Human Action Segmentation Based on a Streaming Uniform Entropy Slice Method

Cheng Peng
*Macau University of Science and Technology*

Sio-Long Lo
*Macau University of Science and Technology*

Jie Huang
*Macau University of Science and Technology*

Ah Chung Tsoi
*University of Wollongong, act@uow.edu.au*

Publication Details
Human Action Segmentation Based on a Streaming Uniform Entropy Slice Method

Abstract
Segmentation of human actions is a major research problem in video understanding. A number of existing approaches demonstrate that performing action segmentation before action recognition results in better recognition performance. In this paper, we address the problem of action segmentation in an online manner. We first extend the clustering-based image segmentation approach into a temporal one, where hierarchical supervoxel levels for action segmentation are generated accordingly. We then propose a streaming approach to flatten the hierarchical levels into one based on Uniform Entropy Slice (UES), in order to preserve important information in the video. The flattened level contains the silhouette of a human with the structure of body parts labelled in different labels. We then combine the human structure information and the original video frames to strengthen the action in a video, which paves the way for accurate action recognition. The experimental results show that our online approach achieves satisfactory performance regarding action segmentation or recognition on various publicly available datasets, including the DAVIS dataset, the UCF Sports dataset, and the KTH dataset.

Keywords
slice, method, entropy, human, uniform, streaming, segmentation, action

Disciplines
Engineering | Science and Technology Studies

Publication Details
Human Action Segmentation Based on a Streaming Uniform Entropy Slice Method

CHENG PENG1, SIO-LONG LO1, JIE HUANG1, AND AH CHUNG TSOI2,3

1Faculty of Information Technology, Macau University of Science and Technology, Macau 999078, China
2Department of Computer Science, Chu Hai College of Higher Education, Hong Kong 999077, China
3University of Wollongong, Wollongong, NSW 2522, Australia

Corresponding author: Sio-Long Lo (slllo@must.edu.mo)

This work was supported by the Macao Science and Technology Development Fund under Grant 019/2014/A1 and Grant 112/2014/A3.

ABSTRACT Segmentation of human actions is a major research problem in video understanding. A number of existing approaches demonstrate that performing action segmentation before action recognition results in better recognition performance. In this paper, we address the problem of action segmentation in an online manner. We first extend the clustering-based image segmentation approach into a temporal one, where hierarchical supervoxel levels for action segmentation are generated accordingly. We then propose a streaming approach to flatten the hierarchical levels into one based on uniform entropy slice, in order to preserve important information in the video. The flattened level contains the silhouette of a human with the structure of body parts labeled in different labels. We then combine the human structure information and the original video frames to “strengthen” the action in a video, which paves the way for accurate action recognition. The experimental results show that our online approach achieves satisfactory performance regarding action segmentation or recognition on various publicly available data sets, including the DA VIS data set, the UCF Sports data set, and the KTH data set.

INDEX TERMS Action segmentation, streaming uniform entropy slice, supervoxel tree, action recognition.

I. INTRODUCTION

Human action segmentation has been very remarkable in many research and application fields, such as scene understanding [1], [2], driverless vehicles [3], [4] and human-robot interactions [5]. With the technology of video segmentation has a great progress over the past decades and a variety of segmentation methods [6]–[9] have been proposed to process with the video, which makes easy to analyze the video. It seems that the video segmentation has emerged as a credible first step in early processing of unlimited length of videos, it does not need to make any assumption on a static background as earlier methods need to do so.

Without considering the temporal coherence, image segmentation methods can be used to segment the video on a frame-by-frame basis, but it could not express even small frame-to-frame changes as a continuous function in general. The hierarchical video segmentation method performs quite well for it can segment the video into hierarchical levels which contain a rich multiscale decomposition of the video, from low level to high level, the segmentation become over segmentation to under segmentation, but it is a challenge to use the results to analyze the video. Despite remarkable progress in recent years, video segmentation still remains some challenging problems such as lacking for the precise action boundaries in space-time or the deeper information for the video.

Human action segmentation can avoid the traditional video segmentation limitations which aim to segment human action. The precise human silhouette boundaries have a strong influence on action recognition or classification. For action recognition, actions always happen in the temporal and spatial domain in a continuously frames, the time sequence relations between successive frames will be strong in the temporal domain and in the spatial domain, body parts will also affect each other. Our human action segmentation method is different with most of recent works which are aim to segment human’s silhouette, we were inspired by hierarchical video segmentation method to proposed our human action segmentation method which is a little different with the traditional hierarchy, in our hierarchies, high levels will be over segmentation with the human action and background information, but the low level will only contain the human’s silhouette. By flattening the hierarchy into one level, the result can satisfy the requirement of the action recognition. Our result contents the human’s silhouette with the structure of human, it not only can demonstrate the static body parts, but also can show the articulating body parts.
In this paper, we proposed our method for spatio-temporal human action segmentation. We segment the foreground human into several regions and the background into one region to get hierarchical levels. And then, the levels can be flattened into one level. A streaming mode for the flattening method is proposed to improve the segmentation efficiency. The main contributions of this paper are summarized as follows:

1) We propose a novel hierarchical human action segmentation method which is able to incorporate the human action into hierarchical levels.

2) We propose a streaming mode for uniform entropy slice which can process long video with high definition.

3) We extract the motion boundary trajectories on the condition of the segmentation result to show the human action segmentation result can improve the accuracy of the human action recognition.

The rest of the paper is organized as follows: Section II reviews the research work related to action segmentation and recognition in videos. Section III and IV introduce our proposed approach to address the problem of action segmentation and recognition, respectively. The experimental results are shown in Section V. Finally, we draw some conclusions and provide some directions on future work in Section VI.

II. RELATED WORK

Human action segmentation and recognition [10], [11] are very challenging topics in computer vision. Human action segmentation can benefit the action recognition [12], [13]. In [14], they showed that using the foreground moving regions can perform better than using the full video in action classifications. The structure of action representations might benefit it with good models of human segmentation property [15]. Jhuang et al. [16] showed that the action recognition can be impacted by the precise human contour boundaries and found that the ground truth information on human provides significant help towards better action recognition.

Most researchers resort to working with superpixels or supervoxels [17] to segment the video for the reason that supervoxel plays an important role in video segmentation tasks in which moving objects in the video are detected and their external boundaries are obtained. This is due to the observation that hierarchies of supervoxel can process the video into a multiscale decomposition with more information for later analysis [18], [19] than other approaches using the concept of supervoxels to video segmentation. Xu and Corso evaluated five methods based on supervoxels [20]. The basic idea is that supervoxels have great potential in advancing video analysis methods, as superpixels have for image analysis. They found that the hierarchical graph-based segmentation using weighted aggregation methods perform best and almost equally-well.

Hierarchical video segmentation [21] perform very well due to the way in which multiscale region similarity was re-evaluated as the hierarchy was generated, but it has not been actively adopted for the human action segmentation. If we just arbitrarily take a level, it would lose some useful information. The Uniform Entropy Slice [22], [23] can help to select the supervoxel and flattening them into one level. It claims that it can obtain good performance results, but such methods do not work online and it will lose objects when processing long videos. Peng and Lo improved the [22] method to a streaming mode [24]. In this manner, we can obtain most important information from the video, but for human action recognition or understanding, there are still much ‘useless’ background information which do not help in the recognition process.

In the recent decade, automated action segmentation has received increasing attention. Ma et al. [25] provides a method to extract hierarchical spatial-temporal segments from videos, where the segments contain both parts and whole body of the human; their method can be used as a preliminary step for human action recognition, since they generate the silhouette of the human which can be regarded as the human action segmentation. Reference [26] defined the notion of actionness, the main idea of it is to use an ordinal random field model to distinguish different motions. Their work paved a new way on class-independent action analysis and video understanding. Lan et al. proposed a method [27] which join the action localization and recognition; the main idea is to represent the action with the figure-centric structural information which can improve the recognition accuracy. While their method recognizes the human action, the location which is treated as a latent variable will also be inferred. These represent work both on the segmentation and recognition or localization, very few researchers worked on the action segmentation, but the segmentation can actually improve the accuracies of the recognition and localization.

Recently, Lu et al. [28] proposed a method which focuses on the human action segmentation. They first proposed a new human motion saliency representation as a MRF (Markov random filed) model to segment the human action, the model is based on a hierarchy of supervoxels which is formed by the hierarchical graph-based video segmentation [21]. In their experiments, they showed that their results can improve the accuracy of the recognition and localization. Unlike their work which is only based on two levels from the hierarchy, our proposed method can work with arbitrary number of levels, so our hierarchical segmentation method can keep more detailed information of the video frames than theirs.

III. ACTION SEGMENTATION

Our action segmentation algorithm consists of several steps as follows.

1) We use the simple linear iterative clustering (SLIC) algorithm) [29] to obtain superpixels of each frame of the video, which will be the basic fundamental blocks.

2) A graph based hierarchical clustering algorithm will be applied to group the superpixels into a number of hierarchical clusters spatially and temporally.
The uniform entropy slice (UES) method will be applied to select an appropriate coarseness for each cluster from the hierarchy. The selected clusters will be the result of action segmentation.

A. APPEARANCE AND MOTION FEATURES

We begin by segmenting each video frame into superpixels. This serves as a way of reducing the amount of redundant information contained in the frame such that the same amount of information is contained in the superpixels. Given a video $M$, $M = \{I_i\}_{i=1}^{n}$ and $I_i \in \mathcal{R}^{W \times H}$, where $n$ is the number of video frames, $W$ and $H$ are the width and height of the frame respectively. The simple linear iterative clustering (SLIC) algorithm [29] groups pixels in an image based on their color similarity and proximity. A user determined parameter $K$ which controls the number of superpixels that will be segmented for each frame would need to be set before applying the algorithm [29]. Applying SLIC frame by frame independently, a video $M$ will be segmented into $N \cdot K$ superpixels as $S = \{s_{1,1}, s_{1,2}, \ldots, s_{1,j}, \ldots, s_{N,K} \}$, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, K$, where the $j$-th individual superpixel of the $i$-th frame is denoted by $s_{i,j}$. After the superpixels are obtained, their colors and associated motion as represented by optical flows can be used to calculate the appearance and motion features. In other words, each superpixel in a frame may be considered analogously to a pixel, its color and its relationship with neighbouring superpixels in the same frame can be used to extract appearance-based features, and the movements of the superpixel in between adjacent frames can be considered analogously to pixel movements in between adjacent frames to obtain motion-based features.

Motion discontinuities (boundaries) of superpixels exerts a strong influence on the action segmentation. To preserve motion boundaries and emphasize the boundaries of moving objects, we apply a state-of-the-art optical flow algorithm: Epicflow [30], with motion discontinuities detection which was originally designed for optical flow determinations of pixels in adjacent frames, with with some modifications, by decomposing the influence into two components: edge-based, and motion-based, to superpixels. The Epicflow algorithm [30] introduces dense matching by edge-preserving interpolation as a flow initialization step in a variational based optical flow estimation technique to preserve motion boundaries. For two adjacent video frames, a dense matching procedure [30], [31] is applied to compute an initial sparse set of matches. With this set of matches, a sparse-to-dense interpolation technique is employed by using an edge-preserving distance. An edge-preserving distance between two pixels is defined as the shortest geodesic distance of all possible paths between two pixels $p$ and $q$ that cross a given boundary as follows:

$$D(p, q) = \inf_{\Gamma \in \mathcal{P}_{pq}} \int_{\Gamma} C(p) dp$$  \hspace{1cm} (1)$$

where $\mathcal{P}_{pq}$ denotes the set of all possible paths between the two pixels $p$ and $q$, and $C$ the cost map denotes the cost of crossing pixel $p_s$. To focus the response on the boundaries of moving objects, we decompose our cost map $C$ into two components as follows:

$$C(p) = C_e(p) \cdot C_m(p)$$  \hspace{1cm} (2)$$

where $C_e$ is the edge map which is obtained by a state-of-the-art edge detector: the structured edge detector (SED) [32], and $C_m$ is the motion boundary which is obtained by a state-of-the-art motion boundary detector [33] using a random forest approach. The final optical flow estimation is calculated by using a variational energy minimization approach with the initial dense interpolation [30], [34].

Some results of our proposed motion boundary emphasized optical flow technique are shown in Fig. 1, the images on the left column show the original frames, those in the middle column show the results of Epicflow and those in the right column show the results of our proposed motion boundary emphasized optical flow. The results show that our proposed motion boundary emphasized optical flow obtains much more silhouette details on the human body than those obtained using the Epicflow method. This improved result can be attributed to the decomposition of the cost map into two components, edge-based and motion-based as indicated in Eq(2).
is the number of superpixels in the frame and the edge set $E$ represents the association between superpixels. For the graph weights $w(n, m)$, we follow [35] and compute the spatial neighbor connection weight $w_s(n, m)$ and temporal neighbor connection weight $w_t(n, m)$ for aggregating the spatial and temporal connections of superpixels.

Spatial connection edges represent the association between superpixels $n$ and $m$ in the frame $I_i$, for $i = 1, 2, \ldots, N$ and superpixels $n$ and $m$ should be in the same frame as $n, m \in V_i = \{s_i(1) \cdots K_i\}_i$. For the spatial connection, we first consider the association between each pair of neighboring superpixels $n$ and $m$, named these the first-order spatial edges, $n, m \in V_i$ and $n$ and $m$ are neighbors. The weight of first-order spatial edges $w_s(n, m)$ is defined as follows:

$$w_s(n, m) = \varphi_c d_c(n, m) + \varphi_g d_g(n, m) + \varphi_{hcol} d_{hcol}(n, m) + \varphi_{hflow} d_{hflow}(n, m)$$

(3)

where the terms $d(\cdot)$ are defined as follows:

1) $d_c(n, m) = \min(||c(n) - c(m)||, 30)$ is a robust threshold-distance between the color means [35], and the threshold is set to be 30;

2) $d_{hcol}(n, m) = D_x(h_{col}(n), h_{col}(m)))$ and $d_{hflow}(n, m) = D_x(h_{flow}(n), h_{flow}(m)))$ are the chi-squared distances between the color histograms and flow histograms [35] respectively;

3) $d_g(n, m)$ are the geodesic distances between the superpixels centroids using different cost maps [35];

4) The norms of the gradient of the flow and structured edge detector [32] are used to define the corresponding cost maps respectively [35];

5) $d_{bound}(n, m)$ is the motion boundary between $n$ and $m$ that is obtained by a state-of-the-art motion boundary detector [33] and $\varphi$s are the weights for the corresponding distances as defined in Eq (3).

For the computation to be robust to possibly small occlusions, we follow [35] to add second-order spatial edges $w_{s2}(n, m)$ to connect neighbors of neighbors. At first, edge weights $w_{s2}(n, m)$ are defined as follows:

$$w_{s2}(n, m) = \varphi_{hcol} d_{hcol}(n, m) + \varphi_{hflow} d_{hflow}(n, m)$$

(4)

$w_{s2}(n, m)$ use a constant $\varphi_{hcol}$ insert of geodesic distances $d_g$ to avoid the affect of object occlusions [35]. Additionally, to avoid spurious connections of second-order spatial connection, the edges will be dropped when $w_{s2}(n, k) + w_{s2}(k, m) \leq w_{s2}(n, m)$ [35]. Then second-order spatial edges are $w_{s2}(n, m)$ [35]:

$$w_{s2}(n, m) = \begin{cases} 0, & w_{s2}(n, k) + w_{s2}(k, m) \leq w_{s2}(n, m) \\ w_{s2}(n, m), & \text{otherwise} \end{cases}$$

(5)

For temporal connections, the association between superpixels $n \in V_i$ and $m \in V_j$ of neighboring frames $I_i$ and $I_j$, $j = i+1$ will be considered. We only consider the superpixels that matched in both the forward and backward temporal directions. In detail, a superpixel $n \in V_i$ in frame $I_i$, its best forward matching $\tilde{m} = \arg \max_{m \in V_j} \text{match}(n, m)$ in frame $I_j$ and its corresponding backward matching $\tilde{n} = \arg \max_{m \in V_i} \text{match}(n, \tilde{m})$ in frame $I_i$ are found, if they are the same, $n = \tilde{n}$, the temporal edge of $n$ and $m = \tilde{m}$, $w_{t}(n, m)$ is defined as follows [35]:

$$w_{t}(n, m) = \alpha_c d_c(n, m) + \alpha_{hcol} d_{hcol}(n, m) + \alpha_{hflow} d_{hflow}(n, m)$$

(6)

where $\alpha$’s are the weights of different cues. Here the match $(n, m)$ is based on the optical flow and pixel-wise color differences of the superpixels [35].

As summary, the weight $w(n, m)$ of the edge is computed by $w(n, m) = w_s(n, m) + w_t(n, m) = w_{s1}(n, m) + w_{s2}(n, m) + w_{t}(n, m)$.

2) GRAPH BASED HIERARCHICAL CLUSTERING

After the superpixel graph $G = (V, E, w)$ is constructed, we apply a hierarchical graph based clustering algorithm [36] to group the superpixels to hierarchical clusters. The algorithm starts from an over-segmentation of the video and merges regions in a greedy manner and recursively. Until all the regions are merged, the algorithm will produce a tree of regions in the form of hierarchical clusters as a result. In each iteration, the algorithm merges the most similar regions as the vertices of the graph $G_h = (V_h, E_h, w_h)$ to produce a newer graph $G_{h+1} = (V_{h+1}, E_{h+1}, w_{h+1})$ for the next iteration as described in [36]:

1) Select the edge with minimum weight (i.e., distance):

$$e^* = (n^*, m^*) = \arg \min_{(n, m) \in E_h} w_h(n, m)$$

2) Let $r = n^* \cup m^*$

3) Set $V_{h+1} = V_h \setminus \{n^*, m^*\} \cup \{r\}$ and $E_{h+1} = E_h \setminus \{e^*\}$

4) Compute weights $w_h + (n, m, \forall (n, m) \in E_{h+1})$

5) Stop when $E_{h+1} = \emptyset$

The resulting hierarchical tree structure is represented by $H = \{G_0, G_1, \ldots, G_h, \ldots\}$ and $G_h = (V_h, E_h, w_h), h = 1, 2, 3, \ldots, |H|$. The leaves of the tree $G_0$ are the initial elements of $V_0$, in our case, which is the superpixels of the video, and the root $G_{|H|}$ is the entire video. In Step 4 of the algorithm, the weight updating is needed to be defined for the algorithm. For the image segmentation problem, the average strength of the common boundary of the clusters is a good measure to update the weights [36]. The average strength weight is defined as follows:

$$w_{(h+1)}(n, m) = \frac{1}{|L_h(n, m)|} \sum_{(s, t) \in E_h(n, m)} w_h(s, t)$$

(7)

where, $L_h(n, m) = \{(s, t) | s \in n, t \in m, (s, t) \in E_h\}$. Note that the vertex $n \in V_h$ in a higher level $G_h$ is a cluster of vertices $n \subset V_j$ in lower level $G_j$, where $l < h$. $L_h(n, m)$ is
the set of edges connecting two clusters in $G_h$, $w_{h+1}^{(avg)}(n, m)$ is the distance between two clusters as the average edge weight along their common boundary.

In our action segmentation, we follow the idea as indicated in [35] and use a temporal penalty function $\varphi_{dis}$ for solving the problem that in some frames, different objects may be “accidentally” connected, and define the weight required in our proposed algorithm as follows:

$$w_h(n, m) = \frac{c(n, m) \cdot w_{h+1}^{(avg)}(n, m) + s(n, m) \cdot \varphi_{dis}}{c(n, m) + s(n, m)}$$ (8)

where $c(n, m)$ is the number of frames which are directly connected or connected with a second-order spatial link, and $s(n, m)$ denotes the number of frames where both are present but not connected [35]. If the two clusters are present, a real connection with weight or otherwise a virtual connection with weight is set as shown in Fig. 2.

During the clustering process, the cost of merging hierarchical clusters is a major influence, we set the penalty $\varphi_{dis}$ to be the value of the hardest merge [35].

In summary, the segmentation is obtained by merging based on the weight of the connected two superpixels and several pairwise features over the superpixels. If we set the number of clusters to be two, the results are shown in the bottom row in Fig. 3, we will obtain both the foreground and the background as the segmentation results.

3) TEMPORAL CONSTRAINT SUPERVOXEL TREE

The graph based hierarchical clustering algorithm introduced above merges the regions as demarcated by superpixels in the same frame one by one, the resulting structure is a deep structure and not all the levels will be used in further processing. By setting the numbers of desired clusters $C$, $C = \{c_1, c_h, \ldots, c_L\} \subseteq \{1, 2, \ldots, M\}$, $M = N \cdot K$, $c_i, c_j \in C$ and $c_i < c_j$ for $i < j$, $c_i \in C$ and $c_h \leq M$ for $h = 1, 2, \ldots, L$, as an increasing sequence of numbers, to indicate a hierarchical decomposition of the video with $L$ levels having $c_h$ regions in level $h$, where $N$ is the number of frames and $K$ is the (maximum) number of the superpixels in each frame. Based on the given number of clusters $C$, the video is decomposed hierarchically and is represented by a supervoxel tree $T$, $T = \{T_{c_1}, T_{c_2}, \ldots, T_{c_L}\} \subseteq H = \{G_M, G_{M-1}, \ldots, G_0\}$ and $T_h \in T$ and $T_h = G_{M-h} = (V_{M-h}, E_{M-h}, w_{M-h})$, the weighted graph $G_{M-h}$. For simplifying the notation of supervoxel tree $T$, we denote the graph of $T_h$ by $G'_h = G_{M-h}$ and $T_h = (V'_h, E'_h, w'_h)$ with $V'_h = V_{M-h}$, $E'_h = E_{M-h}$ and $w'_h = w_{M-h}$. The supervoxel tree is a top-down decomposition, as it moves from the entire video, and decompose it into a tree while the hierarchical clustering is a bottom-up construction as it moves from each frame and constructs the hierarchical set of clusters across a number of frames.

![FIGURE 3. Supervoxel comparison on videos in the UCF Sports dataset: video frames (top), the following are the results from the coarse level to finer levels. (a) Original frames. (b) Hierarchical frames.](image)

![FIGURE 4. The steps of generating a hierarchical video decomposition. The description of this decomposition is given in the text.](image)

In our action segmentation, we set the number of clusters: $C = \{1, 2, \ldots, L\}$. The root of the supervoxel tree $T$ represents the entire video, and the nodes in level $h$ of the hierarchy represent $h$ segments at that level $h$. The individual supervoxels at level $h$ is denoted by using subscripts as $v^h_i$. The level superscript $h$ for $v^h_i$ can be dropped and the individual supervoxel is denoted as $v_i$ whenever the level is irrelevant in the discussion. The hierarchical decomposition of video is represented by a tree structure as shown in Fig. 4.
The supervoxel tree constructed from root to leaves based on an assumption that it only considers the connection between every adjacent levels. Fig. 4 shows an example of video decomposition. The cuboids denote the video segmentation at each levels in a supervoxel hierarchy, $T_1$, $T_2$, $T_3$ and $T_4$ respectively. In the decomposition $T_3$ and $T_4$, the nodes $V_3^4$ and $V_4^2$ in $T_4$ are connected to $V_3^2$ in $T_3$. This connection is based on the similarities of their features and positions in the video as introduced in Section III-A. The locations of $V_3^4$ and $V_4^2$ are in the same area of $V_3^1$ in the video. In $T_2$ and $T_3$, and $T_2$ and $T_1$, the situations are similar. The corresponding tree in Fig. 4 is shown in Fig. 5, in the supervoxel tree, except for the root node and leaves, each intermediate node has one and only one parent, the root node only has child(ren), i.e., no incoming connections from elsewhere and the leaf node only has parent, i.e., no outgoing connections to elsewhere.

C. ACTION SEGMENTATION USING UNIFORM ENTROPY SLICE

After having the hierarchical supervoxel tree which contains the segmentation information of the video, the selection of the supervoxels from the hierarchy for video analysis such as analyzing the human in the video becomes complex. Flattening supervoxel hierarchies or unfolding the hierarchy by a uniform entropy slice method (UES) [22] can help to overcome the problem.

A tree slice is a set of nodes in a hierarchical tree such that on each root-to-leaf path in the hierarchy and only one node in the slice set as shown in Fig. 5. The set of all tree slices contains both useful and useless node selections. Useful is in the sense that they lead to good segmentation of the video.

Consider a binary variable $s_t$ for each node $v_t$, $v_t \in \bigcup_h V^h_t$ in the tree. The binary variable $s_t$ takes a value of 1 if node $v_t$ is a part of the slice and a value of 0 otherwise. Denote the full set of these over the entire tree as $x$. Any instance of $x$ induces a selection of nodes in the tree $T$, but not all instances are valid. In a valid slice, each root-to-leaf path in the segmentation tree $T$ has one and only one node being selected. In other words, to be a valid path, it can only contain one node being selected in this manner. We formulate this constraint linearly. Let $P$ denote an $R \times N$ binary matrix, where $R = |V^1_L|$ is the number of leaf nodes in $T$, and $N = |\bigcup_h V^h_L|$ is the total number of nodes in $T$. Each row of the matrix $P$ encodes a root-to-leaf path by setting the corresponding columns for the nodes on the path as 1 and 0 otherwise. Such a matrix enumerates all possible root-to-leaf paths in $T$. Much more details can be found in Table 1. The path which highlighted in yellow shows the path matrix in which each row specifies a root-to-leaf path through the tree in Fig. 5. By computing the path matrix and a constraint which will be introduced in the following section, we can obtain a valid tree slice.

The uniform entropy slice method balances the amount of information in the selected supervoxels for a given feature criterion. The feature criterion has a major influence on the information content. The information content of each node in the hierarchy can be computed by the entropy over a certain feature criterion, such as motion or humanness. Assume we have a feature criterion $F = (\cdot)$ that maps $v_t$ a node to a discrete distribution over the feature range and considering that we wish to segment the action of the human, in other words this is to drive the slice pays attention on regions of the video that are moving and contain the information of the human. We use optical flow and the product of the edge structure and the motion that computed over the video and a bivariate discrete distribution over a set of flow magnitudes and flow directions for $F_{flow}$. Similarly, $F_{edge}$ and $F_{motion}$ can be computed in a similar manner. We can obtain the selected supervoxel entropy as follows:

$$E(v_t) = -\sum_{\gamma} P_{F(v_t)}(\gamma) \log P_{F(v_t)}(\gamma)$$

and

$$F(v_t) = F_{flow}(v_t) + F_{edge}(v_t) + F_{motion}(v_t)$$

with $\gamma$ varying over the bivariate discrete feature range. In order to seek a tree slice that balances the overall information content of the selected nodes and consider the node information content, so the uniform entropy objective is proposed (see [22]). It seeks a valid tree slice that minimizes the difference in entropy of selected nodes and the minimization is over all valid tree slices.

$$\min \sum_{s} \alpha_s x_s + \delta \sum_{s,t} \beta_{s,t} x_s x_t$$

subject to $Px = 1_p$  

$$x = \{0, 1\}^N$$

where $1_p$ is an $p$-length vector of all ones. We can set $\alpha_s = |V^1_L|$ on the condition of $v_s \in V^1_L$ for each node $v_s$, $x_s$ will take on the value of 1 if the node is on the slice and 0 otherwise and so does the $x_t$. The quadratic term implements the uniform entropy objective [22]:

$$\beta_{s,t} = |E(v_s)| \cdot |E(v_t)| \cdot |E(v_s) - E(v_t)|$$

TABLE 1. The path matrix for Supervoxel granularity selection.

<table>
<thead>
<tr>
<th>$p_0$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
<th>$p_6$</th>
<th>$p_7$</th>
<th>$p_8$</th>
<th>$p_9$</th>
<th>$p_{10}$</th>
<th>$p_{11}$</th>
<th>$p_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

FIGURE 5. Supervoxel tree corresponding and one root-to-leaf path.
where \(|E(v_s)|\) and \(|E(v_t)|\) denote the volume of the supervoxels \(v_s\) and \(v_t\) respectively. Although nodes in the coarser levels of the tree have relatively higher entropy than nodes in the finer levels, the number of coarser level nodes is dramatically less than those in the finer levels. By adding the volume factors, we push the selection down the hierarchy unless uniform motion entropy has already been achieved. We use the standard mixed-integer programming solver (IBM CPLEX) to solve this minimization problem.

In Fig. 6, the top row shows the original frames; the bottom row shows the results of the hierarchical supervoxel selection which select from the middle frames which can be obtained by our hierarchical human action segmentation method. In the middle rows, we can observe that the frames of the low level are with many backgrounds, we can also say that they are noises. The frames of the high level are very clear, but the silhouette of human is in one label, this will lose too much information. In the bottom levels, we can observe that the results not only can keep the silhouette of the person, but also can keep the important regions inside the silhouette of the human.

**D. STREAMING ALGORITHM FOR ACTION SEGMENTATION**

We extend the UES into a streaming mode for solving useless segmentation results such as losing the objects in a long video. If we separate a long video into small video clips that can make the UES method work more efficiently but the consistency of the supervoxels between video clips will be lost. The reason of inconsistency is because of the next video clip does not have the information of its previous clip. So it lacks a constraint to link the two clips together as it does not know that the separation into two clips was an arbitrary action by the user to make the processing easier and more efficient.

Our streaming video segmentation is based on UES and hierarchical selection between video clips. Consistently, the algorithm we proposed not only can process the video clip by clip, but also can keep the label consistency between the clips; it is named as StreamUES [24]. We divide the original video into a set of smaller video clips, and segment them into a set of supervoxel trees as indicated in Section III-B, and represent them as \(T = \{T^{(1)}, T^{(2)}, \ldots, T^{(b)}, \ldots, T^{(B)}\}\), where \(T^{(b)} = \{T^{(b)}_h\}_{h=1}^H\) and \(T^{(b)}_h = (V^{(b)}_h, E^{(b)}_h, w^{(b)}_h), b\) is the index of the video clip and \(V^{(b)}_h\) is the supervoxel set at level \(h\) of clip \(b\). We use subscripts \(v^{h,(b)}\) to denote the individual supervoxel at level \(h\) of clip \(b\). The level superscript \(h\) for \(v_s^{h,(b)}\) can be dropped as \(v^{(b)}_s\) when the level is irrelevant in the discussion. \(|V^{(b)}_h|\) is the total number of supervoxels in clip \(b\) level \(h\).

For the streaming of video segmentation, the selection of the tree slice between video clips has to be consistent. To make the selection consistent, we formulate the streaming video segmentation by solving B minimization problems with supervoxel selection consistent constraints. In this manner, we can make sure that the current clip slice will contain the previous nodes if the nodes are the same. For video clip \(b\), the minimization problem is as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_s \alpha_s x_s^{(b)} + \delta \sum_{s,t} \beta_{st} x_s^{(b)} x_t^{(b)} \\
& \quad + \gamma \sum_u \sum_v (x_u^{(b)} - x_v^{(b-1)})^2 \cdot \mathbb{1}_{v_u^{(b)} = v_v^{(b-1)}} \\
\text{subject to} & \quad \mathbf{x}^{(b)} = \mathbf{1}_p \\
& \quad x^{(b)} = \{0, 1\}^N
\end{align*}
\]

where \(\alpha_s = |V^{(b)}_h|\) is the condition \(v^{h,(b)}_s \in V^{(b)}_h\) in level \(h\) of clip \(b\), for each node \(v^{h,(b)}_s\) will take 1 if the node is on the slice and 0 otherwise and so does the \(v^{h,(b-1)}_s\) is the node in the previous step and \(x^{(b-1)}\) is the solution of corresponding binary variable of the previous clips, the last term indicates the selection of the path between the two clips should be consistent, \(\delta\) and \(\gamma\) control the balance between these three terms. Our method begins to process the first clip of the video initially, it will not have the constraint and the constraint is null which is the same as the UES problem (10). After we obtained the results of the clip, we can obtain an optimal slice. In the second clip, there must be some nodes in the path which will appear in the second clip. By comparing the node set in the first clip with the node set in the second clip, the same nodes will be stored and they will also be marked in the second clip’s valid initial path. The initial path is used as an input to calculate the current clip’s finial path.

**IV. ACTION RECOGNITION**

The action feature extraction is the most important step as it can express the action characteristics of the video. With the results of action segmentation, the action characteristics can be represented based on the action segmentation of the video.
A. FEATURES EXTENDED BY ACTION SEQUENCE

To extract the action features, we run the action recognition based on our action segmentation results. Fig. 7 shows the idea of our action recognition method. For the reason that the edge of the human action has a strong inference for the action recognition, we dilate the action segmentation results to avoid missing the important feature points. In Fig. 7, in order to show our idea clearly, we combine the original video frames and the segmentation results. In this manner, we can obtain the silhouette of the human with little background edges.

We follow [37] and use the motion boundary trajectories which capture more information between moving objects; the initial feature points are based on the Harris corner condition:

$$\Gamma_{\text{corner}} = \varphi_1 \times \max \min \left( \lambda_1^{P_i}, \lambda_2^{P_i} \right)$$  \hspace{1cm} (13)

where \((\lambda_1^{P_i}, \lambda_2^{P_i})\) are the eigenvalues of the auto-correlation matrix of point \(P_i\) in frame \(I_t\). We then use a threshold \(\Gamma_{\text{corner}}\) to filter the motion boundary which is defined as follows:

$$\text{\widehat{H}}_{P_i} = \begin{cases} \| \nabla \mu_{P_i} \|^2 + \| \nabla \nu_{P_i} \|^2, & \min (\lambda_1^{P_i}, \lambda_2^{P_i}) \geq \Gamma_{\text{corner}} \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (14)

where \((\mu_{P_i}, \nu_{P_i})\) is the flow vector of point \(P_i\). Another threshold condition is used to determine if a point is a point of interest:

$$\Gamma_{\text{motion}} = \varphi_2 \times \max_{P_i \in I^t} \text{\widehat{H}}_{P_i} + \varphi_3$$  \hspace{1cm} (15)

The point \(P_i\) will be selected, if its magnitude is greater than the threshold, i.e., \(\text{\widehat{H}}_{P_i} > \Gamma_{\text{motion}}\). In our method, we set \(\varphi_1 = 0.0001, \varphi_2 = 0.001\) and \(\varphi_3 = 0.002\). From the above process, we can determine which subsampled point \(P_i\) will need to be considered in the initial feature points.

On the condition of the same number of feature points, we can obtain more useful points by only sampling in the human region as white circles show in Fig. 7. We follow [38] in our method, every feature point will track in each scale separately. Each point \(P_i = (x_i, y_i)\) in frame \(t\) is tracked to the next frame \(t+1\) by a median filter in a dense optical flow field \(\tilde{\omega}' = (\tilde{\mu}', \tilde{\nu}')\).

$$P_i^{t+1} = (x_i^{t+1}, y_i^{t+1}) = (x_i, y_i) + \tilde{\omega}'_{(\tilde{\mu}', \tilde{\nu}')}$$  \hspace{1cm} (16)

The trajectory: \((P_i, P_i^{t+1}, P_i^{t+2}, \ldots)\) is obtained by concatenating the points of posterior frames. The optical flow algorithm [39] is used to extract dense optical flows.

The shape of a trajectory can be described as a sequence \(S_t = (\Delta P_i, \ldots, \Delta P_i^{t+L+1})\) of displacement vectors \(\Delta P_i = (P_i^{t+1} - P_i) = (x_i^{t+1} - x_i, y_i^{t+1} - y_i)\) with the trajectory length of \(L = 15\) frames. The sum of the magnitudes of the displacement vectors is used to normalize the resulting vector:

$$S'_t = \frac{(\Delta P_i, \ldots, \Delta P_i^{t+L+1})}{\sum_{i=1}^{t+L+1} ||\Delta P_i||}$$  \hspace{1cm} (17)

where the vector \(S'_t\) is referred to as the trajectory descriptor.

Several descriptors are computed for every trajectory. These are:

1) HOG (Histogram of Oriented Gradient) along a trajectory focuses on the static part of the appearance of a local patch of the video.
2) HOF (histograms of optical flow) captures the local motion information based on the optical flow field and
3) MBH (motion boundary histograms) uses the gradient of the optical flow to cancel out most of the effects of camera motion.

The descriptors are calculated within a space-time volume around the trajectory to make use of the motion information in dense trajectories. The size of the volume is \(N \times N\) pixels and \(L\) frames. The volume is subdivided into a spatio-temporal grid of size \(n_\delta \times n_\delta \times n_t\) to embed structural information in the representation. We use \(N = 32, n_\delta = 2, n_t = 3\) for our experiments. The RootSIFT is used to normalize the descriptors. The maximum magnitude is calculated for each descriptor to filter the consistent descriptors. The final
dimensions of the descriptors are 30 for trajectories, 96 for HOG, 108 for HOF and 192 for MBH.

B. TRAINING OF CLASSIFIER MODEL

The bag of features and Fisher vector [40] are used to encode features. The bag of features was set similar to the one which was used in [38]. For every descriptor type, we use 100,000 randomly sampled features to train a codebook with \( k \)-means and the size of the codebook is set to 4,000.

The first and second order statistics between the video descriptors and a Gaussian Mixture Model (GMM) can be both encoded by the Fisher vector, this characteristics of the Fisher vector can improve the performance of action classification [41]. The encoding of the Fisher vector is different from the bag of feature, the first step is to use a factor of two using a Principal Component Analysis (PCA) [42] to reduce the descriptor dimensionality. To estimate the GMM, we randomly sample a subset of 256,000 features from the training set and set the number of Gaussians as \( \kappa = 256 \). After that, we can use a 2D \( \chi^2 \) dimensional Fisher Vector for each descriptor type, where \( D \) is the descriptor dimension after performing a PCA to represent each video. The power contained in the signal and the \( L_2 \) norm are used to normalize the Fisher vector. We then connect the normalized Fisher vectors of different descriptor types to combine different descriptor types.

For human action classification, we use a nonlinear support vector machine (SVM) with a multichannel \( \chi^2 \) kernel which can robustly combine channels [43]. The multichannel Gaussian kernel is defined as follows:

\[
K(H_i, H_j) = \exp(-\sum_{c \in CH} \frac{1}{A_c} D_{\chi^2}(H_i, H_j))
\]  

where \( H_i = \{h_{in}\} \) and \( H_j = \{h_{jn}\} \) are the histograms for channel \( c \) and \( D_{\chi^2}(H_i, H_j) \) is the \( \chi^2 \) distance between the videos \( H_i \) and \( H_j \) with respect to the \( c \) channel. This can be defined as:

\[
D_{\chi^2}(H_i, H_j) = \frac{1}{2} \sum_{n=1}^{Z} \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}}
\]  

with \( Z \) denotes the vocabulary size. The parameter \( A_c \) is the mean value of the distances between the training samples for the channel \( c \) [44]. For a given training set, the best set of channels \( CH \) can be obtained by a greedy method. In the case of multiclass classification, we use a one-against-all approach and select the class with the highest score.

C. DISCUSSIONS

In this section, we discuss the influence of the three descriptors on action recognition. With different descriptors, the accuracy of the action recognition will be different for their attribute focusing on different parts of the human being. The HOG has the minimum influence for it focuses on the static appearance information. The local motion information has a strong influence on human action recognition and HOF captures the local motion information, so the accuracy in using the HOF descriptor is better than that using a HOG descriptor. MBH is known for its robustness to camera motion by splitting the optical flow into horizontal and vertical components, and quantizes the derivatives of each component. It can encode the relative motion between pixels. So the MBH has the most influence on human action recognition accuracies. In order to avoid the disadvantages of these three descriptors, we combine them together and find that it can improve the recognition accuracy. These intuition are verified in our experiments.

V. EXPERIMENTS

A. Datasets

We conduct our experiments on some publicly available datasets.

- Dataset for Segmentation. We use two challenging datasets to evaluate our proposed action segmentation method. One is the DAVIS dataset [45] which contains 50 high quality, full HD video sequences. All video frames of the dataset are segmented with pixel-level annotations where the objects are separated from the background. The second dataset is the UCF Sports Action dataset which is provided in [46]. This dataset consists of a set of actions collected from various sports venues which are typically featured on broadcast television channels such as the BBC and ESPN. The video sequences were obtained from a wide range of stock footage websites including BBC Motion gallery, and Getty Images. This dataset contains 150 video sequences at a resolution of 720 \( \times \) 480. The dataset contains various number of videos which can be divided into 10 actions: diving, golf swinging, kicking, lifting, horse riding, running, skating, swing bench, swinging side, and walking. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints.

- Dataset for Recognition. For the evaluation of the proposed action recognition method, we use the UCF Sports Action dataset [46] and the KTH dataset [47]. The UCF Sports Action dataset is described above. The KTH dataset contains six human actions: walking, jogging, running, boxing, hand waving and hand clapping performed several times by 25 subjects in four different scenarios. The total number of sequences in the dataset is 2,391 and all the sequence are taken with homogeneous backgrounds with a static camera at the rate of 25 frames per second, down-sampled to the spatial resolution of 160 \( \times \) 120 pixels. The KTH dataset does not contain different background, so we use this dataset to evaluate our action recognition method.

B. PERFORMANCE MEASURE

For the measurement of the performance of the DAVIS dataset, we use the measures of (a) region similarity, (b) contour accuracy and (c) temporal stability to evaluate segmentation results as those performed in [45]. For the UCF-Sports dataset, we evaluate (a) the Intersection Over

16966

VOLUME 6, 2018
Union (IOU) value, (b) precision (IOU over the segmented area) and (c) the recall (IOU over ground truth area), similar to those conducted in [35].

The IOU of a human being contained in the ground truth \( v \) is given by the proposal \( p \) in the set of proposals \( P_v \) for \( v \) that maximizes the average overlap with the ground truth bounding boxes across all annotated frames:

\[
\text{IOU}(v) = \max_{p \in P_v} \frac{1}{|T_v|} \sum_{t \in T_v} \text{Overlap}(p_t, b^*_v) 
\]

(20)

where \( T_v \) is the set of frames for the human \( v \) with ground truth annotation, \( b^*_v \) denotes the bounding box of \( v \) in frame \( t \), and \( p_t \) is the region of the proposal occupied by the human in that frame.

We also use the confusion matrix to observe and analyze the performance of our proposed action recognition method. In the confusion matrix, each column represents the instances in the dataset (such as running, diving), while each row represents all possible predicted conditions in the dataset. The sum of each row of the confusion matrix is the total number of a category of videos, e.g., the total number of videos which are classified as “running”. The diagonal indicates the number of the videos that we predict successfully, i.e., if the video is classified as “running”, and our proposed method predicted it as “running”, then this is considered as one count of success; the number contained in the diagonal indicates the number of counts of successes. We can obtain the accuracy using:

\[
\text{ACC} = \frac{\sum \text{TP} + \sum \text{TN}}{\sum \text{Total Population}} 
\]

(21)

where the ‘TP’ denotes the true positive: the correct hit, ‘TN’ denotes the true negative which denotes the correct rejection. From the confusion matrix, divide the diagonal entry by the total number of videos in that category, e.g., ‘running’, we can obtain the prediction accuracy for that category of videos.

**TABLE 2.** Comparison of the results obtained using our method against those obtained using state-of-the-art methods (in %).

<table>
<thead>
<tr>
<th>Measure</th>
<th>[49]</th>
<th>[48]</th>
<th>[50]</th>
<th>[51]</th>
<th>[52]</th>
<th>[53]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>J Mean</td>
<td>57.1</td>
<td>64.1</td>
<td>54.3</td>
<td>56.9</td>
<td>51.4</td>
<td>30.1</td>
<td>42.6</td>
</tr>
<tr>
<td>J Recall</td>
<td>65.2</td>
<td>73.1</td>
<td>63.6</td>
<td>67.1</td>
<td>58.1</td>
<td>56.0</td>
<td>38.6</td>
</tr>
<tr>
<td>J Decay</td>
<td>4.4</td>
<td>8.6</td>
<td>2.8</td>
<td>7.5</td>
<td>12.7</td>
<td>5.0</td>
<td>8.4</td>
</tr>
<tr>
<td>F Mean</td>
<td>53.6</td>
<td>59.3</td>
<td>52.5</td>
<td>50.3</td>
<td>49.0</td>
<td>47.8</td>
<td>38.3</td>
</tr>
<tr>
<td>F Recall</td>
<td>57.9</td>
<td>65.8</td>
<td>61.3</td>
<td>53.4</td>
<td>57.8</td>
<td>51.9</td>
<td>26.4</td>
</tr>
<tr>
<td>F Decay</td>
<td>6.3</td>
<td>8.6</td>
<td>5.7</td>
<td>7.9</td>
<td>13.8</td>
<td>6.6</td>
<td>7.2</td>
</tr>
<tr>
<td>T</td>
<td>29.3</td>
<td>36.6</td>
<td>26.3</td>
<td>21.0</td>
<td>25.6</td>
<td>34.5</td>
<td>61.6</td>
</tr>
</tbody>
</table>

**C. EVALUATION OF THE ACTION SEGMENTATION RESULTS**

The experimental results show that our method not only can work on the human action, but also can work on the objects which are moving. We evaluate our method on the DA VIS dataset. Table 2 summarizes the results for a representative subset of the DA VIS sequences and the average performance over the whole dataset. One of the advantages of our is that it focuses on the boundaries, so on the region similarity measure, our method does not work as well as [48], but on the other areas, our results are better.

For the UCF-Sports dataset, we evaluate our human action segmentation method with the IOU, for it reflects the coverage with the ground truth. Table 3 summarizes the mean IOU, the mean precision and the mean recall results. These demonstrate that our proposed method can obtain \( \approx 10% \) better results than those obtained in [28]. For the IOU measure, our precision results are a little higher than those obtained and reported in [28], while for the recall, our method is \( \approx 10% \) higher than those obtained and reported in [28]. The score shows that our method can work very efficiently when compared with those of other state-of-the-art methods [28]. Figures 9 to 12 show our proposed method can work very effectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>IOU</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF-Sport</td>
<td>47.6</td>
<td>56.1</td>
<td>73.8</td>
</tr>
<tr>
<td>Ours</td>
<td>76.5</td>
<td>60.6</td>
<td>71.0</td>
</tr>
</tbody>
</table>

**D. EVALUATION OF ACTION RECOGNITION**

In action recognition, we evaluate our recognition method on the UCF-Sports and KTH datasets. In all experiments we fix \( C = 100 \) for the SVM, for it has been shown, the details of which are not shown here, that this can give good results when validating on a subset of the training samples. In the following, we use the Fisher vector encoding technique since the results of applying this give an improved performance than when it was not applied. In the case of multiclass classification, we use a one-against-all approach and select the class.
TABLE 4. Action localization results measured using the average IOU (in %) on the UCF sports dataset.

<table>
<thead>
<tr>
<th>Action</th>
<th>Diving</th>
<th>Golf</th>
<th>Kick</th>
<th>Lift</th>
<th>Ride</th>
<th>Run</th>
<th>Skate</th>
<th>Swing-b</th>
<th>Swing-s</th>
<th>Walk</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[28]</td>
<td>48.3</td>
<td>50.0</td>
<td>35.5</td>
<td>57.1</td>
<td>29.8</td>
<td>33.7</td>
<td>45.9</td>
<td>62.3</td>
<td>34.9</td>
<td>39.0</td>
<td>48.0</td>
</tr>
<tr>
<td>[25]</td>
<td>44.3</td>
<td>50.5</td>
<td>48.3</td>
<td>51.4</td>
<td>30.6</td>
<td>33.1</td>
<td>38.5</td>
<td>54.3</td>
<td>20.6</td>
<td>39.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Ours</td>
<td>49.1</td>
<td>62.0</td>
<td>58.7</td>
<td>60.8</td>
<td>28.2</td>
<td>55.8</td>
<td>47.8</td>
<td>75.1</td>
<td>57.9</td>
<td>51.0</td>
<td>56.1</td>
</tr>
</tbody>
</table>

FIGURE 8. The accuracy of action recognition results for the UCF-Sports dataset on the segmentation results and the comparison with the state-of-the-art methods.

FIGURE 9. The result of human action of diving.

FIGURE 10. The result of human action of swing bench.

FIGURE 11. The result of human action of swing bench.

FIGURE 12. The result of human action of walking.

with the highest score. For every type of the descriptors, we carry out five experiments to obtain the average results. In our results, we show the mean accuracy, the maximum accuracy and the minimum accuracy.

We evaluate our proposed recognition method on the UCF Sports dataset. Fig. 8 shows the accuracies as obtained using our method with different trajectory features and the accuracies of the state-of-the-art methods; for some state-of-the-art methods [12], [13], [19], [25] did not provide results on the maximum and minimum accuracies, we set them to be the same as the mean accuracy to show the bar clearly. In Fig. 8, the left six bars are the recognition result of the original dataset, comparing with the state-of-the-art methods (right five bars), it shows that the minimum accuracy obtained using our method is higher than all the state-of-the-art methods; the maximum accuracy obtained using our method is almost 10% higher than those obtained in [12], [13], and [25].

For the recognition of the UCF Sports dataset, we consider the situation: if we recognize the dataset using the action segmentation method, it will lose some information with the determination of the boundaries. Based on this intuition, we combine the results which were obtained using the segmentation results and the original video together. The accuracies are obtained as shown in Figure 8. The feature name contains ‘n’ and ‘n’ means the feature from different video data (one is the original video; another is the video with the segmentation results.). As Fig. 8 shows, we can improve the accuracy of the action segmentation, the highest accuracy
The confusion matrix of the proposed method on the UCF sports dataset (in %).

<table>
<thead>
<tr>
<th></th>
<th>Diving</th>
<th>Golf</th>
<th>Kick</th>
<th>Lift</th>
<th>Ride</th>
<th>Run</th>
<th>Skate</th>
<th>Swing-b</th>
<th>Swing-s</th>
<th>Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diving</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Golf</td>
<td>0.0</td>
<td>83.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.7</td>
</tr>
<tr>
<td>Kick</td>
<td>0.0</td>
<td>0.0</td>
<td>83.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.7</td>
</tr>
<tr>
<td>Lift</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Ride</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Run</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Skate</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>50.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Swing-b</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Swing-s</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Walk</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

The accuracy of action recognition result based on the KTH on the condition of segmentation result comparing with the state of the art methods.

The confusion matrix of the proposed method on the KTH dataset (in %).

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Jogging</th>
<th>Hand-waving</th>
<th>Hand-clapping</th>
<th>Boxing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Running</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Jogging</td>
<td>0.0</td>
<td>0.0</td>
<td>94.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Hand-waving</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Hand-clapping</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.0</td>
<td>89.0</td>
</tr>
<tr>
<td>Boxing</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

For the KTH dataset, similar to the UCF Sports dataset, the results are shown in Fig. 13, and these are compared against the accuracies of some of the state-of-the-art methods, we find that the accuracies obtained using our proposed method are the highest; the highest accuracy is 97.22% and the corresponding confusion matrix is shown in Table 6. As the recognition accuracy of the KTH dataset is very high, further improvement is a very hard.\(^1\)

Based on the performance evaluations, the accuracies obtained for the UCF Sports dataset are higher than other methods, with an improvement in the range of 3%—10%. For the KTH dataset, the accuracies obtained using our method have exceeded 97% which is better than those obtained using other methods such as 95% [38]. The main reason is that our method can capture more feature points on the human and the background of the KTH dataset is simple, monotonous and uniform, so our segmentation results cannot help much to improve the recognition accuracy. From the confusion matrix shown in Table 6, we find the main error occurred in ‘Hand clapping’ as this action is very similar to ‘Hand waving’. The UCF Sports dataset is different to the KTH dataset, the background of the UCF Sports dataset is related to respective action category.

To fully evaluate our method, we also report the results of the ‘actionness’ for it can reflect the performance of the action recognition. We measure the mean average precision (mAP) of the actionness ranking of our method, compare them with those obtained in [26] which generate foreground motion saliency map and [28] which generate actionness ranking of motion saliency map and human motion saliency map, we obtain the results as shown in Table 7. In Table 7, it shows the quantitative comparisons of our method against baseline methods. We observe the human action segmentation method shows a significant improvement of almost 20% mAP gain over a state-of-the-art method [26] and almost 10% mAP gain over another state-of-the-art method [28].

VI. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

In this paper, we introduced an online approach to human action segmentation in videos. Our approach consists of two stages: In the first stage, we adopt a clustering approach to image segmentation in a temporal fashion, where we obtain hierarchical supervoxel levels, which will be flattened.

\(^1\)In this case, We find that combining the segmentation results with the original video does not lead to improved results.
into one level using an online video segmentation approach, StreamUES, that was proposed in [24]. The flattened level preserves information about human structure information in the video. In the second stage, we combine the information on the human structure, as well as the original video frames, so as to “strengthen” the human action in a video. We evaluate our approach on a number of datasets, including the DAVIS dataset [45], the UCFSports dataset [46] and the KTH dataset [47]; and achieve promising results in terms of action segmentation and recognition.

In future work, it would be interesting to consider ways to improve on the supervoxel selection in the action segmentation stage by adopting more feature criteria, such as object-ness [55] and others [56].

The methods proposed in this paper pertain to using feature extraction techniques using specially designed techniques. In light of the great successes in using parametric models based on convolutional neural networks, and deep learning neural networks (multilayer perceptrons) to image processing [57], an interesting direction for future research would be to consider ways in which the predomnately 2D convolutional neural network applied to image processing could be extended to three dimensional, to handle temporal aspects associated with a video. If this can be achieved it may alleviate the complexity with which we need to go into in clustering the superpixels both spatially and temporally, and the large number of features associated with the dense sampling approach in the action recognition as indicated in this paper.

REFERENCES


CHENG PENG received the B.Sc. degree from Beijing Normal University, Zhuhai, in 2013, and the M.Sc. degree from the Macau University of Science and Technology in 2015, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and artificial intelligence.

SIO-LONG LO received the B.Sc., M.Sc., and Ph.D. degrees from the Macau University of Science and Technology (MUST), in 2007, 2009, and 2012, respectively. From 2012 to 2013, he was a Post-Doctoral Researcher with MUST, where he is currently an Assistant Professor with the Faculty of Information Technology. His research interests include computer vision, machine learning, and artificial intelligence.

JIE HUANG received the B.E. degree from the South China Institute of Software Engineering, Guangzhou University, in 2013, and the M.Sc. degree from the Macau University of Science and Technology in 2015, where she is currently pursuing the Ph.D. degree. Her current research interests are image retrieval and geo-localization of images.

AH CHUNG TSOI received the M.Sc. and Ph.D. degrees from the University of Salford, England. He was with a number of institutions, and became a Professor with the Department of Electrical Engineering, University of Queensland, in 1992, the Dean of the Faculty of Informatics, and then the inaugural Pro Vice-Chancellor (Information Technology and Communications), University of Wollongong, from 1996 to 2003, an Executive Director, Mathematics, Information and Communication Sciences Inter-Disciplinary Cluster, Australian Research Council, from 2003 to 2005, the Director of the Monash eResearch Centre, Monash University, from 2005 to 2007, the Vice President (Research and Institutional Advancement) with Hong Kong Baptist University, from 2007 to 2010, the Dean and a Professor with the Faculty of Information Technology, Macau University of Science and Technology, from 2010 to 2016. He is currently an Emeritus Professor with the University of Wollongong. His research interests include machine learning, neural networks, artificial intelligence, and their applications to practical problems.

* * *