Pose Filter Based Hidden-CRF Models for Activity Detection

Prithviraj Banerjee and Ram Nevatia

University of Southern California, Los Angeles, USA

Abstract. Detecting activities which involve a sequence of complex pose and motion changes in unsegmented videos is a challenging task, and common approaches use sequential graphical models to infer the human pose-state in every frame. We propose an alternative model based on detecting the key-poses in a video, where only the temporal positions of a few key-poses are inferred. We also introduce a novel pose summarization algorithm to automatically discover the key-poses of an activity. We learn a detection filter for each key-pose, which along with a bag-of-words root filter are combined in an HCRF model, whose parameters are learned using the latent-SVM optimization. We evaluate the performance of our model for detection on unsegmented videos on four human action datasets, which include challenging crowded scenes with dynamic backgrounds, inter-person occlusions, multi-human interactions and hard-to-detect daily use objects.

Keywords: Activity detection, Key-poses, CRFs, Latent-SVM.

1 Introduction

There has been considerable research in classifying segmented videos, however there has been comparatively less progress on the more challenging task of activity detection, where multiple instances of an activity are simultaneously localized and classified in un-segmented videos. Detection is an important task, as in real world applications like surveillance, the activities of interest occur only for a part of the video. We propose a novel activity detection algorithm based on automatically discovering the key-poses in the activity, and learning a key-pose filter based Hidden Conditional Random Field (HCRF) model. We focus on activities primarily defined by a sequence of complex pose and motion changes, which can involve interactions with objects or other humans in the scene.

Activity recognition algorithms can be broadly categorized based on their structure modeling capabilities. A common class of approaches [9,26] train classifiers on video-wide statistics of local features, and ignore the local temporal dynamics of the activity. To classify unsegmented videos, they typically use an inefficient sliding window approach [30], which can be sensitive to window size. A complementary approach [11,12] learns a sequential motion model, and performs classification based on state assignments inferred from every frame in the video; to keep the inference tractable, these further require a Markovian assumption

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 711–726, 2014.

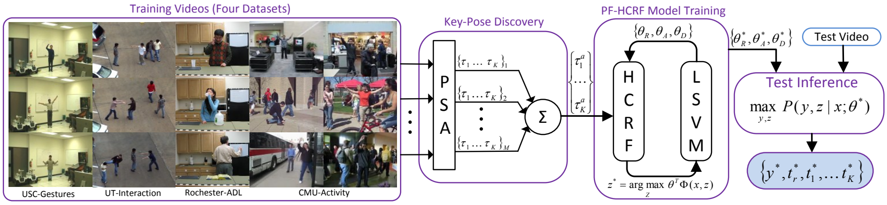

Training Videos (Four Datasets)
USC-Gestures
UT-Interaction
Rochester-ADL
CMU-Activity
Key-Pose Discovery
P S A
$\{\tau_1 \dots \tau_K\}_1$
$\{\tau_1 \dots \tau_K\}_2$
$\{\tau_1 \dots \tau_K\}_M$
Σ
$\{\tau_1^a \dots \tau_K^a\}$
PF-HCRF Model Training
$\{\theta_R, \theta_A, \theta_D\}$
HCRF
LSVM
$z^* = \arg\max_z \theta^T \Phi(x,z)$
$\{\theta_R^*, \theta_A^*, \theta_D^*\}$
Test Video
Test Inference
$\max_{y,z} P(y,z \mid x;\theta^*)$
$\{y^*, t_r^*, t_1^*, \dots t_K^*\}$

Fig. 1. Flow diagram of our proposed algorithm

between adjacent frames, and fail to capture long range dynamics in the activity, making them sensitive to variations in activity styles and action-durations.

We argue that for activity *detection*, it is sufficient to determine the *presence or absence* of certain key states in an observation sequence, and whether certain temporal relationships between the state detections are satisfied. Recognizing actions using a subset of the frames has been explored previously [3,15,20,21], however these do not address the problem of automatically discovering the important states/sub-sequences of an activity, and either perform exhaustive search over all possible sub-sequences [20], rely on hand annotations [3,21], or use a manually defined list of relevant poses [15], requiring separate annotated pose data for each pose-detector. There exist methods for automatic discovery of key-states [11,31], however [11] relies on hard to obtain mocap data, and [31] ignores temporal structure.

We propose a novel graphical model for activity detection, where the random variables to be inferred are the temporal locations of the key-poses. Key-poses represent the important human pose configurations in an activity, and are a natural choice for defining a key-state in our model. Our algorithm automatically discovers the relevant key-pose definitions in an activity, and learns a set of key-pose detection filters, and pools their detection responses, while satisfying the temporal relationships between them.

Our contributions are multi-fold: (1) The relevant key-poses are discovered automatically, (2) the key-pose detection filters are learned jointly in a discriminative HCRF framework, and do not require manually annotated pose-specific training data, (3) the temporal locations of the key-pose detections correspond to the active segments in the video stream, enabling activity detection in unsegmented videos, and (4) the key-poses correspond to a natural semantic interpretation. We show results on 4 datasets, which include challenging crowded scenes with dynamic backgrounds and inter-person occlusions, multi-person interactions, and actions involving hard-to-detect daily use objects.

2 Related Work

We briefly survey classification methods using subsequence based models, and methods for activity detection.

[**Subsequence Models**] The discriminative advantage of short snippets of video for activity classification has been recognized before [21]. There exist approaches that identify the single most important subsegment of a video [21], while others represent the video as a sequence of manually annotated atomic actions [3], or a set of discriminative spatio-temporal patches [5], while ignoring their temporal ordering and distribution. [14,24] extended the deformable part object detector [2] to the temporal domain for activity classification, and decompose the video into discriminative sub volumes based on their correlations to a global feature distribution. The sub-volumes need not correspond to any semantic interpretation, and classify pre-segmented videos only. [15,25] propose a closely related model for learning discriminative key-pose sequences, with focus on interactions between a pair of humans. These models ensure only ordering constraints, but ignore the uncertainty in temporal placement of the poses and do not detect multiple instances of the activity in a video. [15] also manually defines a list of relevant poselets, where each poselet detector requires separate annotated training data, placing a practical limit on the range of poses it can model.

[**Activity Detection**] Structure models like in [11,13] perform automatic video segmentation by learning densely linked finite state machines, which combine all the activities in a single model. They do not scale well with the number of activities (inference is quadratic in the number of states), and require extensive manual annotations. Spatio-temporal volumetric feature based algorithms [6,29] rely on global statistics to detect action events, with no semantic reasoning of the underlying activity. [6] further requires enumeration of all possible sub volumes, and resorts to sub-sampling for tractable learning. There exist techniques [17,22] based on maximizing the volumetric correlation of 3D templates to localize single primitive actions, however it is unclear how multiple templates for complex activities can be combined.

3 Model Overview

We define a human activity as a sequence of key-poses. Pose inference is a difficult problem in itself; instead, we compute features that are related to pose but do not make the pose information, such as joint positions, explicit. We introduce a novel Pose Summarization Algorithm (PSA) to discover the key-poses in an activity during training, along with their expected temporal positions. Key-pose detection filters are discriminatively learned from the observed HoG-HoF features at the discovered key-pose locations. We also define a probabilistic temporal position distribution for each key-pose, to model its detection uncertainty.

The pose features in a video are quantized to a vocabulary of pose-codewords. The global distribution of the poses present in the sequence is learned using a *root filter*, which is a function of the histogram of pose-codewords. The multiple key-pose filters, root filters and their corresponding temporal relationships are jointly modeled in a probabilistic framework, resulting in an HCRF model. The parameters of this model are learned in a discriminative max margin framework using a latent support vector machine. Figure 1 shows the flow diagram of our proposed algorithm. Final classification and detection is performed by inferring the class labels, and temporal positions from the HCRF model.

While there exist methods [6,17,22,29] which perform detection in both space and time dimensions, we argue that spatial detection of the human is better solved by dedicated pedestrian trackers. We use trajectory results $\boldsymbol{x} = \{x_t\}$ from a standard tracking algorithm [4] as input to our activity detection framework, where x_t is the human detection box in the t^{th} frame. HoG/HoF [2] features $\boldsymbol{f}(x_t) \in \mathcal{R}^D$ are computed from the detection box x_t centered around the human, and hence the features capture the human pose configuration at time t.

4 Key-Pose Discovery

Automatic decomposition of an activity in a video segment into its constituent key-poses is defined as the Key Pose Discovery problem. This is a prerequisite for learning a key-pose detector, as we need to first discover what are the important key-poses in an activity, and determine their expected temporal position in an activity sequence, before learning how to detect them. Algorithms for automatic key pose discovery rely on variants of change detection in the pose dynamics [11], however they require accurate human limb estimates from motion capture data, which are difficult to obtain. Another approach is to perform hierarchical clustering of the pose features [31], followed by vector quantization to learn a vocabulary of pose based codewords. However these codewords do not take into account the temporal structure present in the pose sequence of an activity.

Inspired by existing techniques for video summarization [10], we solve the key pose discovery problem using *pose sequence summarization*. Given N poses in an activity sequence, our task is to select the $K < N$ subset of poses, which best summarize the complete pose sequence w.r.t. a cost function defined on the pose space.

4.1 Pose Summarization Algorithm (PSA)

Let $\boldsymbol{f}(x_t) \in \mathcal{R}^D$ define a D dimensional feature vector describing the human pose present in the window x_t. Let $\tau_1 \cdots \tau_K$ define the temporal location of the K key poses in an activity segment. By definition, each key pose $f(x_{\tau_k})$ best summarizes the poses present in the pose sequence $\left\{f(x_{b_k}) \cdots f(x_{b_{k+1}})\right\}$ present between frames b_k and b_{k+1}. Hence, for a given temporal range $[b_k, b_{k+1})$, the optimal key-pose location τ_k is computed as $\tau_k = \arg\min_{\hat{\tau}} C(\hat{\tau}, b_k, b_{k+1})$, where function $C\left(\hat{\tau}, b_k, b_{k+1}\right) = \sum_{t=b_k}^{b_{k+1}-1} \|\boldsymbol{f}\left(x_t\right) - \boldsymbol{f}\left(x_{\hat{\tau}}\right)\|_2^2$ is the Pose Summarization Error. The total cost incurred in summarizing the entire pose sequence using just K key-poses is given by the error function $E(K, \{\tau_k\}, \{b_k\}) = \sum_{k=1}^{K} C\left(\tau_k, b_k, b_{k+1}\right)$.

The optimal assignments of key-poses $\{\tau_k\}$, and their respective temporal boundaries $\{b_k\}$ are determined by minimizing $E(\cdot)$. A dynamic programming algorithm for video summarization was proposed in [10], which is easily adapted for our purpose. The key insight is that given the temporal boundaries $\{b_k\}$, the corresponding key-pose locations $\{\tau_k\}$ can be determined in $O(T^2)$ time for a video segment of length T. This suggests an algorithm which recursively determines the optimal temporal boundary locations. The dynamic program has

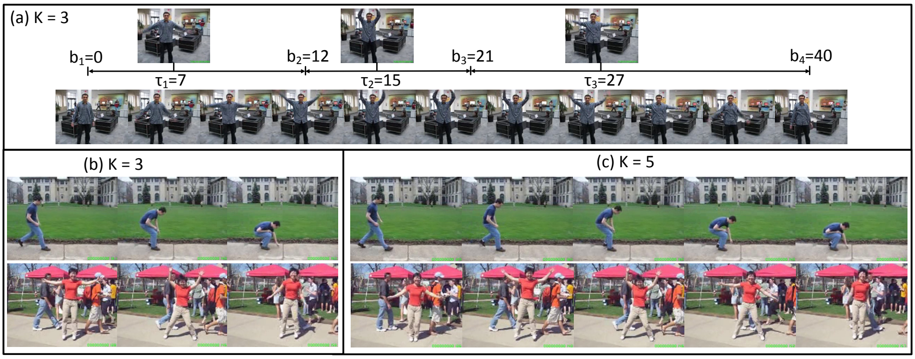


Fig. 2. (a) PSA discovers the key-poses for K=3 in a video segment. (b-d) Sample results for K=$\{3, 5\}$. (Detection boxes x_t are omitted for clarity).

a computational complexity of $O\left(KT^3\right)$, and hence is efficient for reasonably sized video segments with $T < 200$ frames.

We present results of the key-pose discovery on a sample video in Fig. 2, and observe that the discovered key-poses match closely to an intuitive definition of key-poses by humans. Note that with increasing K, adjacent key poses are more similar in appearance, and harder to distinguish from each other. The optimum value of K varies depending on the activity. The expected temporal location of the k^{th} key-pose in an activity segment is given by the temporal *anchor position* $\tau_k^a = N^{-1}\sum_n^N \tau_k^{(n)}$ computed over N videos. This is analogous to the anchor position of parts in object detection frameworks [2].

5 Pose Filter based HCRF Model (PF-HCRF)

We define temporal distributions to model the location of the key-poses relative to their anchor positions, and define an HCRF model to learn a set of discriminative key-pose detection filters by searching in the neighborhood of their corresponding anchor positions. Running inference on the HCRF model solves the detection and classification tasks simultaneously. HCRFs have been used before for part-based object and action classification [27]. We define the individual key-pose filters as the 'parts' in our HCRF model, resulting in a Pose Filter Hidden Conditional Random Field (PF-HCRF) model. Let y be a binary class variable signifying the presence/absence of an activity class. Our objective is to learn a distribution $P(y|\boldsymbol{x})$ to infer the class label y, given the trajectory $\boldsymbol{x}$:

$$y^* = \underset{y\in\{+,-\}}{\arg\max}\, P\left(y|\boldsymbol{x}\right) \propto \underset{y,\boldsymbol{z}}{\arg\max}\, P\left(y|\boldsymbol{z},\boldsymbol{x}\right) P\left(\boldsymbol{z}|\boldsymbol{x}\right) \qquad (1)$$

where $\boldsymbol{z} = \{t_r, t_1, t_2 \cdots t_K\}$ are the latent variables in our model. t_r determines the starting position of the action segment in the trajectory, while the variables $\{t_k\}$ determine the temporal location of the key poses constituting the activity. Figure 3(a) shows the factor graph representation of the PF-HCRF. Solving eqn.

1 provides us with localization of the activity segment in the trajectory, along with the class label. The key-pose locations also provide us with a description of the activity in terms of its key-poses. We model the probability distribution $P\left(y, \boldsymbol{z}|\boldsymbol{x}\right)$ using root $\boldsymbol{\theta}_{\boldsymbol{R}}^T\boldsymbol{\Phi}_{\boldsymbol{R}}$ and key-pose appearance $\boldsymbol{\theta}_{\boldsymbol{A}}^T\boldsymbol{\Phi}_{\boldsymbol{A}}$ filters, as follows:

$$P\left(y_{+1}|\boldsymbol{z}, \boldsymbol{x}\right) \propto exp\left\{\boldsymbol{\theta}_{\boldsymbol{R}}^T\boldsymbol{\Phi}_{\boldsymbol{R}}(\boldsymbol{x}, t_r) + \sum_{k\in\mathcal{K}}\boldsymbol{\theta}_{\boldsymbol{A_k}}^T\boldsymbol{\Phi}_{\boldsymbol{A}}(\boldsymbol{x}, t_k)\right\}$$

$$P\left(z|\boldsymbol{x}\right) \propto P(t_r|\boldsymbol{x})\prod_{k\in\mathcal{K}}P(t_k|\boldsymbol{x}) \propto \mathcal{C}\prod_{k\in\mathcal{K}}\mathcal{N}\left(t_k|\tau_k^a + \delta_k, \sigma_k^2\right)$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the standard normal distribution with mean μ and variance σ^2. The filters $\{\boldsymbol{\theta}_{\boldsymbol{R}}, \boldsymbol{\theta}_{\boldsymbol{A}}\}$ are a single dimensional template specifying the weights of the features $\{\boldsymbol{\Phi}_{\boldsymbol{R}}, \boldsymbol{\Phi}_{\boldsymbol{A}}\}$ appearing in a segment of the trajectory. Their dot product is the filter score when applied to the segment.

5.1 Root Filter

The root filter $\boldsymbol{\theta}_{\boldsymbol{R}}^T\boldsymbol{\Phi}_{\boldsymbol{R}}$ captures the global distribution of poses present in a given activity segment. First, a vocabulary of codewords $\mathcal{W}$ is learned over the pose features $f(x_i)$ extracted from all the videos in the training set. Then, each trajectory window x_i is assigned to the closest pose-codeword $w \in \mathcal{W}$, the mapping being defined by a function $g(x_t) : \mathcal{R}^D \rightarrow \mathcal{W}$. An activity segment is said to start from time t_r and has a length of L frames. The root filter computes the histogram of pose-code words (Fig. 3 c,d) present in the temporal window $[t_r, t_r + L]$, and is defined as:

$$\boldsymbol{\theta}_{\boldsymbol{R}}^T\boldsymbol{\Phi}_{\boldsymbol{R}}(\boldsymbol{x}, t_r) = \sum_{w\in\mathcal{W}}\eta_w\sum_{t=t_r}^{t_r+L}\mathbf{1}_{g(x_t)=w} \tag{2}$$

where $\mathbf{1}_{g(x_t)=w}$ returns 1 if $g(x_t)$=w is true, otherwise returns 0. Parameter L is the temporal bandwidth of the root filter, and is set to the average length of an activity segment determined from training examples.

5.2 Key-Pose Appearance Filter

Filter $\boldsymbol{\theta}_{\boldsymbol{A_k}}^T\boldsymbol{\Phi}_{\boldsymbol{A}}$ models the appearance of the k^{th} key-pose. Accurate key-pose detection requires the HoG-HoF descriptors to be computed from detections centered at the human figure. Misaligned detections (Fig. 4(c)) capture only the partial human image, and produce noisy HoG-HoF features, which in turn leads to inaccurate key-pose detections. We incorporate a scale-alignment search around the trajectory detection box $x_{t_k} = [\boldsymbol{c}, w, h]$, where $\boldsymbol{c}$ is the center of the detection box and (w, h) are its width and height:

$$\boldsymbol{\theta}_{\boldsymbol{A_k}}^T\boldsymbol{\Phi}_{\boldsymbol{A}}(\boldsymbol{x}, t_k) = \max_{s\in\mathcal{S}, \boldsymbol{p}\in\mathcal{P}}\gamma_k^T\boldsymbol{f}\left(x_{t_k} = [\boldsymbol{c} + \boldsymbol{p}, sw, sh]\right) \tag{3}$$

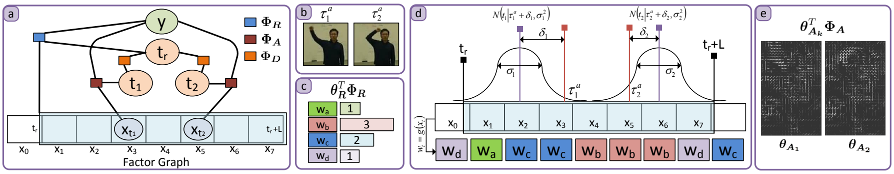


Fig. 3. Panel (a) shows the factor graph representation of the PF-HCRF model for $K = 2$ key-poses. Panel (b) shows the two key-poses identified by the pose summarization algorithm, with their corresponding anchor times: τ_1^a, τ_2^a. Panel (d) shows the feature descriptors x_t, and their corresponding codewords assignments $w_t \in \mathcal{W}$ below. The root filter $\boldsymbol{\theta}_{\boldsymbol{R}}^T \Phi_R$ is shown in cyan, being applied between frames t_r and $t_r + L$ where it models the pose-codeword frequencies, as shown in panel (c). Sample results of key-pose filters $\boldsymbol{\theta}_{\boldsymbol{A_k}}^T \Phi_A$ learned by the LSVM are shown in panel (e). Their temporal location is modeled with a normal distribution about their corresponding anchor location τ_k^a, shown in panel (d).

where $\mathcal{S}$ is the scale pyramid and $\mathcal{P}$ is the alignment search grid. We learn a conditional model, and hence the weight vector $\boldsymbol{\gamma}_k$ corresponds to the discriminative ability of the appearance of the k^{th} key-pose to classify the overall activity segment. Figure 3(e) show examples of appearance models learned for detecting key-poses. The weight magnitudes show a clear visual correlation with the discriminative key-pose present in the video.

5.3 Temporal Location Distribution

The latent variables $\boldsymbol{z} = \{t_r, t_1, t_2 \cdots t_K\}$ define the temporal location of the activity segment, and its constituent key-pose locations. As we do not have prior knowledge of the global temporal location of the activity segment, we set its distribution to a constant : $P(t_r|\boldsymbol{x}) \propto C$.

The temporal distribution of the key-poses $P(t_k|\boldsymbol{x})$ is modeled using a standard normal distribution $\mathcal{N}(\cdot)$ with mean $\tau_k^a + \delta_k$ and variance σ_k^2:

$$\log P(t_k|\boldsymbol{x}) \propto \boldsymbol{\theta}_{\boldsymbol{D_k}}^T \boldsymbol{\Phi_D} (t_k, t_r) \propto \log \mathcal{N} \left(t_k | \tau_k^a + \delta_k, \sigma_k^2 \right) \tag{4}$$

where parameter τ_k^a is the temporal anchor position (section 4.1) of the k^{th} key-pose, and remains unchanged during model training. The optimal key-pose locations τ_k in each video (determined by the pose summarization algorithm) need not be centered within their corresponding temporal boundaries $[b_k, b_{k+1})$ (see Fig. 2). Parameter δ_k accounts for this offset, and measures the linear shift in the key-pose location from its anchor position τ_k^a, while parameter σ_k^2 measures the uncertainty in the temporal location. Figure 3(d) shows the parameterization of the normal distribution. Both δ_k and σ_k are learned during model training, however it is more convenient to learn the equivalent log-probability parameters $a_k = 1/\sigma_k^2$ and $c_k = 2\delta_k a_k$.

6 Model Training

The HCRF model is trained using Max-margin criteria:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \Lambda : \forall_i \frac{\max_{\boldsymbol{z}^i} P(y^i, \boldsymbol{z}^i|\boldsymbol{x}^i;\theta)}{1 - \min_{\boldsymbol{z}^i} P(y^i, \boldsymbol{z}^i|\boldsymbol{x}^i;\theta)} > \Lambda \tag{5}$$

where Λ is the margin between the positive and negative examples. It has been argued [27] that the max-margin criteria is better suited for the classification task compared to the traditional Max Likelihood criteria. Solving equation 5 is equivalent to optimizing a Latent Support Vector Machine [28] in the log-probability domain. Transforming the probability distributions to the log domain results in the following energy function:

$$\begin{aligned} & E(\boldsymbol{x}, \boldsymbol{z}) = \log P\left(y = +1, \boldsymbol{z}|\boldsymbol{x}\right) = \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{z}) \\ & = \boldsymbol{\theta}_{\boldsymbol{R}}^T \boldsymbol{\Phi}_{\boldsymbol{R}}(\boldsymbol{x}, t_r) + \sum_{k \in \mathcal{K}} \left\{ \boldsymbol{\theta}_{\boldsymbol{D_k}}^T \boldsymbol{\Phi}_{\boldsymbol{D}}\left(t_k, t_r\right) + \boldsymbol{\theta}_{\boldsymbol{A_k}}^T \boldsymbol{\Phi}_{\boldsymbol{A}}(\boldsymbol{x}, t_k) \right\} \end{aligned} \tag{6}$$

6.1 Latent Support Vector Machine

A Latent Support Vector Machine (LSVM) incorporates latent variable inference in the SVM optimization algorithm. Yu et al [28] proposed a Concave-Convex Procedure (CCCP) for efficiently solving the LSVM optimization:

$$\begin{aligned} \min_{\boldsymbol{\theta}} & \left[\frac{1}{2} \|\boldsymbol{\theta}\|_2^2 + C \sum_{i=1}^{n} \max_{\hat{y}, \hat{\boldsymbol{z}}} \left[\boldsymbol{\theta}^T \Psi\left(\boldsymbol{x}_i, \hat{y}, \hat{\boldsymbol{z}}\right) + \Delta_L(y_i, \hat{y}, \hat{\boldsymbol{z}}) \right] \right] \\ & - \left[C \sum_{i=1}^{n} \max_{\tilde{\boldsymbol{z}}} \boldsymbol{\theta}^T \Psi\left(\boldsymbol{x}_i, y_i, \tilde{\boldsymbol{z}}\right) \right] \end{aligned} \tag{7}$$

where Δ_L is the loss function, and Ψ is the class augmented feature function. The optimization is solved using CCCP, which minimizes $f(\boldsymbol{\theta}) - g(\boldsymbol{\theta})$ where both f and g are convex. To map our activity model into the LSVM formulation while satisfying the convexity requirements of f and g, the feature function Ψ is defined as: $\Psi\left(\boldsymbol{x}_i, y_i, \tilde{\boldsymbol{z}}\right) = \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{z})$ for positive examples, and equal to 0 for negative examples. The CCCP algorithm requires solving two sub-problems iteratively: (1) Latent Variable Completion , and (2) Loss-Augmented Inference. Latent variable completion is equivalent to MAP inference on the HCRF model, defined as:

$$\max_{\tilde{\boldsymbol{z}}} \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{x}, \tilde{\boldsymbol{z}}) = \boldsymbol{\theta}_{\boldsymbol{R}}^T \boldsymbol{\Phi}_{\boldsymbol{R}} + \sum_{k \in \mathcal{K}} \max_{\substack{t_k \\ t_r = 0}} \left\{ \boldsymbol{\theta}_{\boldsymbol{D_k}}^T \boldsymbol{\Phi}_{\boldsymbol{D}} + \boldsymbol{\theta}_{\boldsymbol{A_k}}^T \boldsymbol{\Phi}_{\boldsymbol{A}} \right\} \tag{8}$$

where t_r is set to zero as training videos are pre-segmented. Maximization over t_k can be solved in $O(N)$ time (N is length of a trajectory) using distance transform

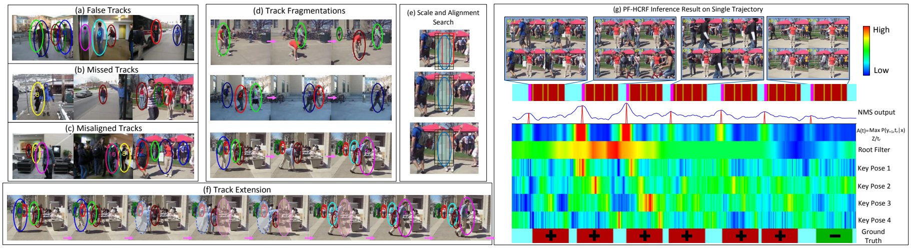


Fig. 4. (a-d) Noisy and erroneous tracks (e) Scale-Alignment search (f) Track extension (g) Heat maps represent the output scores of the key-pose filters, root filters and the inferred detection confidence $A(t)$, along with ground truth and predicted detection segments. Refer the text for more details. (Figure is best viewed in color and magnified)

algorithms [2]. The Loss Augmented Inference problem with zero-one loss for a binary decision problem is solved as follows:

$$\max_{\hat{y},\hat{\boldsymbol{z}}} \left[\boldsymbol{\theta}^T \Psi\left(\boldsymbol{x}_i, \hat{y}, \hat{\boldsymbol{z}}\right) + \Delta(y_i, \hat{y}, \hat{\boldsymbol{z}})\right] = \begin{cases} \max\left\{1, \max_{\hat{\boldsymbol{z}}} \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{x}, \hat{\boldsymbol{z}})\right\} & \text{if } y_i = +1 \\ \max\left\{0, 1 + \max_{\hat{\boldsymbol{z}}} \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{x}, \hat{\boldsymbol{z}})\right\} & \text{if } y_i = -1 \end{cases}$$

[**Weight Initialization**] LSVM optimization is non-convex, and careful initialization of the weights has been suggested in previous work [2,14]. We train standard SVMs separately on the root filter and the appearance filter features, and initialize $\boldsymbol{\theta}_R$ and $\boldsymbol{\theta}_A$ respectively to the learned weights. c_k is initialized using the mean displacement of the key-pose locations τ_k (obtained from Pose Summarization Algorithm) from the anchor position τ_k^a. Parameter a_k is initialized using the pose-boundary locations (b_k, b_{k+1}), as it represents the variance in the key pose location.

7 Model Inference for Multiple Detections

Detecting and tracking humans in cluttered and crowded environments is a challenging problem. We use a standard appearance based pedestrian tracker [4], trained independently of the datasets used here. Figure 4(a-d) shows some representative results highlighting the challenges. Common inaccuracies include false positive tracks, missed tracks, misaligned tracks, and track fragmentations. The PF-HCRF detector is less sensitive to false positive trajectories, and treats it as a valid human track where ideally no human activity will be detected. However, missed tracks are impossible to recover from; hence we prefer tracking algorithms with higher recall at the expense of precision. Misaligned tracks cause noisy key-pose detections, and hence we perform scale-alignment search (eqn. 3) around the detection box. Figure 4(e) shows the scale-alignment search about a candidate detection (blue ellipse), with the optimal box shown in red. In our experiments, we use a single octave scale pyramid $\mathcal{S}$ with 5 levels centered at the original scale, and a 3×3 alignment search grid $\mathcal{P}$ with 10 pixel step width.

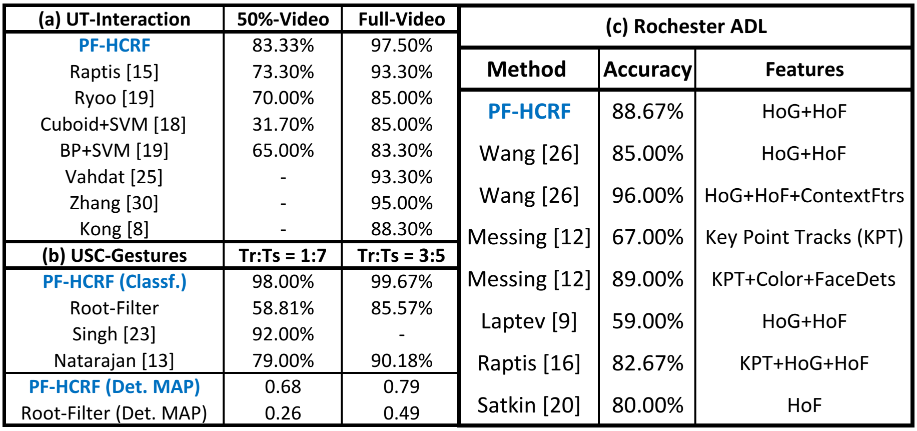
Track fragmentations are frequently caused by human subjects undergoing non-pedestrian pose transitions, which commonly occur during actions such as pickup. To counter the effects of premature track termination, we extend the trajectories beyond their start and end positions. Figure 4(f) shows an example of track fragmentation, and our proposed track extension (shaded-dashed detections in blue and magenta). We note that the extensions may not correspond to human subjects in the video (magenta colored extensions in figure), in which case they are equivalent to false positive tracks, and should not adversely affect our performance.

Detection and classification on a test video is performed by inferring the optimal class labels and root filter location: $\{y^*, t_r^*\} = \max_{y,\boldsymbol{z}} P(y, \boldsymbol{z}|\boldsymbol{x}; \theta^*)$. The optimal root filter location t_r^* is the detected position of the activity segment. For pre-segmented videos, t_r is fixed to zero, and only the optimal class label y^* is inferred. In an activity detection task, multiple instances of the same activity class can exist in a single video. The optimum t_r^* will return only a single detection result. To incorporate multiple detections, we infer the time series $A(t) = \max_{\boldsymbol{z}/t_r} P(y = +1, t_r = t|\boldsymbol{x}; \theta^*)$, representing the detection confidence at each time t. Following object detection algorithms, we apply a Non-Maxima Suppression (NMS) filter to $A(t)$, and declare the resulting maximas as our predicted activity detections.

Figure 4(g) shows an example of the multiple detection inference procedure for the two-handed wave action, where the outputs of the separate key-pose filters, root filters, and the inferred time series $A(t)$ are shown. The ground truth row shows the activity segments for two-handed wave action with positive labels (red), negative labels (green) for other actions and segments with no activity (cyan). The NMS output is given by pink bars, with the inferred key-pose locations marked in yellow, along with the key-pose frames shown above. The sequence of detected key-poses are consistent across segments and describe the activity. The video sequence contains action segments with partial occlusions, where some of the key-poses are not visible. We observe that the individual key-pose and root filter detection confidences are not sufficient for detecting the activity segments, whereas the combined inference result $A(t)$ provides a clear segmentation of the video, hence validating our algorithm. The NMS algorithm also detects a false positive due to the local maxima occurring in that segment; choosing an appropriate confidence-threshold for the detected maximas will remove the weakly scored false positives. We set the threshold to the confidence value corresponding to the maximum F1 score of each detector.

8 Results

We evaluate our algorithm on 4 datasets: UT-Interaction[18], USC-Gestures[13], CMU-Action[7] and Rochester-ADL[12]. The model is trained using a pose-codeword vocabulary size of 500, and by selecting an appropriate number of key-poses $K \in \{3, 4, 5\}$ based on action complexity. PF-HCRF model inference runs at 0.05 fps on a standard PC, and at 2 fps without scale-alignment search.



(a) UT-Interaction	50%-Video	Full-Video
PF-HCRF	83.33%	97.50%
Raptis [15]	73.30%	93.30%
Ryoo [19]	70.00%	85.00%
Cuboid+SVM [18]	31.70%	85.00%
BP+SVM [19]	65.00%	83.30%
Vahdat [25]	-	93.30%
Zhang [30]	-	95.00%
Kong [8]	-	88.30%
(b) USC-Gestures	**Tr:Ts = 1:7**	**Tr:Ts = 3:5**
PF-HCRF (Classf.)	98.00%	99.67%
Root-Filter	58.81%	85.57%
Singh [23]	92.00%	-
Natarajan [13]	79.00%	90.18%
PF-HCRF (Det. MAP)	0.68	0.79
Root-Filter (Det. MAP)	0.26	0.49

(c) Rochester ADL		
Method	**Accuracy**	**Features**
PF-HCRF	88.67%	HoG+HoF
Wang [26]	85.00%	HoG+HoF
Wang [26]	96.00%	HoG+HoF+ContextFtrs
Messing [12]	67.00%	Key Point Tracks (KPT)
Messing [12]	89.00%	KPT+Color+FaceDets
Laptev [9]	59.00%	HoG+HoF
Raptis [16]	82.67%	KPT+HoG+HoF
Satkin [20]	80.00%	HoF

Fig. 5. (a) UT-Interaction: Classification accuracy for observing the initial 50% of the video, and the full video. Result tables for (b) USC-Gestures and (c) Rochester-ADL.

8.1 UT-Interaction [18]

The UT-Interaction Set-1 dataset was released as a part of the contest on Semantic Description of Human Activities (SDHA) [18]. It contains 6 types of human-human interactions: hand-shake, hug, kick, point, punch and push. The dataset is challenging as many actions consist of similar human poses, like "outstretched-hand" occurs in point, punch, push and shake actions. There are 10 video sequences shot in a parking-lot, with 2-5 people performing the interactive actions in random order.

[Classification] SDHA contest [18] recommends using a 10-fold leave-one-out evaluation methodology. PF-HCRF achieves an average classification score of 97.50%, and outperforms all existing approaches (Figure 5a). We also evaluate our model on the streaming task (or activity prediction task), where only the initial θ fraction of the video is observable. This measures the algorithm's performance at classifying videos of incomplete activity executions. Figure 6a plots the classification accuracy for different values of observation ratio θ. The PF-HCRF model out-performs other methods, which can be attributed to its learning a small set of discriminative key-poses, where detecting even the first few key-poses helps in classifying the action. Moreover, the model returns the most likely position of key-poses in the unobserved section of the video, and hence is capable of "gap-filling". [15] also uses a key-frame based algorithm, however it is unable to perform gap-filling, as they only learn the temporal order of the key-frames, whereas PF-HCRF employs a probabilistic model for the key-pose locations, which is learned in a discriminative manner.

[Detection] SDHA contest [18] recommends evaluating the detection performance on the 10 videos using precision-recall curves for the 6 actions, and we present the same in Figure 6b. None of the contest participants report detection PR curves [18], making us the first ones to do so. [15] report detection results while assuming that each video contains one and only one instance of each action

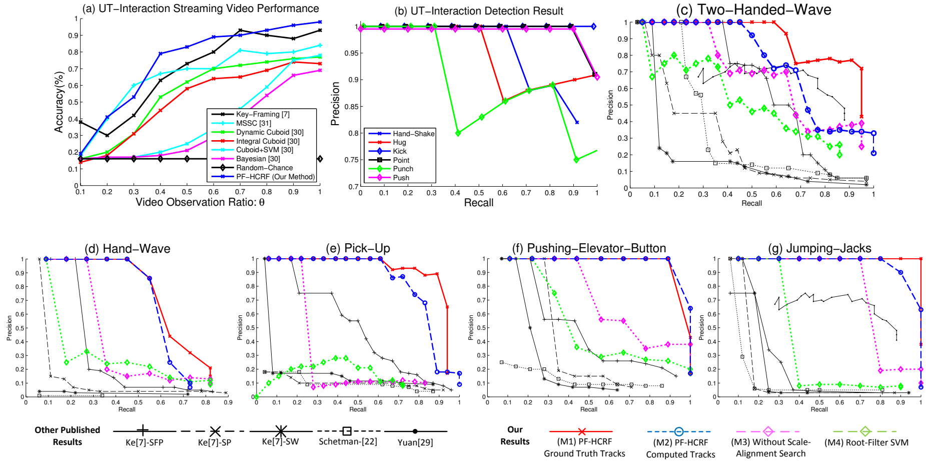


Fig. 6. UT-Interaction:(a) Streaming video performance and (b) Precision-Recall curves for activity detection. (c-g) Precision-Recall curves on CMU-Action.

type, and report an accuracy rate of 86.70% averaged over all actions, where the predicted action has a 50% temporal overlap with the groundtruth. Using the same metric, PF-HCRF model achieves an average detection accuracy of 90.00%.

8.2 USC-Gestures [13]

The dataset consists of 8 video sequences of 8 different actors, each performing 5-6 instances of 12 actions, resulting in 493 action segments. The actions correspond to hand gestures like attention, left-turn, right-turn, flap, close-distance, mount etc. The dataset has a relatively clean background with stationary humans, however it is still challenging due to relatively small pose differences between actions, causing pose-ordering to become a key discriminative factor in recognizing actions.

[Classification] Following [13,23], we evaluate the classification performance using two different train-test ratios: 1:8, and 3:5, averaged over all folds. The PF-HCRF algorithm outperforms previous results (Figure 5b) in all split ratios. Furthermore, [13,23] require manual construction of activity models using 2.5D joint locations for manually identified key-poses; the models also contain pre-defined motion styles and durations. PF-HCRF avoids cumbersome manual annotation of motion styles, while also automatically identifying the key-poses.

[Detection] The dataset has 8 videos (~12000 frames per video) containing continuous executions of 493 action segments. The action segments consist only ~10% of total video frames, and are interspersed with gesture actions other than the 12 gestures used for classification, making it a challenging dataset for activity detection. For baseline comparison, we implemented a Root-filter classifier (Sec.5.1), where a standard SVM is trained using the histogram of pose-codewords. Figure 5b shows the Mean Average Precision score averaged over 12 actions for the detection task. The root-filter does not capture temporal dynamics, and fails to differentiate between gestures with similar key-poses, but different temporal ordering, which explains their lower performance compared to PF-HCRF.

8.3 CMU-Action [7]

This dataset contain events representing real world activities such as picking up object from the ground, waving for a bus, pushing an elevator button, jumping jacks and two handed waves. The dataset consists of ~20 minutes of video containing 110 events of interest, with three to six actors performing multiple instances of the actions. The videos were shot using hand held cameras in a cluttered/crowded environment, with moving people and cars composing a dynamic background. The dataset is challenging due to its poor resolution (160×120), frequent occlusions, high variability in how subjects perform the actions, and also significant spatial and temporal scale differences in the actions.

We evaluate our performance using a 1:2 train:test split. Fig. 6 shows the Precision-Recall curves for the 5 action classes, and for four different method variations. First, PF-HCRF model is applied to manually annotated ground truth tracks (M1). Next it is applied to tracks computed from a pedestrian tracker [4] (M2), and then reapplied without "scale-alignment search and track extensions"(M3). Lastly, the Root-filter based SVM classifier is applied to the computed tracks (M4). We compare our performance to previously published results [7,22,29] on this dataset. Ke et al [7] show results using a flow consistency based correlation model of [22], and three variants of their super-pixel part-based method. Note, that these methods use only a single activity instance for training. Yuan et al [29] combine the two-handed-wave and jumping jack actions, and show results only on this single combined action.

Results on ground truth tracks (M1) provides an upper-bound on our performance in terms of reliance on tracks, and we achieve the best results using PF-HCRF across all actions. With computed tracks (M2), PF-HCRF still outperform other existing techniques across all actions, showing our model's tolerance to noisy tracks. Without "scale-alignment search and track extensions" (M3), the performance on hand-wave and pickup activities is poorer, which we attribute to misaligned and fragmented tracks caused by non-pedestrian poses, however, we still have good results for the other three activities. Lastly the results with the Root-filter classifier (M4) are significantly lower, which validates that our performance improvement is over and above simply using tracking results.

8.4 Rochester ADL [12]

The Activities of Daily Living (ADL) dataset contains 150 videos performed by 5 actors in a kitchen environment, and consists of 10 complex daily-living activities, involving interaction and manipulation of hard to detect objects: answering

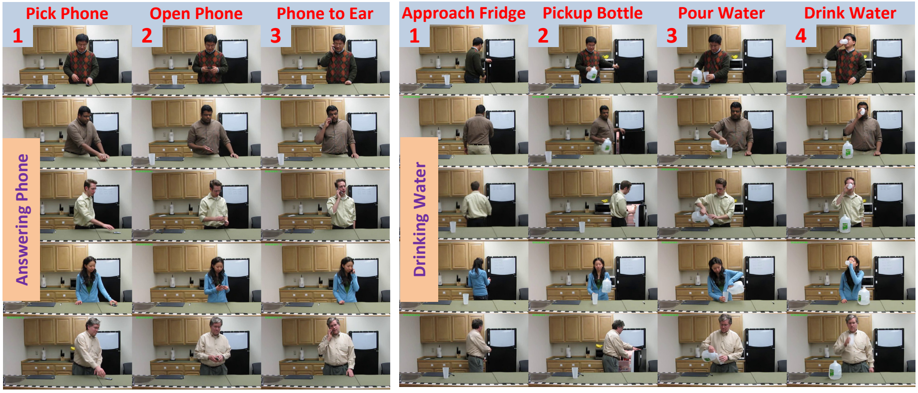


Fig. 7. Key-pose sequences inferred by PF-HCRF gives a semantic description of activity with high consistency

phone, dialing phone, looking up phone directory, writing on whiteboard, drinking water, eating snacks, peeling banana, eating banana, chopping banana and eating using silverware. As we do not have access to an upper-body tracker, the PF-HCRF is applied to the entire frame instead of tracks. Figure 5c summarizes our results on the dataset. PF-HCRF achieves an accuracy rate of 88.67% using only HoG-HoF features. The choice of features is important for this dataset, as special features can be designed to capture elements of the kitchen scene and the various objects, like yellow-banana, white-board, phone-near-face etc. Messing et al [12] augment their model with color and face-detection based features, improving their accuracy from 67% to 89%. Similarly, [26] augments their HoG-HoF descriptors with contextual-interaction based features, causing their accuracy to improve from 85% to 96%; we expect that the PF-HCRF model will also benefit from using simmilar contextual features. Furthermore, PF-HCRF localizes the key-poses of the complex activities, and we observe high consistency in the key-pose appearance across actors (Fig.7), and they seem to correspond to a natural semantic interpretation. Such decompositions are not obtainable using [12,26].

9 Conclusion

We proposed a key-pose filter based HCRF model for detecting multiple instances of activity in unsegmented videos, and generate semantic descriptions. We presented a novel pose summarization algorithm to automatically identify the key poses of an activity sequence. Our model training does not require manual annotation of key-poses, and uses video segment level class labels only.

Acknowledgments. We thank Dr. Remi Trichet and Dr. SungChun Lee for their valuable help and advice. This research was supported, in part, by the Office of Naval Research under grant #N00014-13-1-0493.

References

1. Cao, Y., Barrett, D.: Recognizing Human Activities from Partially Observed Videos. In: CVPR (2013)
2. Felzenszwalb, P., McAllester, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
3. Gaidon, A.: Actom sequence models for efficient action detection. In: CVPR (2011)
4. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
5. Jain, A., Gupta, A., Rodriguez, M., Davis, L.: Representing Videos using Mid-level Discriminative Patches. In: CVPR (2013)
6. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: ICCV (2005)
7. Ke, Y., Sukthankar, R., Hebert, M.: Volumetric Features for Video Event Detection. IJCV (2010)
8. Kong, Y., Jia, Y., Fu, Y.: Learning Human Interaction by Interactive Phrases. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 300–313. Springer, Heidelberg (2012)
9. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
10. Liu, T., Kender, J.R.: Computational approaches to temporal sampling of video sequences. MCCA (2007)
11. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching & viterbi path searching. In: CVPR (2007)
12. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
13. Natarajan, P., Singh, V., Nevatia, R.: Learning 3D Action Models from a few 2D videos. In: CVPR (2010)
14. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
15. Raptis, M., Sigal, L.: Poselet Key-framing: A Model for Human Activity Recognition. In: CVPR (2013)
16. Raptis, M., Soatto, S.: Tracklet Descriptors for Action Modeling and Video Analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 577–590. Springer, Heidelberg (2010)
17. Rodriguez, M., Ahmed, J., Shah, M.: Action Mach A spatio-temporal maximum average correlation height filter for action recognition. In: CVPR (2008)
18. Ryoo, M.S., Chen, C.-C., Aggarwal, J.K., Roy-Chowdhury, A.: An overview of contest on semantic description of human activities (SDHA) 2010. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 270–285. Springer, Heidelberg (2010)
19. Ryoo, M.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV. IEEE (2011)
20. Satkin, S., Hebert, M.: Modeling the Temporal Extent of Actions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 536–548. Springer, Heidelberg (2010)

21. Schindler, K., Van Gool, L.: Action Snippets: How many frames does human action recognition require? In: CVPR (2008)
22. Shechtman, E., Irani, M.: Space-time behavior-based correlation-Or-how to tell if two underlying motion fields are similar without computing them? PAMI (2007)
23. Singh, V., Nevatia, R.: Action recognition in cluttered dynamic scenes using Pose-Specific Part Models. In: ICCV (2011)
24. Tian, Y., Sukthankar, R., Shah, M.: Spatiotemporal Deformable Part Models for Action Detection. In: CVPR (2013)
25. Vahdat, A., Gao, B., Ranjbar, M., Greg Mori: A discriminative key pose sequence model for recognizing human interactions. In: Workshop on Visual Surveillance (2011)
26. Wang, J., Chen, Z., Wu, Y.: Action Recognition with Multiscale Spatio-Temporal Contexts. In: CVPR (2011)
27. Wang, Y., Mori, G.: Hidden Part Models for Human Action Recognition: Probabilistic vs. Max-Margin. PAMI (2010)
28. Yu, C.N.J., Joachims, T.: Learning structural SVMs with latent variables. In: ICML (2009)
29. Yuan, J., Liu, Z., Wu, Y.: Discriminative Subvolume Search for Efficient Action Detection. In: CVPR (2009)
30. Zhang, Y., Liu, X., Chang, M.-C., Ge, W., Chen, T.: Spatio-Temporal Phrases for Activity Recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 707–721. Springer, Heidelberg (2012)
31. Zhuang, Y., Rui, Y.: Adaptive key frame extraction using unsupervised clustering. In: ICIP (1998)