Action Recognition and Localization by Hierarchical Space-Time Segments

Shugao Ma [1] Jianming Zhang [1] Nazli Ikizler-Cinbis [2] Stan Sclaroff [1]
[1] Department of Computer Science, Boston University
[2] Department of Computer Engineering, Hacettepe University
{shugaoma, jmzhang, sclaroff}@bu.edu nazli@cs.hacettepe.edu.tr

Abstract

We propose Hierarchical Space-Time Segments as a new representation for action recognition and localization. This representation has a two level hierarchy. The first level comprises the root space-time segments that may contain a human body. The second level comprises multi-grained space-time segments that contain parts of the root. We present an unsupervised method to generate this representation from video, which extracts both static and non-static relevant space-time segments, and also preserves their hierarchical and temporal relationships. Using simple linear SVM on the resultant bag of hierarchical space-time segments representation, we attain better than, or comparable to, state-of-art action recognition performance on two challenging benchmark datasets and at the same time produce good action localization results.

1. Introduction

Human action recognition is an important topic of interest, due to its wide ranging application in automatic video analysis, video retrieval and more. Many local space-time representations have been proposed for use in the action recognition task. Among them, space-time interest points (STIPs) [13] and dense trajectories [22] are perhaps the most widely used. One major issue for both STIPs and dense trajectories is that they focus on non-static parts of the video, while the static parts are largely discarded. We argue that both non-static *and* relevant static parts in the video are important for action recognition and localization. There are at least two reasons:

- Some static parts of the space-time video volume can be helpful in recognizing human actions. For example, for the *golf swing* action, instead of just relying on the regions that cover the hands and arms, which have significant motion, the overall body pose can also indicate important information that may be exploited to better discriminate this action from others.
- In many applications, estimating the location of the action performer is also desired in addition to recognizing the action. Extracting only the non-static body parts may not lead to accurate localization of the whole body. Therefore, accounting for both static and non-static parts may help.

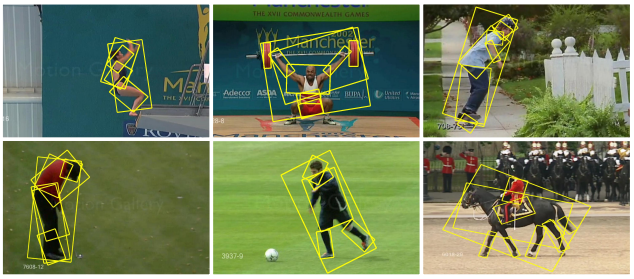

Figure 1. Extracted segments from example video frames of the UCF Sports dataset. Yellow boxes outline the segments. Boxes within a box indicate child-parent relationships.

In this paper, we propose a representation that we call *hierarchical space-time segments* for both action recognition and localization. In this representation, the space-time segments of videos are organized in a two-level hierarchy. The first level comprises the root space-time segments that may contain the whole human body. The second level comprises space-time segments that contain parts of the root.

We present an algorithm to extract hierarchical space-time segments from videos. This algorithm is unsupervised, such that it does not need any pre-trained body or body part detectors that may be constrained by strong priors of common poses present in the related training set. Fig. 1 shows some example video frames and extracted hierarchical segments in the UCF-Sports video dataset [19] and more examples are shown in Fig. 5. The representation of parts is multi-grained in that the parts are allowed to overlap: some parts are actually parts of larger parts, e.g. lower leg and whole leg. These segments are then tracked in time to produce space-time segments as shown in Fig. 2.

Our algorithm comprises three major steps. We first apply hierarchical segmentation on each video frame to get a set of segment trees, each of which is considered as a candidate segment tree of the human body. In the second

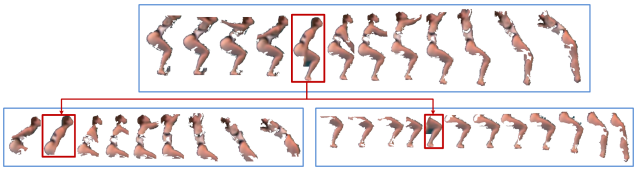

Figure 2. Hierarchical space-time segments extracted from a *diving* action in the UCF Sports dataset. Each blue box shows a space-time segment. Red boxes show a segment tree on a frame, and the space-time segments are produced by tracking these segments.

step, we prune the candidates by exploring several cues such as shape, motion, articulated objects' structure and global foreground color. Finally, we track each segment of the remaining segment trees in time both forward and backward. This process yields the final hierarchical space-time segments. These space-time segments are subsequently grouped into tracks according to their space-time overlap.

We then utilize these space-time segments in computing a bag-of-words representation. Bag-of-words representations have shown promising results for action recognition [13, 22]. Those representations, however, mostly lack the spatial and temporal relationships between regions of interest, whereas there are attempts to include these relationships later via higher order statistics [24, 11, 7, 18, 6]. Our hierarchical segmentation-based representation preserves hierarchical relationships naturally during extraction and, by following the temporal continuity of these space-time segments, the temporal relationships are also preserved. We show in experiments that by using both parts and root space-time segments together, better recognition is achieved. Leveraging temporal relationships among root space-time segments, we can also localize the whole track of the action by identifying a sparse set of space-time segments.

To summarize, the main contributions of this paper are:

1. A new *hierarchical space-time segments* representation designed for both action recognition and localization that incorporates multi-grained representation of the parts and the whole body in a hierarchical way.
2. An algorithm to extract the proposed hierarchical space-time segments that preserves both static *and* non-static relevant space time segments as well as their hierarchical and temporal relationships. Such relationships serve for better recognition and localization.

We evaluated the proposed formulation on challenging benchmark datasets UCF-Sports: [19] and HighFive [17]. These datasets are representative of two major categories of realistic actions, namely sports and daily interactions. Using just a simple linear SVM on the *bag of hiearchical space-time segments* representation, better or comparable to state-of-the-art action recognition performance is achieved without using human bounding box annotations. At the same time, as the results demonstrate, our proposed representation produces good action localization results.

2. Related Work

In recent years, action recognition methods that use bag of space-time interest points [13] or dense trajectories [22] have performed well on many benchmark datasets. However, one issue is that the space-time relationships among the STIP or dense trajectories are not explicitly extracted. Many attempts have been made to explore such relationships for action recognition, which usually resort to higher order statistics of the already extracted STIPs or dense trajectories, such as pairs [24, 11], groups [7], point clouds [3], or clusters [18, 6]. In contrast, the extraction of both space-time segments and their hierarchical and temporal relationships are integrated in our approach.

Action localization is usually done in the action detection setting [20, 21] and relatively few works do both action recognition and localization. Action recognition methods that use holistic representations of the human figure may have the potential to localize the action performer, such as motion history images [2], space-time shape models [8] and human silhouettes [23]. But these approaches may not be robust enough to handle occlusions and cluttered backgrounds in realistic videos. Works that use pre-trained human body or body part detectors also can localize the performer, such as [10, 25, 26]. However, their detectors may be constrained by the human body appearance priors implicitly contained in the training set and may not be flexible enough to deal with varying occlusions and poses in various actions. In [12] the bag of STIP approach was extended beyond action recognition to localization using latent SVM. In this paper, we show that by using hierarchical space-time segments we can do action localization within the bag-of-words framework. We show much better localization performance than [12] in experiments (Table 3).

Our work is also related to recent works in video segmentation. Recent works on hierarchical video segmentation include [9, 5]. The method in [4] applies a general video segmentation method to produce video sub-volumes for action recognition. However, for action recognition and localization, general video segmentation methods may produce much more irrelevant space-time segments than our method that explores human action related cues to effectively prune irrelevant ones. Some work proposed *object centric* video segmentation [14, 16], but these methods do not extract space-time segments of parts.

3. Hierarchical Space-Time Segments

In this section, we describe the major steps of our algorithm for extracting hierarchical space-time segments.

3.1. Video Frame Hierarchical Segmentation

For human action recognition, segments in a video frame that contain motion are useful as they may belong to moving body parts. However, some static segments may belong to

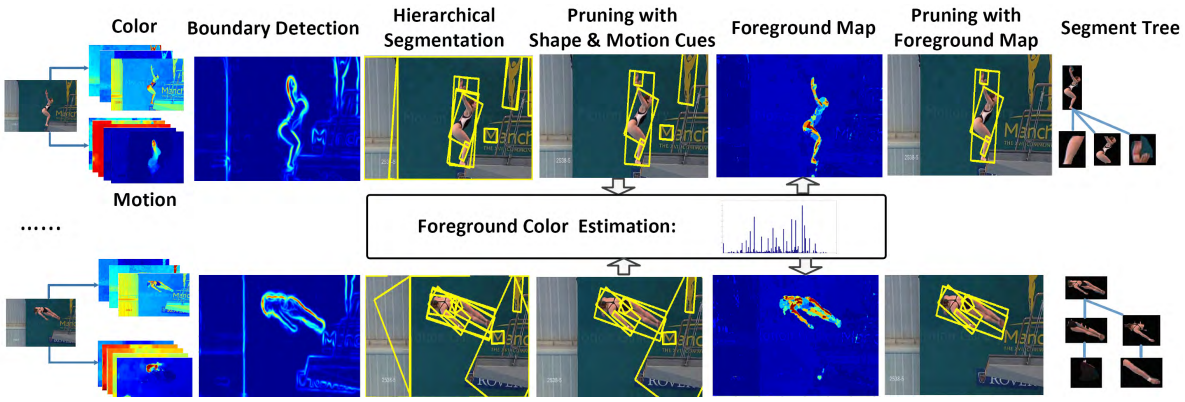

Figure 3. The pipeline for hierarchical video frame segments extraction.

the static body parts, and thus may be useful for the pose information they contain. Moreover, for localizing the action performer, both static and non-static segments of the human body are needed. Based on this observation, we design our video frame segmentation method to preserve segments of the whole body and the parts while suppressing the background. The idea is to use both color and motion information to reduce boundaries within the background and rigid objects and strengthen internal motion boundaries of human body resulting from different motions of body parts. Then, a subsequent hierarchical segmentation may further reduce irrelevant segments of background and rigid objects while retaining dense multi-grained segments on the human body.

In practice, on each video frame, we compute the boundary map by the method in [15] using three color channels and five motion channels including optical flow, unit normalized optical flow and the optical flow magnitude. The boundary map is then used to compute an Ultrametic Contour Map (UCM) by the method in [1].

The UCM represents a hierarchical segmentation of a video frame [1], in which the root is the whole video frame. We traverse this segment tree to remove redundant segments as well as segments that are too large or too small and unlikely to be a human body or body parts. Specifically, at each segment, if its size is larger than $\frac{2}{3}$ of its parent size, or is larger or smaller than some thresholds (parameters of the system), the parent of its children segments is set to its parent segment and it is then removed.

We then remove the root of the segment tree and get a set of segment trees $\mathcal{T}^t$ (t is index of the frame). Each $T_j^t \in \mathcal{T}^t$ is considered as a candidate segment tree of a human body and we denote $T_j^t = \{s_{ij}^t\}$ where each s_{ij}^t is a segment and s_{0j}^t is the root segment. Two example candidate segment trees, which remained after the subsequent pruning process, are shown in the rightmost images in Fig. 3.

3.2. Pruning Candidate Segment Trees

We want to extract both static and non-static relevant segments, so the pruning should preserve segments that are static but relevant. We achieve this by exploring the hierarchical relationships among the segments so that the decision to prune a segment is not made using only the local information contained in the segment itself, but the information of all segments of the same candidate segment tree. This is in contrast with STIPs or dense trajectories in which each local space-time representation is extracted independently.

Specifically, all subsequent pruning is performed at the candidate level, i.e. a candidate segment tree is either removed altogether or kept with all its segments. In this way, we may extract the whole human body even if only a small body part has motion. Example cases can be seen in the output segment trees in Fig. 1 and Fig. 5. Especially in the *golf* action, there are only slight motions at the hands of the human bodies, but we can still correctly extract the whole human body and the static body parts.

We explore multiple action related cues to prune the candidate segment trees, described in the order of our pipeline (Fig. 3) as follows:

Tree pruning with shape and color cues. The segment trees are first pruned using the shape and motion cues.

- Shape cue: Background objects (*e.g.* buildings) with straight boundaries are common in manmade scenes, but human body boundaries contain fewer points of zero curvature. With this observation, for each candidate $T_j^t \in \mathcal{T}^t$, we compute the curvature at all boundary points of its root segment s_{0j}^t. If the ratio of points that have approximately zero curvature is large (> 0.6 in our system), we remove T_j^t. The curvature κ_a at a boundary point (x_a, y_a) is computed as follows:

$$\kappa_a = \left| \frac{x_a - x_{a+\delta}}{y_a - y_{a+\delta}} - \frac{x_{a-\delta} - x_a}{y_{a-\delta} - y_a} \right| \quad (1)$$

  where $(x_{a-\delta}, y_{a-\delta})$ and $(x_{a+\delta}, y_{a+\delta})$ are two nearby points of (x_a, y_a) on the segment boundary.

- Motion cue: For each segment $s_{ij}^t \in T_j^t$, we compute the average motion magnitude of all pixels within s_{ij}^t. To compute the actual motion magnitude, we compute

an affine transformation matrix between the current frame and the consecutive frame to approximate camera motion and calibrate the flow fields accordingly. If at least one segment has average motion magnitude higher than some threshold, T_j^t will be kept, otherwise it will be pruned. The threshold is estimated based on the motion magnitudes of pixels that are not contained in any of $T_j^t \in \mathcal{T}^t$ since these pixels are likely to belong to the background.

Tree pruning using foreground map We explore two general assumptions to estimate foreground maps for further pruning of the candidates. First, as an articulated object, the region of a non-static human body usually contains many internal motion boundaries resulting from different motions of body parts. Thus, the segment tree of the human body usually has a deeper structure with more nodes compared to that of the rigid objects. We account for this as a *structure cue*. Second, segments of the foreground human body are more consistently present than segments caused by artificial edges and erroneous segmentation, and we account for this as a *global color cue* over the whole video sequence. The foreground maps are then constructed as follows:

Denote $\widetilde{\mathcal{T}}^t$ as the set of remaining candidate trees after pruning with shape and motion cues, and let S represent the set of all remaining segments on all frames, i.e. $S = \{s_{ij}^t | \forall\ s_{ij}^t \in T_j^t,\ \forall\ T_j^t \in \widetilde{\mathcal{T}}^t,\ \forall\ t\}$. To avoid cluttered notation, in the following we simply denote $S = \{s_k\}$. We compute the L_∞ normalized color histogram $\mathbf{c}_k$ for every $s_k \in S$ (128 bins in our system). Then the foreground color histogram $\mathbf{c}$ is voted by all segments:

$$\mathbf{c} = \sum_{s_k \in S} 2^{h_k} \cdot \mathbf{c}_k \tag{2}$$

where h_k is the height of segment s_k in its segment tree. For root segment s_{0j}^t we define its height h_{0j}^t to be 1 and for a non-root segment s_{ij}^t its height h_{ij}^t is set to the number of edges on the path from the root to it plus one. The color histogram $\mathbf{c}$ is then $L1$ normalized. As we can see from Eq. 2, colors of more frequently appearing segments and segments with greater heights will receive more votes.

Let F^t denote the foreground map of frame t. Its value at ith pixel is set as $F_i^t = \mathbf{c}(c_i^t)$, where c_i^t is the color of ith pixel of frame t. For each segment s_{ij}^t in a candidate segment tree $T_j^t \in \widetilde{\mathcal{T}}^t$ on frame t, we compute its foreground probability as the average values covered by s_{ij}^t in F^t. If all segments of T_j^t have low foreground probability, T_j^t is pruned, otherwise it will be kept. One can see in Fig. 3 that by using the foreground map, we can effectively prune the background segments in the example frames.

3.3. Extracting Hierarchical Space-Time Segments

After candidate segment tree pruning, we extract a set $\widehat{\mathcal{T}}^t$ that contains remaining candidate segment trees. To capture temporal dynamics of the human body or body parts, for each $T_j^t \in \widehat{\mathcal{T}}^t$ we track every segment $s_{ij}^t \in T_j^t$ to construct a space-time segment.

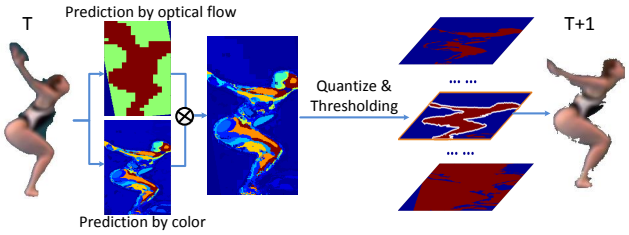

Figure 4. Our non-regid region tracking method.

In this work, we propose a novel method for non-rigid region tracking (Fig. 4). Let R denote the tracked region on the current frame. Let $a = (x_a,\ y_a)$ denote the coordinates of a point in R and let $(\Delta x_a,\ \Delta y_a)$ denote its corresponding flow vector from median filtered optical flow fields. The predicted region at the next frame by flow is then $R' = \{(x'_a,\ y'_a)\} = \{(x_a{+}\Delta x_a,\ y_a{+}\Delta y_a)\}$. Let B denote the bounding box of R' whose edges are parallel to horizontal and vertical axes and let $\hat{B}$ represent the tight bounding rectangle of R' whose longer edge is parallel with R''s axis of least inertia. Let $\mathbf{h}$ represent the color histogram of the original segment being tracked . We then compute a flow prediction map M_f and color prediction map M_c over B. Suppose a point $b' \in B$ on the next frame has color $c_{b'}$, then we set $M_c(b') = \mathbf{h}(c_{b'})$ and set $M_f(b')$ as:

$$M_f(b') = \begin{cases} 2 & b' \in R' \\ 1 & b' \in \hat{B} \wedge b' \notin R' \\ 0 & otherwise \end{cases} \tag{3}$$

We combine the two maps, $M(b') = M_f(b') \cdot M_c(b')$. In practice, when we compute M_f we use a grid over $\hat{B}$ so that points in the same cell will be set to the maximum value in that cell. This is to reduce holes caused by noise in the optical flow field.

The map M is scaled and quantized to contain integer values in the range $[0, 20]$. By applying a set of thresholds δ_m of integer values from 1 to 20, we get 20 binary maps. The size of every connected component in these binary maps is computed and the one with most similar size to R is selected as the candidate. Note that multiple thresholds are necessary. This is because the color distribution of the template may be either peaked, if the template has uniform color, or relatively flat, if the template contains many colors. Thus the range of values in $M(b')$ may vary in different situations and a fixed threshold will not work well. The number of thresholds is experimentally chosen for the trade-off between performance and speed.

If the ratio of size between the selected candidate and R being tracked is within a reasonable range (we use $[0.7, 1.3]$), we set the candidate as the tracked region, oth-

erwise the target is considered as being lost. Since all the computations are performed locally in B and implemented as matrix operations utilizing efficient linear algebra software packages, this tracking method is very fast. We track each region at most 7 frames backward, where backward optical flow is used, and 7 frames forward. In this respect, our space-time segments have a relatively short extent, with a maximum temporal length of 15.

Although segment tree T_j^t may have deep structures with height larger than 3, these structures may not persistently occur in other video frames due to the change in human motion and the errors in segmentation. For robustness and simplicity, we have only two levels in the resultant hierarchical structure of space-time segments: the root level is the space-time segment by tracking the root segment s_{0j}^t of T_j^t, and the part level are the space-time segments by tracking non-root segments $s_{ij}^t, \forall i \neq 0$.

Since space-time segments are constructed using segment trees of every video frame, many of them may temporally overlap and contain the same object, *e.g.* space-time segments produced by tracking the segments that contain the same human body but are from two consecutive frames. This provides a temporally dense representation of the video. We explore this dense representation to construct longer tracks of objects despite the short temporal extent of individual space-time segment. Specifically, two root space-time segments that have significant spatial overlap on any frame are grouped into the same track. The spatial overlap is measured by the ratio of the intersection over the area of the smaller segment and we empirically set the overlap threshold to be 0.7. The part space-time segments are subsequently grouped into the same track as their roots. Note that now we have temporal relationships among root space-time segments of the same track.

For each track, we then compute bounding boxes on all frame it spans. As described before, many root space-time segments of a track may temporally overlap on some frames. On each of those frames, every overlapping root space-time segment will provide a candidate bounding box. We choose the bounding box that has the largest average foreground probability using the foreground map described in section 3.2. As we assume the foreground human body is relatively consistently present in the video, we further prune irrelevant space-time segments by removing tracks of short temporal extent. We set the temporal length threshold to be one fourth of the video length but keep the longest track if no track has length greater than the threshold.

4. Action Recognition and Localization

To better illustrate the effectiveness of hierarchical space-time segments for action recognition and localization, we use simple learning techniques to train action classifiers and the learned models are then used for action localization.

For each space-time segment, we divide its axis parallel bounding boxes using a space-time grid, compute features within each space-time cell and concatenate the features from all cells to make the final feature vector. Here we want to mention that we do not limit the space-time segments to have the same length as the method in [22] did for dense trajectories. This is to deal with the variations of action speeds of different performers. We do not use space-time segments that are too short (length < 6) as they may not be discriminative enough. We also split long space-time segments (length > 12) into two to produce shorter ones while keeping the original one. This is to get a denser representation of the action.

We build separate codebooks for root and part space-time segments using k-means clustering. Subsequently each test video is encoded in the BoW (bag of words) representation using max pooling over the similarity values between its space-time segments and the code words, where the similarity is measured by histogram intersection. We train one-vs-all linear SVMs on the training videos' BoW representation for multiclass action classification, and the action label of a test video is given by:

$$y = \underset{y \in \mathcal{Y}}{arg\max} \begin{pmatrix} \mathbf{w}_y^r \\ \mathbf{w}_y^p \end{pmatrix} (\mathbf{x}^r \ \mathbf{x}^p) + b_y \tag{4}$$

where $\mathbf{x}^r$ and $\mathbf{x}^p$ are the BoW representations of root and part space-time segments of the test video respectively, $\mathbf{w}_y^r$ and $\mathbf{w}_y^p$ are entries of the trained separation hyperplane for roots and parts respectively, b_y is the bias term and $\mathcal{Y}$ is the set of action class labels under consideration.

For action localization, in a test video we find space-time segments that have positive contribution to the classification of the video and output the tracks that contain them. Specifically, given a testing video as a set of root space-time segments $S^p = \{\mathbf{s}_a^r\}$ and a set of part space-time segments $S^p = \{\mathbf{s}_b^p\}$, denote $C^r = \{\mathbf{c}_k^r\}$ and $C^p = \{\mathbf{c}_k^p\}$ as the set of code words that correspond to positive entries of $\mathbf{w}_y^r$ and $\mathbf{w}_y^p$ respectively. We compute the set U as

$$\begin{aligned} U = &\left\{ \hat{s}^r : \hat{s}^r = \underset{\mathbf{s}_a^r \in S^r}{arg\max}\, h(\mathbf{s}_a^r,\ \mathbf{c}_k^r),\ \forall \mathbf{c}_k^r \in C^r \right\} \\ &\cup \left\{ \hat{s}^p : \hat{s}^p = \underset{\mathbf{s}_b^p \in S^p}{arg\max}\, h(\mathbf{s}_b^p,\ \mathbf{c}_k^p),\ \forall \mathbf{c}_k^p \in C^p \right\} \end{aligned} \tag{5}$$

where function h measures the similarity of two space-time segments, for which we use histogram intersection of their feature vectors. We then output all the tracks that have at least one space-time segment in the set U as action localization results. In this way, although space-time segments in U may only cover a sparse set of frames, our algorithm is able to output denser localization results. Essentially these results benefit from the temporal relationships (*before, after*) among the root space-time segments in the same track.

Method	Supervision	Accuracy
Ours (Root + Part)	label	81.7%
Ours (Part only)	label	71.3%
Raptis *et al.* [18]	label + box	79.4%
Lan *et al.* [12]	label + box	73.1%

Table 1. Mean per-class classification accuracies on the UCF-Sports dataset. The training/testing split follows [12]. Unlike [12, 18], we do not require bounding box annotation for training.

Method	mAP
Ours (Root + Part)	53.3 %
Ours (Part only)	46.3 %
Gaidon *et al.* [6]	55.6%
Wang *et al.* [22]	53.4%
Laptev *et al.* [13]	36.9%
Patron-Perez *et al.* [17]	32.8 %

Table 2. Mean average precision (mAP) on the HighFive dataset.

	subset of frames				all frames			
	[20]	[21]	[12]	Ours	[20]	[21]	[12]	Ours
dive	16.4	36.5	43.4	**46.7**	22.6	37.0	-	**44.3**
golf	-	-	37.1	**51.3**	-	-	-	**50.5**
kick	-	-	36.8	**50.6**	-	-	-	**48.3**
lift	-	-	**68.8**	55.0	-	-	-	**51.4**
ride	62.2	**68.1**	21.9	29.5	63.1	**64.0**	-	30.6
run	50.2	**61.4**	20.1	34.3	48.1	**61.9**	-	33.1
skate	-	-	13.0	**40.0**	-	-	-	**38.5**
swing-b	-	-	32.7	**54.8**	-	-	-	**54.3**
swing-s	-	-	16.4	**19.3**	-	-	-	**20.6**
walk	-	-	28.3	**39.5**	-	-	-	**39.0**
Avg.	-	-	31.8	**42.1**	-	-	-	**41.0**

Table 3. Action localization results measured as average IOU (in %) on the UCF Sports dataset. '-' means result is not available. Note that, [20] and [21] need bounding boxes in training and their models are only for binary action detection, so their results are not directly comparable to ours.

Class	hand shake	high five	hug	kiss	**Avg.**
IOU	26.9	32.9	34.2	29.2	30.8
Recall	79.4	88.8	82.6	80.8	82.3

Table 4. Action localization performance measured as average IOU (in %) and recall (in %) on the High Five dataset.

5. Experiments

We conducted experiments on the UCF-Sports [19] and High Five [17] datasets to evaluate the proposed hierarchical space-time segments representation. These two datasets are challenging and representative of two different major types of actions: sports and daily interactions.

5.1. Experimental Setup

We implemented our formulation in Matlab.[1] The parameters for extracting hierarchical space-time segments are empirically chosen without extensive tuning and mostly kept the same for both datasets.

UCF-Sports Dataset: The UCF-Sports dataset [19] contains 150 videos of 10 different classes of actions. We use the training/testing split of [12]. For each root space-time segment, we use a $3 \times 3 \times 3$ space-time grid and compute HoG (histogram of oriented gradients), HoF (histogram of optical flow) and MBH (histogram of motion boundary) features in each space-time cell. The number of orientation bins used is 9. For part space-time segments, we use a $2 \times 2 \times 2$ space-time grid and the other settings are the same. We build a codebook of 2000 words for root space-time segments and 4000 words for parts. This dataset contains bounding box annotations on each video frame. While the compared methods use these annotations in their training, we do not use them in ours.

HighFive TV-interactions: The HighFive dataset [17] contains 300 videos from TV programs. 200 of them contain 4 different classes of daily interactions, and the other 100 are labeled as negative. We follow the training/testing split of [17]. We use the same space-time grid setting as in UCF-Sports, but to fairly compare with previous results in [6], we compute only MBH features in each space-time cell. We build a codebook of 1800 words for root space-time segments and 3600 words for parts. Again, we do not use the body bounding box annotations in our training.

5.2. Experimental Results

Action recognition: The action recognition results are shown in Table 1 and Table 2 for the UCF-Sports and High-Five datasets respectively.

[1]Code at http://www.cs.bu.edu/groups/ivc/software/STSegments/.

On the UCF-Sports dataset, our method is compared with two state-of-the-art methods [18, 12]. The method in [18] learns an action model as a Markov random field over a fixed number of dense trajectory clusters. The method in [12] uses a figure-centric visual word representation in a latent SVM formulation for both action localization and recognition. Both compared methods used more complex classifiers than the simple linear SVM used in ours. More importantly, both of them require expensive frame-wise human bounding box annotations, while ours does not. However, our method performs slightly better (by 2.3%) than [18] and significantly better (by 8.6%) than [12]. Although [18] achieves comparable classification performance with ours, it cannot provide meaningful action localization results. The method in [12] can output localization results, but its localization performance is significantly lower than ours (see Table 3), as will be discussed in detail later.

On the High Five dataset, our method is compared to four methods [13, 17, 22, 6]. The method in [6] uses non-linear SVM on a cluster tree of dense trajectories and produces state-of-the-art results. The results for [22] and [13] are produced by using SVM with the histogram intersection kernel on bag of dense trajectories and STIPs respectively.

The method in [17] uses structured SVM. Despite its implicity, our method achieves comparable performance with [6] and [22] and significantly better performance than [13] and [17]. None of the compared methods perform action localization as our method does.

To assess the benefit of extracting the relevant static space-time regions that are contained in the root space-time segments, we compare with a baseline that only uses space-time segments of parts. The results show that there is a significant performance drop (10.4% on UCF-Sports and 7.0% on High Five) if space-time segments of roots are not used. This supports our hypothesis that pose information captured by root space-time segments is useful for action recognition.

Action localization: Table 3 and Table 4 show the action localization results on the UCF-Sports and HighFive datasets. Fig. 6 visualizes localization results on some example frames of both datasets. The localization score is computed as the average IOU (intersection-over-union) over tested frames.

On the UCF-Sports dataset, the method of [12] can only produce localization results on a subset of frames, so we include comparisons on this subset. On this subset of frames, our method performs better than [12] on 9 out of 10 classes and the average performance is higher by 10.3%. We also provide our performances on all frames, which are similar to those on subset of frames. The methods of [20] and [21] only report action localization results on 3 classes (*running*, *diving* and *horse riding*) of UCF Sports. Our performance is higher than [20, 21] in one class (*diving*) but lower in the other two. All compared methods use expensive human bounding box annotations and their learning are much more complex than ours.

On the HighFive dataset, the IOU is measured over the frames that are annotated as containing the interactions. We achieve an IOU of 30.8%, which is still reasonably good but lower than our results on UCF Sports. We suspect this may be partly due to the low quality of human bounding box annotations used for evaluation, as most of them are too small, covering only the head area of the actors (see Fig. 6). To verify this, we also compute the recall, which is measured as the ratio of the area of intersection over the annotated action area. The high recall values reported in Table 4 confirm that the annotated action areas are mostly identified by our method. [20] and [21] have reported action localization results only on the *kiss* class, which are 18.5% and 39.5% respectively. Again, these results are not directly comparable, since our method requires much less supervision (only labels) compared to [20] and [21] which require human bounding boxes.

6. Conclusion and Future Work

In this work, we propose hierarchical space-time segments for action recognition that can be utilized to effectively answer *what* and *where* an action happened in realistic videos as demonstrated in our experiments. Compared to previous methods such as STIPs and dense trajectories, this representation preserves both relevant static and non-static space-time segments as well as their hierarchical relationships, which helped us in both action recognition and localization. One direction for future work is to make the method more robust to low video quality, as it may fail to extract good space-time segments when there is significant blur or jerky camera motion. A particularly promising direction for future work is to apply more advanced machine learning techniques to explore the hierarchical and temporal relationships provided within this representation for even better action recognition and localization.

Acknowledgments. This work was supported in part through a Google Faculty Research Award and by US NSF grants 0855065, 0910908, and 1029430.

References

[1] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001.

[3] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *CVPR*, 2009.

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[5] C. X. Chenliang Xu and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.

[6] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. In *BMVC*, 2012.

[7] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009.

[8] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *TPAMI*, 29(12):2247–2253, 2007.

[9] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[10] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010.

[11] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[12] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

Figure 5. Extracted segments for example video frames from the UCF Sports and High Five dataset. Each example frame is from a different video. The yellow boxes outlines the extracted segments. The inclusion of one box in another indicates the parent-children relationships

Figure 6. Example of our action localization results. The area covered by green mask is the ground truth from annotation, and the area covered by red mask is our action localization output. First five rows are from UCF-Sports, and last two rows are from HighFive.

[14] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[15] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV*, 2012.

[16] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[17] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. D. Reid. High five: Recognising human interactions in tv shows. In *BMVC*, 2010.

[18] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.

[19] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[20] D. Tran and J. Yuan. Optimal spatio-temporal path discovery for video event detection. In *CVPR*, 2011.

[21] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012.

[22] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[23] Y. Wang, K. Huang, and T. Tan. Human activity recognition based on r transform. In *CVPR*, 2007.

[24] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *CVPR*, 2011.

[25] Y. Xie, H. Chang, Z. Li, L. Liang, X. Chen, and D. Zhao. A unified framework for locating and recognizing human actions. In *CVPR*, 2011.

[26] Z. L. Yang Wang, Duan Tran and D. Forsyth. Discriminative hierarchical part-based models for human parsing and action recognition. *JMLR*, 13:30753102, 2012.