methods based on kernelized cut or motion energy. The experimental results demonstrate that our approach out-performances the state-of-the-art.

We leave the following two questions for future studies: (1) Our current unsupervised method uses the output produced by several feature extraction methods. It would be meaningful to build an end-to-end approach by combining these two techniques together. (2) Since unsupervised temporal pooling in untrimmed input sequences are not fully solved yet, we plan to improve temporal pooling techniques as future work.

Acknowledgements. This work is supported by a grant of the Federal Ministry of Education and Research of Germany (BMBF) for the project SenseEmotion.

References

[1] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6373–6382, 2017.

[2] Yu Cheng, Quanfu Fan, Sharath Pankanti, and Alok Choudhary. Temporal sequence modeling for video event detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2227–2234, 2014.

[3] Banş Evrim Demiröz, İsmail Ari, Orhan Eroğlu, Albert Ali Salah, and Laie Akarun. Feature-based tracking on a multi-omnidirectional camera dataset. In *5th International Symposium on Communications Control and Signal Processing (ISCCSP)*, pages 1–5. IEEE, 2012.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[5] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *International Conference on Computer Vision (ICCV)*, pages 1491–1498. IEEE, 2009.

[6] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision (ECCV)*, pages 768–784, 2016.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016.

[8] Dian Gong, Gérard Medioni, Sikai Zhu, and Xuemei Zhao. Kernelized temporal cut for online temporal segmentation and recognition. In *European Conference on Computer Vision (ECCV)*, 2012.

[9] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–742, 2016.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017.

[11] Dong Huang, Shitong Yao, Yi Wang, and Fernando De La Torre. Sequential max-margin event detectors. In *European conference on computer vision (ECCV)*, pages 410–424, 2014.

[12] Björn Krüger, Anna Vögele, Tobias Willig, Angela Yao, Reinhard Klein, and Andreas Weber. Efficient unsupervised temporal segmentation of motion data. *IEEE Transactions on Multimedia*, 19(4):797–812, 2017.

[13] Georg Layher, Martin A Giese, and Heiko Neumann. Learning representations of animated motion sequences - a neural model. *Topics in Cognitive Science*, 6(1):170–182, 2014.

[14] Sheng Li, Kang Li, and Yun Fu. Temporal subspace clustering for human motion segmentation. In *International Conference on Computer Vision (ICCV)*, pages 4453–4461. IEEE, 2015.

[15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[16] Markos Markou and Sameer Singh. Novelty detection: a review - part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[17] Ross Messing, Chris Pal, and Henry Kautz. Activity recognition using the velocity histories of tracked keypoints. In *International Conference on Computer Vision (ICCV)*, pages 104–111. IEEE, 2009.

[18] Fabian Nater, Helmut Grabner, and Luc Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2014–2021, 2010.

[19] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs: Spectral clustering works! *SIAM Journal on Computing*, 46(2):710–743, 2017.

[20] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016.

[21] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3131–3140, 2016.

[22] Haoquan Shen, Shoou-I Yu, Yi Yang, Deyu Meng, and Alexander Hauptmann. Unsupervised video adaptation for parsing human motion. In *European Conference on Computer Vision (ECCV)*, pages 347–360. Springer, 2014.

[23] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):888–905, 2000.

[24] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016.

[25] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014.

[26] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009.

[27] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

[28] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision (ICCV)*, pages 3551–3558. IEEE, 2013.

[29] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] Yan Zhang, Georg Layher, and Heiko Neumann. Continuous activity understanding based on accumulative pose-context visual patterns. In *Seventh International Conference on Image Processing Theory, Tools and Applications*, pages 1–6. IEEE, 2017.

[31] Yan Zhang, Georg Layher, Steffen Walter, Viktor Kessler, and Heiko Neumann. Visual confusion recognition in movement patterns from walking path and motion energy. In *International Conference on Smart Homes and Health Telematics*, pages 124–135. Springer, 2017.

[32] Yan Zhang, He Sun, Siyu Tang, and Heiko Neumann. Temporal human action segmentation via dynamic clustering. *arXiv preprint arXiv:1803.05790*, 2018.

[33] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *arXiv preprint arXiv:1704.06228*, 2017.

[34] Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *8th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–7, 2008.

[35] Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(3):582–596, 2013.