Video Action Detection with Relational Dynamic-Poselets

Limin Wang[1,2], Yu Qiao[2,⋆], Xiaoou Tang[1,2]

[1] Department of Information Engineering, The Chinese University of Hong Kong
[2] Shenzhen Key Lab of CVPR, Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences, Shenzhen, China
07wanglimin@gmail.com, yu.qiao@siat.ac.cn, xtang@ie.cuhk.edu.hk

Abstract. Action detection is of great importance in understanding human motion from video. Compared with action recognition, it not only recognizes action type, but also localizes its spatiotemporal extent. This paper presents a relational model for action detection, which first decomposes human action into temporal "key poses" and then further into spatial "action parts". Specifically, we start by clustering cuboids around each human joint into dynamic-poselets using a new descriptor. The cuboids from the same cluster share consistent geometric and dynamic structure, and each cluster acts as a mixture of body parts. We then propose a sequential skeleton model to capture the relations among dynamic-poselets. This model unifies the tasks of learning the composites of mixture dynamic-poselets, the spatiotemporal structures of action parts, and the local model for each action part in a single framework. Our model not only allows to localize the action in a video stream, but also enables a detailed pose estimation of an actor. We formulate the model learning problem in a structured SVM framework and speed up model inference by dynamic programming. We conduct experiments on three challenging action detection datasets: the MSR-II dataset, the UCF Sports dataset, and the JHMDB dataset. The results show that our method achieves superior performance to the state-of-the-art methods on these datasets.

Keywords: Action detection, dynamic-poselet, sequential skeleton model

1 Introduction

Action understanding in video [1] has attracted a great deal of attention in the computer vision community due to its wide applications in surveillance, human computer interaction, and content-based retrieval. Most of the research efforts have been devoted to the problem of action recognition using the Bag of Visual Words (BoVW) framework or variants thereof [29, 24, 11]. These particular designed methods for action recognition usually require a short video clip to be cropped from a continuous video stream. Apart from the class label, however,

⋆ corresponding author

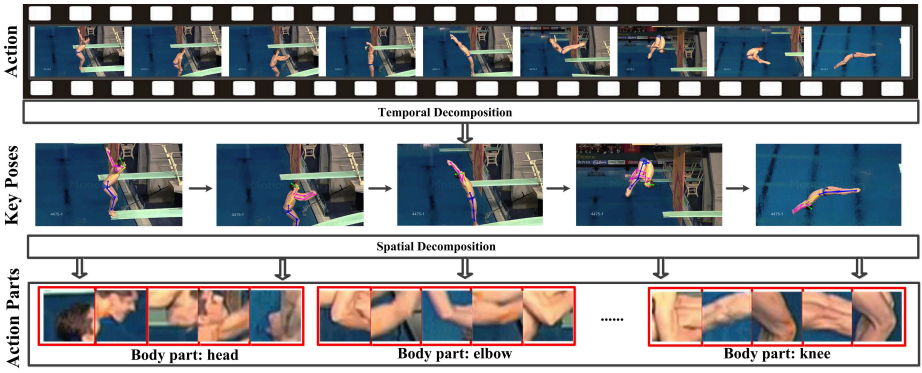

Fig. 1. Illustration of action decomposition. A video sequence first can be temporally decomposed into several short snippets, each of which corresponds to a key pose. For each key pose, the action can then be further decomposed spatially into several action parts (red boxes), each of which describes the appearance and motion of body part in a specific configuration. A body part is described by multiple action parts. Best view in color.

they cannot provide further information about the action, such as the location and pose of the actor. To overcome these limitations, we focus on the problem of action detection. Given a long video stream, we aim not only to recognize on-going action class, but also to localize its spatiotemporal extent (that is, the bounding box of the actor and the temporal duration of action), and estimate the pose of the actor.

Previous studies have shown that *pose* [23, 13, 10, 17, 7] and *motion* [27, 30, 15, 5] are key elements in understanding human actions from videos. Pose captures the static configurations and geometric constraints of human body parts, while motion refers to the local articulated movements of body parts and global rigid kinematics. As Figure 1 shows, an action sequence can be decomposed temporally into several snippets. In these snippets, the actors exhibit discriminative configurations of body parts for action understanding. We call these discriminative configurations of body parts as the *key poses* of an action. There is a temporal structure and global rigid motion (for example, translation) among these key poses. Each key pose can be further broken down spatially to *action parts*, each of which describes the appearance and motion of body part in a specific configuration. As in Figure 1, each red box corresponds to an action part and a body part is described by multiple action parts. However, modeling the action class still presents the following challenges:

– How to discover a collection of tightly-clustered action parts from videos. As the same body part exhibits large variations in the action (see Figure 1), it is not feasible to describe the body part using a single template. Mixture model will be a more suitable choice to handle large intra-variations of body parts. The cuboids belonging to the same mixture (action part) should not

only share similar visual appearance and pose configuration, but also exhibit consistent motion patterns. It is necessary to design effective descriptors to help tightly cluster body parts and satisfy these requirements.
– How to model the spatiotemporal relations of action parts. To handle large intra-class variation, each body part is represented by a mixture of action part. Each mixture component (action part) represents the feature template of the body part in a specific pose and motion configuration. A key pose can be viewed as a spatial arrangement of action parts, and an action contains a sequence of moving key poses. Thus, the action model must take into account the spatiotemporal relations among body part, co-occurrences of different mixture types, and local part templates jointly.

In order to address these issues, this paper proposes a unified approach to discover effective action parts and model their relations. Specifically, we first annotate articulated human poses in training video sequences to leverage the human-supervised information. Based on these annotations, we design an effective descriptor to encode both the geometric and motion properties of each cuboid. Using this descriptor, we are able to cluster cuboids that share similar pose configuration and motion patterns into consistent action parts, which we call *dynamic-poselets*. These dynamic-poselets then act as mixture components of body parts, and we propose a relational model, called *sequential skeleton model* (SSM), that is able to jointly learn the composites of mixture dynamic-poselets, spatiotemporal structures of action parts, and the local model for each part. Using a mixture of dynamic-poselet enables SSM to be robust for large intra-class variation, such as viewpoint changes and motion speed variations. We formulate the model learning problem in a structured SVM framework [21] and use the dual coordinate-descent solver [31] for parameter optimization. Due to the fact that the sequential skeleton model is tree-structured, we can efficiently detect the action instance by dynamic programming algorithm. We conduct experiments on three public datasets: the MSR-II dataset [4], the UCF Sports dataset [14], and the JHMDB dataset [7]. We show that our framework achieves state-of-the-art performance for action detection in these challenging datasets.

2 Related Works

Action recognition has been extensively studied in recent years [1]. This section only covers the works related to our method.

Action Detection. Action detection has been comprehensively studied [8, 14, 34, 4, 5, 32, 9, 33, 20, 19]. Methods in [4, 34, 9] used Bag of Visual Words (BoVW) representation to describe action and conduct a sliding window scheme for detection. Yuan *et al.* [34] focused on improving search efficiency, while Cao *et al.* [4] mainly evaluated cross-dataset performance. Methods in [8, 14, 5] untilized global template matching with different features. Yao *et al.* [32] and Yu *et al.* [33] resorted to the Hough voting method of local cuboids for action detection, while Lan *et al.* [9] resorted to latent learning to locate action automatically.

Tran *et al.* [20] casted action detection task as a spatiotemporal structure regression problem and leveraged efficient Max-Path search method for detection. Tian *et al.* [19] extended the 2D part deformable model to 3D cases. Our method is different from these other methods in that we consider motion and pose in a unified framework for video-based action detection.

Parts in Action. The concept of "action part" appeared in several previous works, either implicitly or explicitly [12, 13, 27, 22, 26]. Raptis *et al.* [12] clustered trajectories of similar motion speed in a local region, with each cluster center corresponding to an action part. They modeled action in a graphical model framework to constrain the spatiotemporal relations among parts. Ullah *et al* [22] presented a supervised approach to learn the motion descriptor of actlets from synthetic videos. Wang *et al.* [27] proposed to cluster cuboids with high-motion salience into 3D parts, called *motionlets*, based on low-level features such as HOG and HOE. Raptis *et al.* [13] resorted to the poselet part proposed for static image, and used a sequence model to model the temporal structure. Wang *et al.* [26] designed a discriminative clustering method to discover the "temporal parts" of action, called *motion atoms*. Inspired by the success of poselets [2] and phraselets [6] in image-based tasks, we have designed spatiotemporal action part, called dynamic-poselets. Dynamic-poselets capture both the pose configuration and motion pattern of local cuboids, which are suitable for action detection in video.

Relational Model in Action. Several previous works [9, 3, 19, 28, 18] have considered the relations among parts for action recognition and detection. Lan *et al.* [9] detected 2D parts frame-by-frame with tracking constraints using CRF. Brendel *et al.* [3] proposed a spatiotemporal graph to model the relations over tubes and to represent the structure of action. Tian *et al.* [19] proposed a spatiotemporal deformable part models for action and obtained state-of-the-art performance. Wang *et al.* [28] designed a Latent Hierarchical Model (LHM) to capture the temporal structure among segments in a coarse-to-fine manner. Sun *et al.* [18] considered the temporal relations of segments by exploiting activity concept transitions in video events. Our relational model differs from these models in two main aspects. Firstly, our model is constructed by explicitly modeling the human pose, which has been proved to be an important cue for action understanding [7]. Secondly, our model is composed of mixtures of parts, similar to that of [31] for static pose estimation, and this mixture representation is effective at handling large intra-class variations in action.

3 Dynamic-Poselets

This section describes the method for learning action parts or dynamic-poselets, specific to a given action class. Dynamic-poselets are cuboids that are tightly clustered in both pose and motion configuration space. Due to the large intra-class variation and low resolution quality of action video, it is difficult to directly group cuboids based on low-level appearance and motion features such as HOG and HOF [24]. Similar to the methods of constructing image representation such

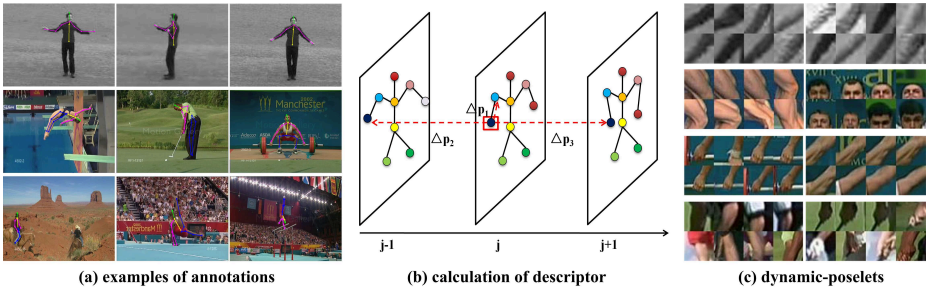

Fig. 2. Illustration of dynamic-poselet construction. (a) Some examples of human pose annotations in the training videos. (b) The descriptor extraction for dynamic-poselet clustering. For each joint, we calculate its spatial offset ΔP_1 with respect to its parent, and its temporal offsets ΔP_2 and ΔP_3 with respect to itself in previous and subsequent frames. (c) Some examples of clusters (dynamic-poselets) in training videos.

as poselet [2] and phraselet [6], we leverage the human annotations of human joints, and propose a new descriptor based on the geometric configuration and the moving direction of a cuboid.

For a specific action class, we assume that we have been given training videos with human joint annotations. Typical human joints (body parts) include head, shoulder, elbow and so on. Some annotation examples are shown in Figure 2. Let K be the number of body parts in our annotations, and $i \in \{1, \cdots, K\}$ denote the i^{th} human body part. Let $p^v_{i,j} = (x^v_{i,j}, y^v_{i,j})$ denote the position of body part i in the j^{th} frame of video v. Let M_i be the number of mixture for body part i and $t^v_{i,j} \in \{1, \cdots, M_i\}$ denote the mixture type of body part i in the j^{th} frame of video v. In the remaining part of this section, we will show how to obtain the mixture types of body parts for training videos.

Intuitively, the spatial geometric configuration of a human body part with respect to others in the same frames will determine its pose and appearance, and the temporal displacement with respect to the same joints from adjacent frames will represent the articulated motion. Based on this assumption, we have designed the following new descriptor for a cuboid around each human joint:

$$f(p^v_{i,j}) = [\Delta p^{v,1}_{i,j}, \Delta p^{v,2}_{i,j}, \Delta p^{v,3}_{i,j}], \tag{1}$$

where $\Delta p^{v,1}_{i,j} = p^v_{i,j} - p^v_{par(i),j}$ is the offset of joint i with respect to its parent $par(i)$ in current frame j of video v, $\Delta p^{v,2}_{i,j} = p^v_{i,j} - p^v_{i,j-1}$ and $\Delta p^{v,3}_{i,j} = p^v_{i,j} - p^v_{i,j+1}$ denote the temporal displacements of joint i with respect to the same joints in previous and subsequent frames of video v, respectively (see Figure 2). Essentially, $\Delta p^{v,1}_{i,j}$ encodes the pose and appearance information, and $\Delta p^{v,2}_{i,j}$ and $\Delta p^{v,3}_{i,j}$ capture the motion information.

To make the descriptor invariant to scale, we estimate the scale for each body part in a video v. The scale of body part is estimated by $s^v_{i,j} = \text{headlength}^v_j \times \text{scale}_{i,j}$, where headlength^v_j is the head length of the j^{th} frame in video v, $\text{scale}_{i,j}$

is the canonical scale of joint part (i, j) measured in human head length, whose value is usually 1 or 2. Thus, we obtain the scale invariant descriptor as follows:

$$\begin{aligned} \overline{f(p_{i,j}^v)} &= [\overline{\Delta p_{i,j}^{v,1}}, \overline{\Delta p_{i,j}^{v,2}}, \overline{\Delta p_{i,j}^{v,3}}], \\ \overline{\Delta p_{i,j}^{v,k}} &= [\Delta x_{i,j}^{v,k}/s_{i,j}^v, \Delta y_{i,j}^{v,k}/s_{i,j}^v](k = 1, 2, 3). \end{aligned} \tag{2}$$

Using the descriptor above, for each body part, we separately run k-means clustering algorithm over the cuboids around this joint extracted from training videos. Each cluster corresponds to an action part, called *dynamic poselet*, and the body part is represented as a mixture of action part (dynamic poselet). The cluster label is the mixture type t of body parts in training videos. Some examples of clusters (dynamic-poselets) are shown in Figure 2. These results indicate that the proposed descriptor is effective at obtaining tightly-clustered cuboids with similar pose, appearance, and movement. Meanwhile, we find it is important to leverage the motion term (i.e., $\Delta p_{i,j}^{v,2}, \Delta p_{i,j}^{v,3}$) in the descriptor to cluster dynamic-poselets. See the examples of the top row in Figure 2, where the two kinds of dynamic-poselets are from hand-waving action. If we ignore the motion term in our descriptor, the two kinds of dynamic-poselets will be merged in the same cluster because they share similar appearance and pose configuration. However, the two kinds of dynamic-poselets are different in motion, with one corresponds to moving down and the other to moving up.

4 Sequential Skeleton Model

Figure 3 provides an overview of our approach. During the training phase, we first cluster the cuboids into consistent dynamic-poselets using the descriptor (Equation (2)) in the previous section. Then, based on the clustering the results, we develop a *Sequential Skeleton Model* (SSM) to describe each action class. The SSM is described in the remainder of this section, and the learning and inference algorithms are proposed in the next section.

We now propose the SSM of a specific action class to describe the spatiotemporal configuration of a collection of action parts (dynamic-poselets). Our model not only imposes the spatiotemporal structure and geometric arrangement of dynamic-poselets, but also learns the co-occurrence of mixture types for action parts. The two goals interact with each other, and the geometric arrangement of action parts affects the mixture types, and vice versa. To encode such relationships jointly, we extend the framework of mixture-of-parts [31] to spatiotemporal domain and design a relational model (that is, SSM).

Let $G = (V, E)$ be a spatiotemporal graph with node $V = \{(i, j)\}_{i,j=1}^{K,N}$ denoting the body part in human action where K is the number of body parts, N is the number of key poses, and edge $E = \{(i, j) \sim (m, n)\}$ denote the relations among adjacent body parts (see Figure 3). How to determine the location of key pose of training videos will be specified in next section. Let v be a video clip, p be the pixel positions of body parts in key poses, and t be the mixture types of

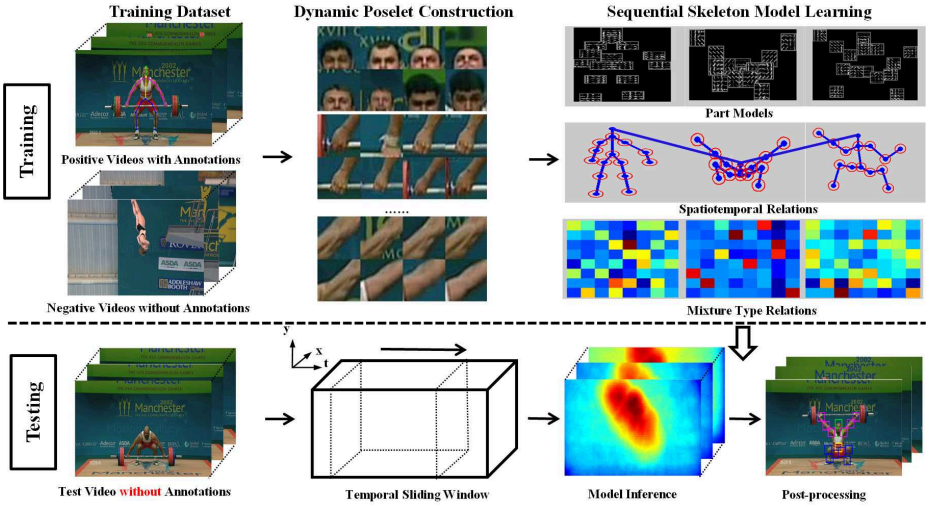

Fig. 3. Overview of our approach. For training, we annotate human joints for several key poses in the positive samples. We first cluster the cuboids around each human joint into dynamic-poselets. Then, each dynamic-poselet acts as a mixture of body parts and is fed into the SSM training. Our SSM is composed of three components: part models, spatiotemporal relations, and mixture type relations. For testing, we first use a temporal sliding window and then conduct inference of SSM. Finally, we resort to post-processing techniques such as no-maximum suppression to obtain the detection results. It is worth **noting that** there is no annotation for testing samples.

body parts in key poses. The discriminative score with the current configuration of dynamic poselets is then defined as follows:

$$S(v,p,t) = b(t) + \sum_{j=1}^{N}\sum_{i=1}^{K} \alpha_i^{t_{i,j}}\phi(v,p_{i,j}) + \sum_{(i,j)\sim(m,n)} \beta_{(i,j),(m,n)}^{t_{i,j}t_{m,n}}\psi(p_{i,j},p_{m,n}), \quad (3)$$

where $(\{b\},\{\alpha\},\{\beta\})$ are model parameters, ϕ and ψ are visual features.

Mixture Type Relations. $b(t)$ is used to define a "prior" with preference to some mixture combinations, which factors into a summation of the following terms :

$$b(t) = \sum_{j=1}^{N}\sum_{i=1}^{K} b_{i,j}^{t_{i,j}} + \sum_{(i,j)\sim(m,n)} b_{(i,j),(m,n)}^{t_{i,j}t_{m,n}}, \quad (4)$$

where term $b_{(i,j),(m,n)}^{t_{i,j}t_{m,n}}$ encodes the compatibility of mixture types. Intuitively, some configurations of mixture types are more compatible with current action class than others. In the case of hand-waving action, moving-up arms tends to co-occur with moving-up hands, while moving-down arms tends to co-occur with moving-down hands. With this term in the relational model, we are able to discover these kinds of co-occurrence patterns.

Part Models. $\alpha_i^{t_{i,j}}\phi(v, p_{i,j})$ is the model for a single action part. We denote $\phi(v, p_{i,j})$ as the feature vector extracted from video v in location $p_{i,j}$. $\alpha_i^{t_{i,j}}$ denotes the feature template for the mixture $t_{i,j}$ of i^{th} body part. Note that the body part template $\alpha_i^{t_{i,j}}$ is shared between different key poses of the same action. The visual features will be specified in Section 6.

Spatiotemporal Relations. We denote $\psi(p_{i,j}, p_{m,n}) = [dx, dy, dz, dx^2, dy^2, dz^2]$ as a quadratic deformation vector computed from the displacement of child node (i, j) relative to its anchor point determined by parent node (m, n). Then $\beta_{(i,j),(m,n)}^{t_{i,j}t_{m,n}}$ represents the parameters of quadratic spring model between mixture type $t_{i,j}$ and $t_{m,n}$. Note that the spring model is related to mixture types, which means the spatiotemporal constraints are dependent on both local appearance and motion. For example, the spatial relationship between hands and arms is different in moving-up and moving-down processes. Currently, we explicitly enforce that the temporal locations of parts should be the same within a key pose.

5 Model Learning and Inference

The *learning task* aims to determine the structure of Graph $G = (V, E)$ and estimate the model parameters $\theta = (\{b\}, \{\alpha\}, \{\beta\})$ in Equation (3) for each action class. For graph structure, we currently resort to a simple initialization method. For each key pose, we determine its structure as a skeleton tree model independently. For each action, in the temporal domain, we add an edge between the heads of adjacent key poses. This method is simple but effective for determining the graph structure.

Given the action-specific graph G and a training set $\{v_i, y_i, p^{v_i}, t^{v_i}\}_{i=1}^M$, the score function of Equation (3) is linear with model parameters θ, and we can rewrite the score function in the form $\theta \cdot \Phi(v_i, p^{v_i}, t^{v_i})$. Thus, we formulate the parameter learning problem in the following structured SVM framework [21]:

$$\begin{aligned} &\arg \min_{\theta, \{\xi_i \geq 0\}} \frac{1}{2}\|\theta\|_2^2 + C\sum_{i=1}^{M}\xi_i \\ \text{s.t. } &\theta \cdot \Phi(v_i, p^{v_i}, t^{v_i}) \geq 1 - \xi_i, \quad \text{if } y_i = 1 \\ &\theta \cdot \Phi(v_i, p, t) \leq -1 + \xi_i, \quad \forall (p, t), \quad \text{if } y_i = -1. \end{aligned} \tag{5}$$

The negative examples are collected from the action videos with different labels. This is a standard convex optimization problem, and many well-tuned solvers are public available. Here we use the dual coordinate-decent solver [31]. Together with the process of dynamic-poselets clustering, the whole learning process is shown in Algorithm 1.

Firstly, for each positive example, we extract the descriptors for the annotated human parts and conduct k-means to cluster these cuboids into dynamic-poselets. From the clustering results, we obtain the mixture labels for the parts of positive examples. We then train each dynamic-poselet independently using classical SVM. This training process provides an initialization for the template

Algorithm 1: Dynamic-poselets clustering and model learning.

Data: Positive samples: $\mathcal{P} = \{v_i, p^{v_i}, y_i\}_{i=1}^{T_1}$, negative samples: $\mathcal{N} = \{v_j, y_j\}_{j=1}^{T_2}$.
Result: Graph: G and parameters: θ.
`// Dynamic-poselets clustering`
- Extract the descriptors of each body part (i, j).
- Using the descriptors, run k-means on the local cuboids, and obtain the mixture type $t_{i,j}$ for each body part.
`// Model parameter learning`
- Initialize the graph structure G.
foreach *part* i *and mixture type* t_i **do**
  $\alpha_i^{t_i} \longleftarrow$ `SVMTrain` $(\{v_i, p^{v_i}, t^{v_i}\}, i, t_i)$.
end
- Use the part template above to initialize the model parameters θ.
for $i \leftarrow 1$ **to** C **do**
  - Mining Hard negative examples: $N \leftarrow$ `NegativeMining`$(\theta, G, \mathcal{N})$.
  - Retrain model jointly: $\theta \leftarrow$ `JointSVMTrain`$(\theta, G, N, \{v_i, p^{v_i}, y^{v_i}\})$.
end
- **return** *graph* G *and parameters* θ.

parameters in the relational model. Based on this initialization, we iterate between mining hard negative examples and retraining model parameters jointly as in the Structured SVM. The iteration is run for a fixed number of times.

Implementation Details. In the current implementation, the number of key poses is set as 3. Due to the subjectivity of key pose, we design a simple yet effective method to determine the locations of key pose given a specific video. We start by dividing the video into three segments of equal duration. Then, in each segment, we uniformly sample a frame as the key pose. To handle the temporal miss-alignment of training videos, we conduct uniform sampling four times and obtain four instances for each positive video. This method also increases the number of training samples for structured SVM learning and makes the learning procedure more stable. The iteration times C of Algorithm 1 is set as 5.

The *inference task* is to determine the locations and mixture types (p, t) of a given video v by maximizing the discriminative score $S(v, p, t)$ defined in Equation (3). Since our relational graph $G = (V, E)$ is a tree, this can be done efficiently with dynamic programming. For a node (i, j) at location $p_{i,j}$ with mixture type $t_{i,j}$, we can compute its score according to the message passed from its children $kids((i, j))$:

$$S_{i,j}(p_{i,j}, t_{i,j}) = b_{i,j}^{t_{i,j}} + \alpha_i^{t_{i,j}} \phi(v, p_{i,j}) + \sum_{(m,n) \in kids((i,j))} C_{m,n}(p_{i,j}, t_{i,j}), \tag{6}$$

$$\begin{aligned} C_{m,n}(p_{i,j}, t_{i,j}) = \max_{t_{m,n}} \Big\{ & b_{(m,n),(i,j)}^{t_{m,n} t_{i,j}} + \\ & \max_{p_{m,n}} \left[S_{m,n}(p_{m,n}, t_{m,n}) + \beta_{(m,n),(i,j)}^{t_{m,n} t_{i,j}} \psi(p_{m,n}, p_{i,j}) \right] \Big\}. \end{aligned} \tag{7}$$

Equation (6) computes the local score of part (i, j) located at $p_{i,j}$ with mixture type $t_{i,j}$, and Equation (7) collects message from the child nodes and computes scores for every mixture type $t_{m,n}$ and possible location $p_{m,n}$ to obtain the best score given the parent's location $p_{i,j}$ and type $t_{i,j}$. Based on these recursive functions, we can evaluate the score in a depth-first-search (DFS) order and pass the message from leaf nodes to the root node. Once the message has been passed to the root node, $S_{1,1}(p_{1,1}, t_{1,1})$ represents the best score for each root position and mixture type.

Implementation Details. During detection, we will use the temporal sliding window of 40 frames with a step size of 20, if the testing sample is a video stream instead of a video clip. For final detection, we choose a threshold of -2 for detection score to generate multiple detections, and use the post-processing technique of non-maximum suppression to avoid repeated detections [31].

6 Experiments

In this section, we present the experimental results on three public datasets: the MSR-II dataset [4], the UCF Sports dataset [14], and the JHMDB dataset [7].

Experiment Details. For all these datasets, we extract Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF) as low-level features [24]. HOG features capture the static appearance and HOF features describe the motion information. The feature cell size is up to the resolution of the video, and we select a cell size of $4 \times 4 \times 2$ for the MSR-II dataset, and $8 \times 8 \times 2$ for the UCF Sports and the JHMDB dataset. The cuboid size of each part is determined automatically according to the size of the person in the video. For the mixture number of each part, the default setting is 8.

Results on the MSR-II Dataset. The MSR-II dataset includes three action classes: boxing, hand-waving, and hand-clapping. The dataset is composed of 54 video sequences that are captured in realistic scenarios such as parties, schools, and outer traffics, with cluttered background and moving people. Following the scheme in [4], we use a subset of the KTH [16] for training and test our model on the MSR-II dataset. Specifically, we train our model on the KTH dataset with 20 positive examples for each class and the number of joints is 10 (see Figure 2). For action detection evaluation, we use the same scheme in [4] and report the average precision (AP) for each class. Although the action class is relatively simple, the MSR-II dataset is a challenging benchmark for action detection due to its realistic scene and the cross-dataset testing scheme. The experimental results can demonstrate the effectiveness of our approach for detecting simple action in realistic scenarios.

We plot the precision-recall (PR) curves in Figure 4 and report the average precision (AP) for each class in Table 1. Our method performs quite well on the action of hand waving but relatively poorly on the action of boxing. This result could be due to the fact that the action of boxing involves heavy occlusion with two arms and pose estimation in the action of boxing is more difficult than in hand waving. We compare our results with two other methods: GMM adap-

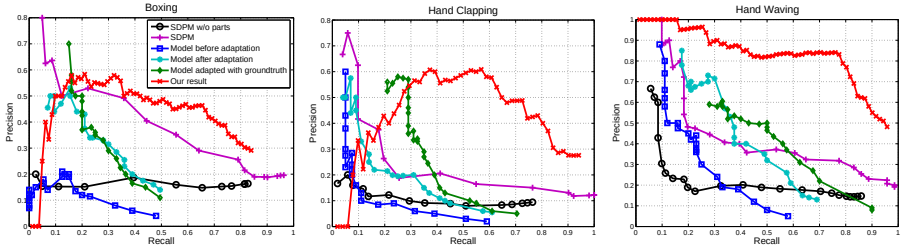

Fig. 4. Results on the MSR-II dataset. We plot the PR curves for the three action classes: boxing, hand-clapping, and hand-waving. We compare our results with GMM methods with or without adaption [4] and SDPM [19] (state-of-the-art). Best viewed in color.

tion method [4] (baseline) and spatiotemporal deformation part model (SDPM) (state-of-the-art method) [19]. We observe that our method outperforms these methods in all action classes. Especially for the actions of hand-waving and hand-clapping, our APs are almost twice those of state-of-the-art results. In these two action classes, key poses are well detected and yield important cues for discriminating action from other classes. For the action of boxing, the improvement of our method is not so significant. The superior performance demonstrates the effectiveness of our key pose based approach for detecting simple actions in realistic scenarios.

Table 1. Results on the the MSR-II dataset. We report the APs for the three action class and mean AP (mAP) over all classes. We compare our results with GMM methods with or without adaption [4] and SDPM [19] (state-of-the-art).

Method	Boxing	Hand-clapping	Hand-waving	mAP
Baseline [4]	17.48%	13.16%	26.71%	19.12%
SDPM [19]	38.86%	23.91%	44.70%	35.82%
Our result	41.70%	50.15%	80.85%	**57.57**%

Results on the UCF Sports Dataset. The UCF Sports dataset [14] is composed of 150 realistic videos from sports broadcasts. The dataset has 10 action classes including diving, lifting, skating and so on (see Figure 5). Following the experimental setting [9], we split the dataset into 103 samples for training and 47 samples for testing. We evaluate the action localization using the "intersection-over-union" criterion and a detection is regarded as correct if the measure is larger than 0.2 and the predicted label matches. We plot the ROC curves and report the AUC for each action class. The UCF Sports dataset is more challenging than the MSR-II dataset due to the fact that the videos are cropped from sports broadcasts with large intra-class variations caused by camera motion, scale changes, viewpoint changes, and background clutter. The experiments on the UCF sports dataset can verify the effectiveness of our approach for more complex actions with articulated poses.

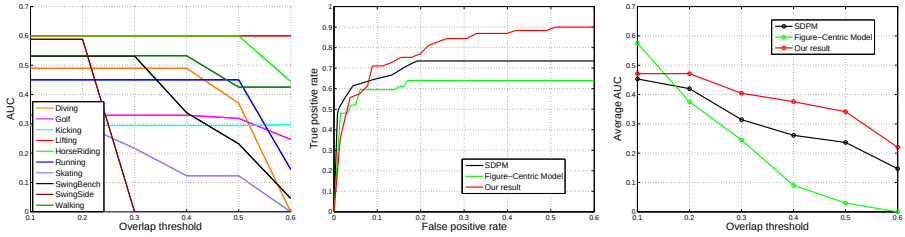

Fig. 5. Results on the UCF Sports dataset. **Left**: We plot the AUC per class of our detection result with a varying overlap thresholds. **Center**: We compare our results with the Figure-Centric Model [9] and the SDPM (state of the art) [19], when the overlap threshold is set as 0.2. **Right**: We compare the detection performances of these methods with varying thresholds (from 0.1 to 0.6). Best viewed in color.

Figure 5 shows the results of our method. We first plot the AUC of the ROC curve for each action class with respect to the varying overlap threshold in the left of Figure 5. These curves show that our method achieves a high detection rate for many action classes, such as lifting, horse-riding, and walking. We compare our approach with two recently published methods: figure-centric model (FCM) [9] and spatiotemporal deformable part model (SDPM) [19]. The FCM resorts to latent learning and detects 2D parts frame-by-frame with smooth constraints. The SDPM obtains the state-of-the-art detection performance on the UCF Sports dataset. From the comparison of the ROC curve and the AUC curve with respect to varying overlap thresholds in Figure 5, we conclude that our method outperforms the others and obtains the state-of-the-art performance on this challenging dataset. These results demonstrate that our method is not only suitable for simple action class such as MSR-II dataset, but also effective for more realistic action classes recorded in unconstrained environment.

Results on the JHMDB Dataset. The JHMDB dataset is a recently proposed dataset with full human annotation of body joints [7]. It is proposed for a systematic action recognition performance evaluation using thoroughly human annotated data. It also selects a subset of videos, called **sub-JHMDB**, each of which has all the joints inside the frames. The sub-JHMDB contains 316 clips distributed over 12 categories, including catch, pick, and swing (see Figure 6). The results in [7] show that this subset is much more challenging for action recognition than the whole dataset. No action detection results are reported in this subset and we have made the first attempt with our method. Using the same evaluation in the UCF Sports dataset, we plot the ROC curves and report the AUC for each action class.

We plot the AUC of ROC curve for each action class with respect to the varying overlap thresholds in the left of Figure 6. From the results, we observe that our method still performs quite well for some action classes on this more challenging dataset, such as golf, swing, and push. However, due to the challenges caused by low resolution, strong camera shaking, illumination changes, some action classes obtain relatively low detection rates such as jump and climb-stairs. In order to compare our method with others, we adapt the state-of-the-art

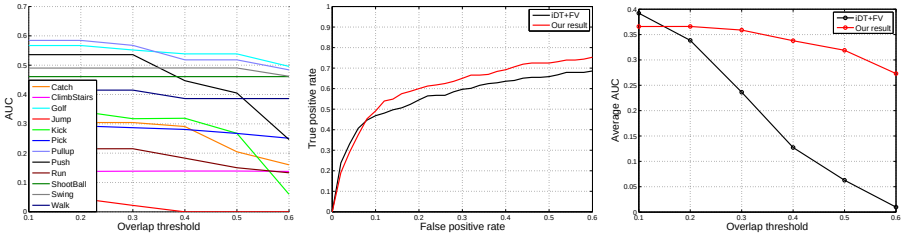

Fig. 6. Results of the sub-JHMDB dataset. **Left**: We plot the AUC per class of our detection result with varying overlap thresholds. **Center**: We compare our results with the state-of-the-art approach in action recognition [25], when the overlap threshold is set as 0.2. **Right**: We compare the detection performance with varying thresholds (from 0.1 to 0.6). Best viewed in color.

approach [25] in action recognition to action detection, and design a very competitive baseline method. Specifically, we use the improved dense trajectories (iDTs) as low-level features and choose Fisher Vector as encoding method. It should be noted that the iDTs are improved version of dense trajectories (DTs) [24] with several pre-processing techniques such as camera motion estimation and compensation, moving human detection, while our method does not require such pre-processing techniques. For each action class, we train a SVM using the fisher vector that aggregates the iDTs from the actor volume; that is, we eliminate the iDTs in the background. For detection, we conduct multiscale window scanning and use non-maximum suppression. Our comparison results are shown in the right of Figure 6 and the results show that our method obtains better detection performance, especially when the overlap threshold is large. The superior performance of our method compared to the state-of-the-art approach in action recognition indicates the importance of human pose in action understanding, especially for accurate action localization.

Examples of Detection Results. Some action detection examples on the three datasets are shown in Figure 7. We show the key poses automatically detected by our method. From these examples, we observe that our model is able to not only detect human actions, but also estimate human pose accurately in most cases.

7 Conclusion

This paper has proposed an approach for action detection in video by taking account of both cues of motion and pose. To handle the large variations of body part in action videos, a *action part* is designed as a mixture component. Guided by *key pose* decomposition, a relational model is then developed for joint modeling of spatiotemporal relations among body part, co-occurrences of mixture type, and local part templates. Our method achieves superior performance, as evidenced by comparing them to the state-of-the-art methods. In addition to action detection, our model is able to estimate human pose accurately in many

Fig. 7. Examples of action detection in three datasets. Our model is able to detect human actions and also estimate human poses accurately in most cases. Best viewed in color.

cases, which also provides insights for the research of human pose estimation in videos.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable suggestions in improving this paper. We also like to thank Xiaojiang Peng and Zhuowei Cai for their help in the annotatios of the UCF Sports dataset. Yu Qiao is supported by National Natural Science Foundation of China (91320101), Shenzhen Basic Research Program (JCYJ20120903092050890, JCYJ20120617114614438, JCYJ20130402113127496), 100 Talents Programme of Chinese Academy of Sciences, and Guangdong Innovative Research Team Program (No.201001D0104648280).

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Comput. Surv. 43(3), 16 (2011)
2. Bourdev, L.D., Maji, S., Malik, J.: Describing people: A poselet-based approach to attribute classification. In: ICCV (2011)

3. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV (2011)
4. Cao, L., Liu, Z., Huang, T.S.: Cross-dataset action detection. In: CVPR (2010)
5. Derpanis, K.G., Sizintsev, M., Cannons, K.J., Wildes, R.P.: Efficient action spotting based on a spacetime oriented structure representation. In: CVPR (2010)
6. Desai, C., Ramanan, D.: Detecting actions, poses, and objects with relational phraselets. In: ECCV (2012)
7. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV (2013)
8. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
9. Lan, T., Wang, Y., Mori, G.: Discriminative figure-centric models for joint action localization and recognition. In: ICCV (2011)
10. Packer, B., Saenko, K., Koller, D.: A combined pose, object, and feature model for action understanding. In: CVPR (2012)
11. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. CoRR abs/1405.4506 (2014)
12. Raptis, M., Kokkinos, I., Soatto, S.: Discovering discriminative action parts from mid-level video representations. In: CVPR (2012)
13. Raptis, M., Sigal, L.: Poselet key-framing: A model for human activity recognition. In: CVPR (2013)
14. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR (2008)
15. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR (2012)
16. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR (2004)
17. Singh, V.K., Nevatia, R.: Action recognition in cluttered dynamic scenes using pose-specific part models. In: ICCV (2011)
18. Sun, C., Nevatia, R.: Active: Activity concept transitions in video event classification. In: ICCV (2013)
19. Tian, Y., Sukthankar, R., Shah, M.: Spatiotemporal deformable part models for action detection. In: CVPR (2013)
20. Tran, D., Yuan, J.: Max-margin structured output regression for spatio-temporal action localization. In: NIPS (2012)
21. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML (2004)
22. Ullah, M.M., Laptev, I.: Actlets: A novel local representation for human action recognition in video. In: ICIP (2012)
23. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: CVPR (2013)
24. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV 103(1) (2013)
25. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
26. Wang, L., Qiao, Y., Tang, X.: Mining motion atoms and phrases for complex action recognition. In: ICCV (2013)
27. Wang, L., Qiao, Y., Tang, X.: Motionlets: Mid-level 3D parts for human motion recognition. In: CVPR (2013)

28. Wang, L., Qiao, Y., Tang, X.: Latent hierarchical model of temporal structure for complex activity classification. TIP 23(2) (2014)
29. Wang, X., Wang, L., Qiao, Y.: A comparative study of encoding, pooling and normalization methods for action recognition. In: ACCV (2012)
30. Yang, Y., Saleemi, I., Shah, M.: Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. TPAMI 35(7) (2013)
31. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
32. Yao, A., Gall, J., Gool, L.J.V.: A Hough transform-based voting framework for action recognition. In: CVPR (2010)
33. Yu, G., Yuan, J., Liu, Z.: Propagative Hough voting for human activity recognition. In: ECCV (2012)
34. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)