Animated Pose Templates for Modeling and Detecting Human Actions

Benjamin Z. Yao, Bruce X. Nie, Zicheng Liu, *Senior Member*, *IEEE*, and Song-Chun Zhu, *Fellow*, *IEEE*

Abstract—This paper presents animated pose templates (APTs) for detecting short-term, long-term, and contextual actions from cluttered scenes in videos. Each *pose template* consists of two components: 1) a shape template with deformable parts represented in an And-node whose appearances are represented by the Histogram of Oriented Gradient (HOG) features, and 2) a motion template specifying the motion of the parts by the Histogram of Optical-Flows (HOF) features. A shape template may have more than one motion template represented by an Or-node. Therefore, each action is defined as a mixture (Or-node) of pose templates in an And-Or tree structure. While this pose template is suitable for detecting short-term action snippets in two to five frames, we extend it in two ways: 1) For long-term actions, we animate the pose templates by adding temporal constraints in a Hidden Markov Model (HMM), and 2) for contextual actions, we treat contextual objects as additional parts of the pose templates and add constraints that encode spatial correlations between parts. To train the model, we manually annotate part locations on several keyframes of each video and cluster them into pose templates using EM. This leaves the unknown parameters for our learning algorithm in two groups: 1) *latent variables* for the unannotated frames including pose-IDs and part locations, 2) *model parameters* shared by all training samples such as weights for HOG and HOF features, canonical part locations of each pose, coefficients penalizing pose-transition and part-deformation. To learn these parameters, we introduce a semi-supervised structural SVM algorithm that iterates between two steps: 1) learning (updating) model parameters using labeled data by solving a structural SVM optimization, and 2) imputing missing variables (i.e., detecting actions on unlabeled frames) with parameters learned from the previous step and progressively accepting high-score frames as newly labeled examples. This algorithm belongs to a family of optimization methods known as the Concave-Convex Procedure (CCCP) that converge to a local optimal solution. The inference algorithm consists of two components: 1) Detecting top candidates for the pose templates, and 2) computing the sequence of pose templates. Both are done by dynamic programming or, more precisely, beam search. In experiments, we demonstrate that this method is capable of discovering salient poses of actions as well as interactions with contextual objects. We test our method on several public action data sets and a challenging outdoor contextual action data set collected by ourselves. The results show that our model achieves comparable or better performance compared to state-of-the-art methods.

Index Terms—Action detection, action recognition, structural SVM, animated pose templates

1 Introduction

1.1 Backgrounds and Motivations

Human action recognition has attracted increasing research interest in recent years motivated by a range of applications from video surveillance, human-computer interaction, to content-based video retrieval. Building a robust system for real-world human action understanding presents challenges at multiple levels: 1) localizing the actions of interest; 2) recognizing the actions; and 3) interpreting the interactions between agents and contextual objects. Recent research has made major progress on classifying actions under idealized conditions in several public data sets, such as the KTH data set [1] and Weizmann data set [2], where each video clip contains one person acting in front of static or uncluttered background with one action per video clip. State-of-the-art methods have achieved nearly 100 percent accuracy on these two data sets. On the other hand, action understanding from real-world videos, which commonly contain heavy clutter and background motions, remains a hard problem.

In general, actions are building blocks for activities or events, and thus are simpler than the latter in terms of the number of agents involved and the time duration, but still have diverse complexities in space and time:

1. In space, actions can be defined by body parts, such as waving and clapping, by human poses, such as walking, or by the human-scene interactions, such as making coffee in an office and washing dishes in a kitchen. In the last case, the whole image provides contextual information for action recognition.
2. In time, actions can be defined in a single frame, such as sitting and meeting, two to five frames such as pushing a button and waving hand which are also called action snippets [3], or a longer duration say in 5 seconds, such as answering a phone call.

In the following, we briefly review the literature in four categories according to their space-time complexity:

1. *Action recognition by template matching*. The idea of *template matching* has been previously exploited by researchers for action recognition. These approaches attempt to characterize the motion by looking at video sequences as

- *B.Z. Yao is with Beijing University of Posts and Telecommunications, China.*
- *B.X. Nie, and S.-C. Zhu are with the Statistics Department, University of California at Los Angeles, 8125 Mathsciences Bldg., Los Angeles, CA 90095. E-mail: {xhnie, sczhu}@stat.ucla.edu.*
- *Z. Liu is with Microsoft Research, One Microsoft Way, Building 113/2116, Redmond, WA 98052. E-mail: zliu@microsoft.com.*

Manuscript received 7 May 2012; revised 5 Jan. 2013; accepted 30 Mar. 2013; published online 2 Aug. 2013.
Recommended for acceptance by F. de la Torre.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-05-0351.
Digital Object Identifier no. 10.1109/TPAMI.2013.144.

either 2D templates or 3D volumes. For example, Essa and Pentland [4] built a detailed, physically based 2D templates of the face using optical flow features. Recognition is then done by directly pattern matching with the templates. Bobick and Davis [5] use motion history images that capture both motion and shape to represent actions. They have introduced the global descriptors motion energy image and motion history image, which are used as templates that could be matched to stored models of known actions. Efros et al. [6] also perform action recognition by correlating optical flow measurements, but they focuses on the case of low-resolution video of human behaviors, targeting what they refer to as "the 30-pixel man." Following this line of research, Gorelick et al. [2] extend the concept of template matching from 2D motion templates to 3D space-time volumes. They extract space-time features for action recognition, such as local space-time saliency, action dynamics, shape structures, and orientation. One common shortcoming of these types of template matching method is that they rely on the restriction of static backgrounds which allows them to segment the foreground using background subtraction. In our method, there is no need of this kind of foreground segmentation. Also, since all these method use rigid templates, rather than the deformable templates used in this paper, they are much more sensitive to appearance and view-point variations.

2. *Action recognition by spatiotemporal interest points (STIP).* To overcome the large geometric and appearance variations in videos of an action, researchers extracted and HoG and HOF features around them for action classification in the SVM framework [7]. Different approaches either pooled the features in a bag-of-word (BoW) representation [1], [8] or embraced a pyramid structure [9].

If we compare with the task of object recognition, it is worth noting that a quantum jump in performance has been achieved in the past few years when researchers departed from the BoW features and adopted the deformable part-based model (DPM) [10], especially for human detection in images [11], [12]. Thus we expect that a better representation for human action should extend the DPM to the temporal domain and capture some important information missed by the STIP representations. First, body parts should be explicitly modeled with rich appearance features, unlike the BoW methods that rely on a codebook of quantized features or "visual-words." The quantization of code books through K-mean clustering is often unreliable, largely because these clusters have varying complexities and dimensions. Second, spatial relations between parts should be represented to account for the human poses. Third, the motion information for each part and the poses as a whole should be represented to account for the transitions and, thus, temporal regularity in sequential poses.

The above observations motivated our animated pose template (APT) model, and we shall overview these aspects when we overview APT in the next section.

3. *Action recognition by pose estimation.* Encouraged by the relative success of human detection using the deformable part-based models, recently people have attempted to treat action recognition as a pose-estimation problem in a single image frame. Ferrari et al. [13] retrieved TV shots containing a particular 2D human pose by first estimating the human pose, then searching for shots based on a feature vector extracted from the pose. Johnson and Everingham [14] used a mixture of tree-structured poses. Yang et al. [11] used HOG features and support vector machines (SVMs) classifiers for action recognition, where they argue that it is beneficial to treat poses as latent variable in the SVM training because one action typically contains multiple poses. Using similar features, Yang and Ramanan [12] studied a mixture-of-parts model in for pose estimation. In this case, however, they use strong supervision of body parts and report that it works better than latent parts model. We believe that this discrepancy is most likely due to their different tasks.

However, these pose estimation work all use still images rather than videos, and thus, they do not capture the motion information in short or long time durations.

4. *Action recognition by scene context.* Many actions are defined not by poses but by the human-object interactions. Marszalek et al. [15] proposed to use scene background context for action recognition. Lan et al. [16] used the contextual information between a single person and a group for better action and group activity understanding. Most recent work studied joint models of body pose configuration and object locations, for example, Gupta et al. [17], Wang et al. [11], Yao and Fei-Fei [18], and Pei et al. [19].

But most of these work detect contextual objects through object recognition jointly with poses in static images, and the motion patterns are often not represented, except in Pei et al. [19].

In summary, the literature in human action recognition has made major progresses in various aspects; however, these work still have not covered the full spectrum of actions in space and time complexity.

1.2 Overview of Our Method

Motivated by the various shortcomings in existing methods and the need to cover actions of different space-time complexity, we present an *APT* model for recognizing short term, long term, and contextual actions from real-world videos. In the following, we overview our model and algorithm in comparison to the existing methods:

Short-term actions as moving pose templates (MPTs). Short-term actions or the so-called action snippets [3] refer to actions observed in three to five frames (0.2-0.3 seconds of video), and they contain rich geometry, appearance, and motion information about the action. Fig. 1 shows two examples for clapping and drinking. A more complex action is often composed of a sequence of action snippets. Fig. 2a shows three instances and each instance has three snippets. By clustering these snippets, which implicitly aligns them in time, we learn a dictionary of *moving pose templates*: one for each snippet shown in Fig. 2b.

A moving pose template consists of two components shown in Figs. 2c and 2d:

- *A shape template* having a root window (bounding box) covering a number of deformable parts whose appearance is modeled by the HOG features. Like the DPM model for human detection [10], the geometric relations between the parts are included in a Gaussian distribution.

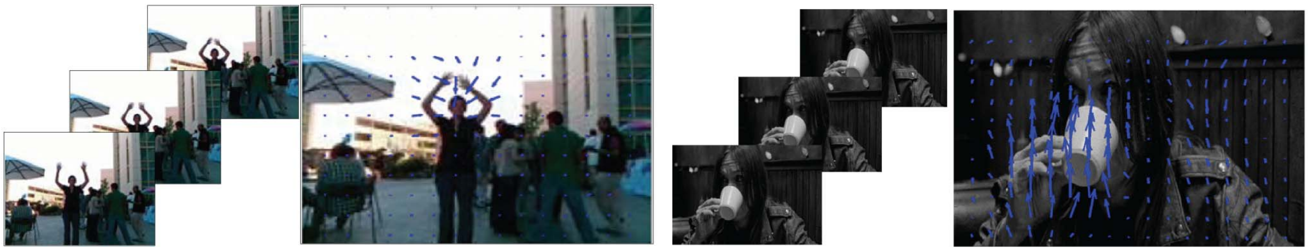


Fig. 1. *Action snippets* contain rich appearance, geometry, and motion information in three frames: 1) the static pose, 2) the short-term motion velocities (illustrated by blue arrows).

- *A motion template* specifying the motion of the parts by the Histogram of Optical-Flows (HOF) features. We compute motion velocities of parts by the Lucas-Kanade algorithm [20] to avoid the complexity of establishing temporal correspondence of parts between frames, since tracking body parts in cluttered video is a notoriously hard problem. The same shape template may have different motion templates, for example, in the clapping action, the raised arms could be moving upwards or downwards.

In comparison with the popular STIP representations, the moving pose templates represent the human geometry, appearance, and motion jointly and explicitly. Thus, it is a stronger model.

Long-term actions as animated pose templates. Long-term and continuous action, such as walking and running can be represented by a sequence of moving pose templates and we call it the animated pose template.

The term "animation" has been used in motion picture as a technique of rapidly displaying a sequence of images to create an illusion of continuous movement. The famous example is the galloping horse [21] created by Muybridge and is shown in Fig. 3. In this example, it is relatively easy to track the rider and the body of the horse over frames; however, it is hard or impossible to track (establishing correspondence between) the fast moving legs. The displacement of the horse legs is so large between two consecutive frames that conventional optical flow algorithms will fail.

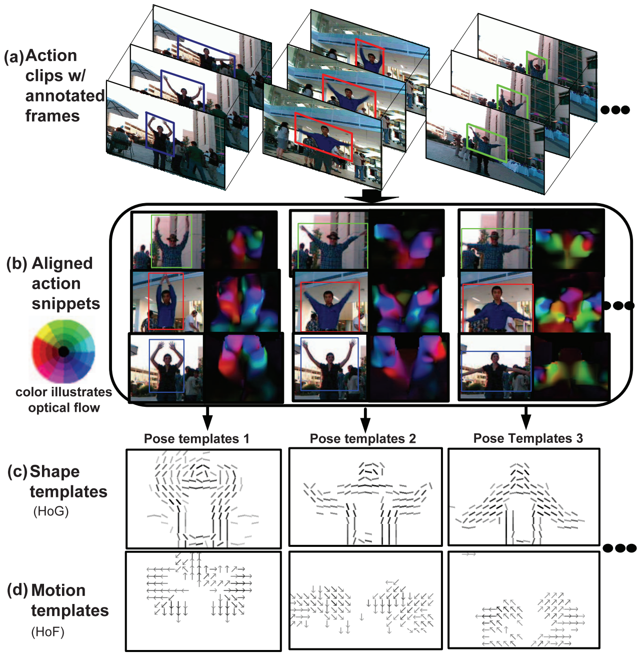

(a) Action clips w/ annotated frames
(b) Aligned action snippets
color illustrates optical flow
Pose templates 1
Pose templates 2
Pose Templates 3
(c) Shape templates (HoG)
(d) Motion templates (HoF)

Fig. 2. Moving pose templates. (a) Training action examples with annotated bounding boxes for the shape template, (b) three action snippets with optical-flow map (speed illustrated with color) for its parts, (c) the shape template has a root bounding box and a number of parts with HOG features, and (d) motion templates with HOF features for its parts.

Gong and Zhu [22] studied the issue of intrackability as a measure for the entropy of the posterior probability of velocity. They show that motion in a video can be partitioned as trackable motion, for which motion correspondence can be computed reliably, and intrackable motion, for which we need to compute a reduced or projected representation, such as a histogram of velocity in an area. The intrackability is affected by factors, such as object density, image scaling (sampling rate in space and time), and stochasticity of the motion.

Our animated pose template is a generative model based on the moving pose templates (or snippets):

- The shape templates between consecutive action snippets are considered trackable. So we will track the bounding boxes for the root node and its parts over time by Hidden Markov Model (HMM) model. The HMM model captures the spatial constraints on the movement of bounding boxes between frames and the transition between the type of pose templates (label of index in the pose template dictionary).
- The details inside each part are considered intrackable, and thus, we calculate the histogram of the flow without pixel level correspondence.

In our previous work [23], we animated a sequence of static active basis templates (sketches) without motion information or correspondence between parts. The current work is a step forward and the detection results show improved performance. The limitation of our model in this paper is our assumption that the human actions are viewed at a medium resolution and the motion information can be captured with Optical Flow (i.e., no dramatical displacements between frames). It is beyond the scope of this paper to study the multiresolution action representation.

Animated pose template with contextual objects. With the discovery of mirror neurons in monkeys [24], there are increasing work studying action embedded in the scene

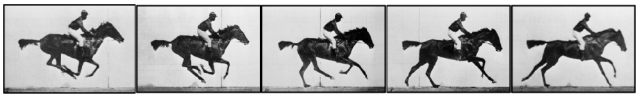


Fig. 3. Five video frames from E. Muybridge's "galloping horse" [21].

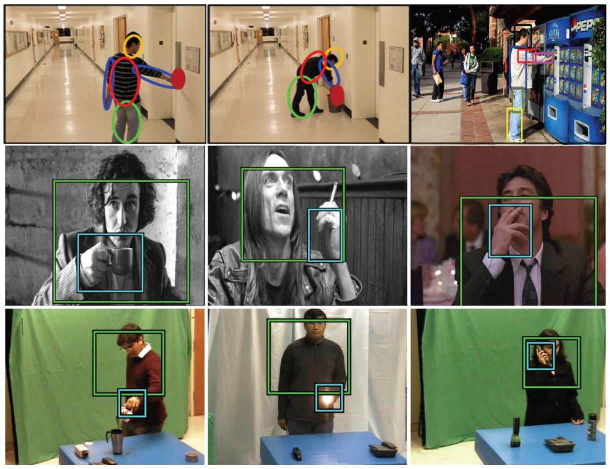


Fig. 4. Actions as interactions between an agent and contextual objects. Three examples from three action data sets that we use in the experiments.

contexts. Various research has demonstrated in vision that objects in scenes support important contextual information for human action understanding [25], [15], [18], and in return human action recognition helps identifying the objects [19].

Fig. 4 displays three examples 1) from the UCLA data set, 2) from the coffee and cigarette data set [26], and 3) from the CMU human-object interaction data set [17]. The objects are annotated with bounding boxes together with the body parts in training examples, such as the button, trash can, and venting machine in the UCLA data set. These boxes are added to the shape templates and, therefore, are trained in the same process.

In summary, our representation considers the complexities in geometry, appearance, motion, intrackability, and context in a coherent framework.

Effective inference by dynamic programming in space and time. Our representation benefits the inference process. Since the moving pose template has an And-Or tree structure and the animated pose template adopts HMM in temporal transition, we can use dynamic programming for fast inference in two steps:

1. Detecting top candidates for the moving pose templates for action snippets by dynamic programming; and
2. computing the animated pose templates with contextual objects by beam search in time using the top candidate moving pose templates. The beam search is a simplified version of dynamic programming as the number of possible candidate pose templates is enormous at each frame.

Furthermore, in Section 3 we show that a cascade algorithm can be used to further accelerate the computation to near real time.

Learning by semi-supervised structural SVM. To train the model, we collect annotated parts and contextual objects with bounding boxes on several keyframes of each video clips of the same action category, and cluster them into pose templates using EM algorithm. This leaves the unknown parameters for our learning algorithm in two groups: 1) *latent variables* for the unannotated frames including pose labels and part locations, and 2) *model parameters* shared by all training samples such as weights for HOG and HOF features, canonical part locations of each pose, coefficients penalizing pose-transition and part-deformation.

To learn these parameters, we use a semi-supervised structural SVM algorithm that iterates between two steps:

1. Learning (updating) model parameters using labeled data by solving a structural SVM optimization.
2. Imputing missing variables (i.e., detecting actions on unlabeled frames) with parameters learned from the previous step and progressively accepting high-scores frames as newly labeled examples. This algorithm belongs to a family of optimization methods known as the concave-convex Procedure (CCCP) that converge to a local optimal solution.

We test our method on several public action data sets and a challenging outdoor contextual action data set collected by ourselves. We demonstrate that our method is capable of discovering salient poses of actions as well as interactions with contextual objects. The result shows that our model achieves comparable or better performance compared to state-of-the-art methods.

Contributions. Our main contribution is a comprehensive model—APTs to represent human actions of diverse complexities in space (geometry, appearance, and context) and in time (short-term and long-term motion). This model accounts for the trackability of various components.

As the model is represented in an And-Or tree structure in space (short-term motion) and an HMM structure in time, it enables effective inference by dynamic programming in both space and time, and achieves near real-time solutions in cluttered videos, as well as results comparable to or better than the state of the arts in several challenging data sets.

Our model can be trained by the semi-supervised structural SVM framework for the large number of parameters. This framework has been successful in the object recognition task [27], [28].

This paper significantly extended our conference version [23], where each pose was represented by an active basis templates without parts, motion, or context objects.

The remainder of the paper is organized as follows: In Section 2, we introduce the formulations of animated pose templates. In Section 3, several inference strategies are introduced for detecting short-term action snippets and long-term actions. Section 4 presents the learning algorithm. In Section 5, we present the experimental results and comparison with other state-of-the-art methods on four public data sets and a contextual action data set we collected.

2 Representation

In this section, we present the formulation of the animated pose template model in three incremental steps: 1) moving pose templates for short-term action snippets; 2) APTs to account for long-term transitions between the pose templates; and 3) APT augmented with contextual objects.

2.1 Moving Pose Templates

Each action is composed of a sequence of key poses or action snippets, and the number of poses depends on the complexity of the action. For example, Fig. 5 displays three poses for a hand-clapping action from the MSR data set.

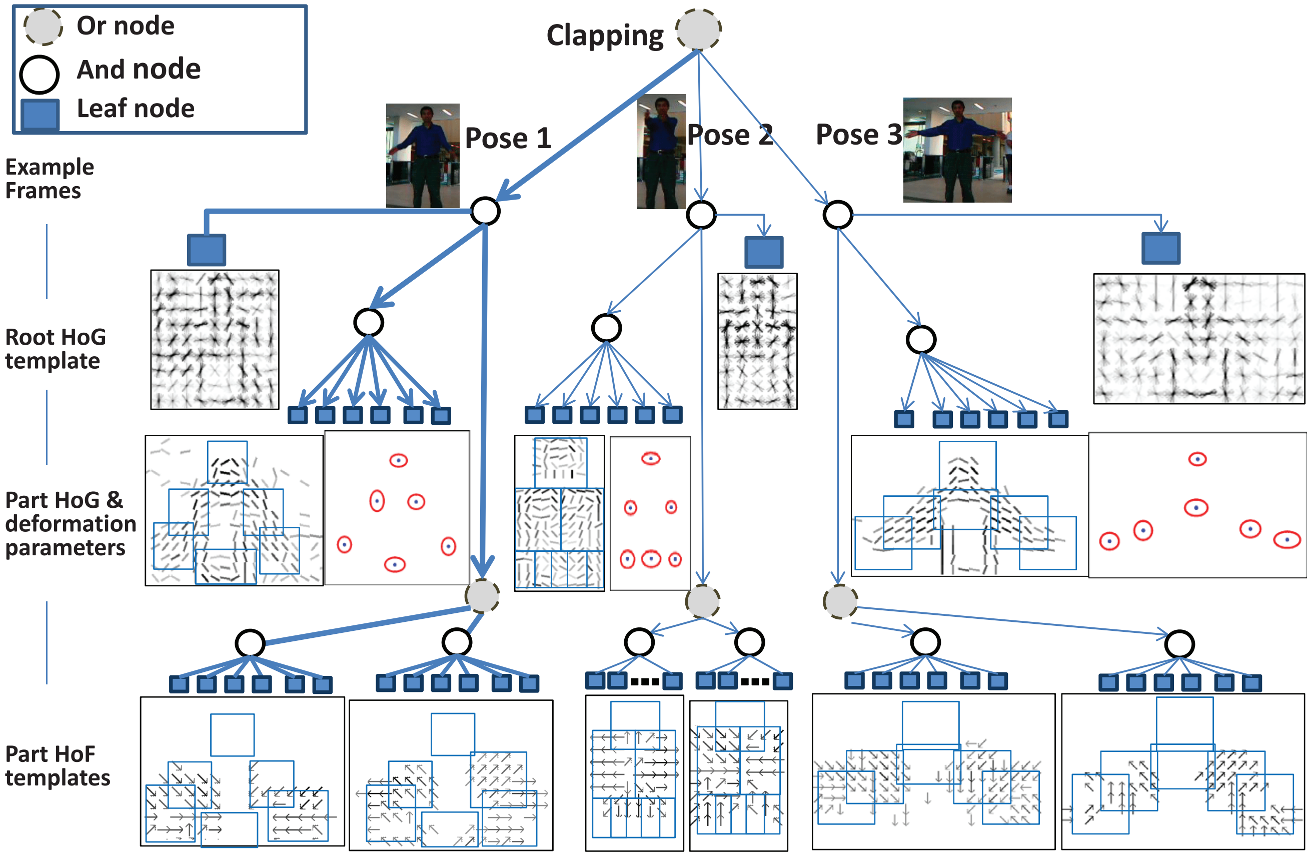


Fig. 5. A hand-clapping action includes six moving pose templates and is represented by a 2-level And-Or tree structure. Each moving pose consists of a shape template and a motion template. The shape template has one “root template” with HOG features for the entire bounding box at a coarse level and some (six in this case) “part templates” with finer scale HOG features. These part templates can deform with respect to the root template governed by 2D Gaussian functions whose mean and variance are illustrated by ellipses. A shape template can be associated with some (two in this case) motion templates for different movements of the parts and the motion of parts is represented by HOF features. We use small arrows representing the dominant flow directions. The three shape (static) templates multiplied by two motion templates represent six action snippets in the And-Or tree, where And-node represents conjunction and Or-nodes switches between alternative choices.

Each pose is represented by a shape template (denoted by **ST**) and one of the two motion templates (denoted by **MT**) depending on whether the arms are swinging upwards (or forwards) or downwards (or backwards). Thus, it generates a total of six moving pose templates organized in a hierarchical And-Or tree structure.

We denote these moving pose templates by a set

$$\Omega_{\text{mpt}} = \{\mathbf{MPT}_i = (\mathbf{ST}_i, \mathbf{MT}_i) : \ i = 1, \dots, n\}. \quad (1)$$

Each shape template $\mathbf{ST}_i$ consists of a root template $\mathbf{ST}_{i0}$ for the coarse level human figure and m templates $\mathbf{ST}_{ij}, j = 1, \dots, m$ for body parts:

$$\mathbf{ST}_i = (T_{i0}, T_{i1}, \dots, T_{im}). \quad (2)$$

This is similar to the DPM model in human detection [10]. Each template T_{ij} is a vector describing the geometry and appearance with the following components:

1. a_{ij} is the body label $a_{ij} \in \Omega_{\text{part}}$. $\Omega_{\text{part}} = \{``figure", ``head", ``torso", ``lefthand", ``righthand", \dots\}$. These parts are different in different actions. In training video clips, the bounding boxes for the root and parts are annotated in keyframes for each action. Fig. 6 shows these oriented boxes are then automatically transformed into rectangular boxes for easy computation of the HOG and HOF features.
2. Z_{ij} represents the window or image domain for the root or part templates. This includes the upper-left corner as anchor point, the window width, and height. To better align articulated body parts, we allow each part to rotate in -20 degrees, 0, $+20$ degrees. This is different from the DPM model.
3. X_{ij} represents the HOG vector extracted at window Z_{ij}. A dense grid of rectangular patches are defined on the feature pyramid a finite number of scales, and pixel-level features are aggregated by using a “soft binning” approach to obtain a cell-based feature map.
4. h_{ij} represents the vector of latent variables of the template, including its displacement $d_{ij} = (dx, dx^2, dy, dy^2)$ with respect to an anchor point and its rotation $d\theta$. Similar to DPM, we use a 2D quadratic function to penalize the displacements d_{ij}. The deformation is governed by 2D Gaussian functions whose mean and variance are illustrated by the red ellipses in Fig. 5. d_{ij} is written in a 4-vector to express the quadratic function in an inner product form. Rotation is not penalized.

The motion template $\mathbf{MT}_i$ is a concatenation vector for the m-parts:

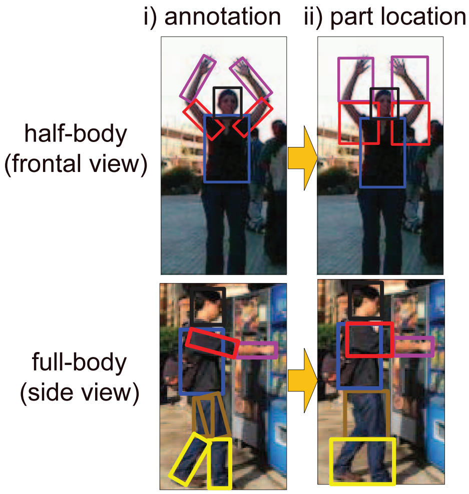


Fig. 6. From annotation of articulated limbs to rectangular parts. Column 1 shows the annotation results, which are orientated bounding boxes; Column 2 shows rectangular part windows derived from the annotation.

$$\mathbf{MT}_i = (V_{i1}, V_{i2}, \ldots, V_{im}). \quad (3)$$

V_{ij} represents the motion of each part. We use a variation of the HOF features [7]. We first compute an optical-flow map F using the Lucas-Kanade algorithm [20]. As this optical flow is not reliable due to intrackability [22], we pool the statistics over a small cell to form a histogram representation, similar to the HOF [7]. Each cell is of 8×8 pixels, and we choose eight directions evenly and project the velocity at each pixel to these directions to extract an eight-dimensional vector, which is then accumulated over all pixels at each cell. The result value is rescaled into $[0, 1)$ using a sigmoid transform. V_{ikj} is a concatenated vector for all the cells in the part j. Our design is slightly different from the HOF feature, which has four bins for four different directions and one bin for static (absolute speed below a threshold) [7]. We find empirically that our construction with finer orientations and continuous scale works better. For computational efficiency, when constructing a HOF feature pyramid, we only compute optical flow maps at three octaves of the image pyramid, the feature maps of the rest of the scales are obtained via interpolation.

Certain parts, such as the head and torso, do not have motion vector in the clapping action. Their features will be all zeros following the above HOF construction.

In summary, a hypothesized moving pose template $\mathbf{mpt}^{(t)}$ at a certain time frame is a long vector with the following variables: its label $\ell^{(t)} \in \{1, 2, \ldots, n\}$, geometry $\{Z_{ij}^{(t)} : j = 0, \ldots, m\}$ in the feature pyramid, deformation $\{d_{ij}^{(t)}: j = 1, \ldots, m\}$, appearance $\{X_{ij}^{(t)} : j = 0, \ldots, m\}$, and motion $\{V_{ij}^{(t)} : j = 1, \ldots, m\}$. The model of each template $\mathbf{MPT}_i$ includes an equal length parameter ω_i for its appearance, deformation, and motion features:

$$\omega_i = (\omega_i^A, \omega_i^D, \omega_i^M). \quad (4)$$

It will be trained through the structural SVM in the next section.

The hypothesis $\mathbf{mpt}^{(t)}$ is evaluated by a score function for $\ell = i$:

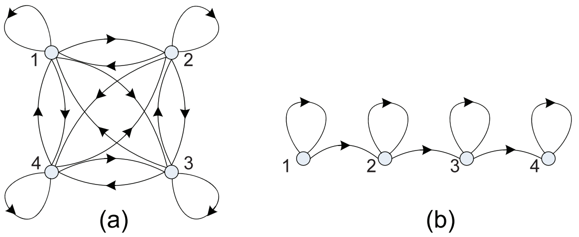


Fig. 7. We use two types of HMM for state transitions. (a) A 4-state ergodic HMM. The circles denote states (e.g., pose label in our paper). The arrows denote probabilistic transitions between states. (b) A 4-state left-right Hidden Markov Model.

$$S(\mathbf{mpt}^{(t)}) = \sum_{j=0}^{m} <\omega_{ij}^A, X_{ij}^{(t)}> + \sum_{j=1}^{m} <\omega_{ij}^D, d_{ij}^{(t)}> + \sum_{j=1}^{m} <\omega_{ij}^M, V_{ij}^{(t)}>. \quad (5)$$

The score function can be interpreted as a log-posterior probability up to a constant. The inference algorithm searches through all the possible $\mathbf{mpt}^{(t)}$ in a feature pyramid at time t for the highest score moving pose templates, and output a top candidate list.

2.2 Animated Pose Templates

An APT is a sequence of moving pose templates following a transition probability p, and we can write the possible APTs in a time interval $[t^s, t^e]$ by a stochastic set:

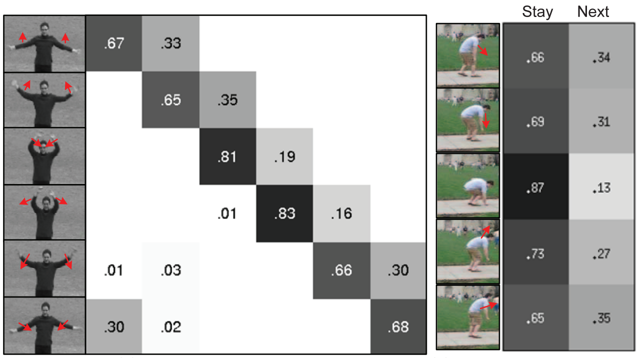
$$\Omega_{\text{apt}} = \left\{\mathbf{apt}[t^s, t^e] = \left(\mathbf{mpt}^{(t^s)}, \ldots, \mathbf{mpt}^{(t^e)}\right)\right\}. \quad (6)$$

The sequence is governed by the Markov chain probability:

$$p\left(\mathbf{mpt}^{(t^s)}\right) \prod_{t=t^s}^{t^e} p\left(\mathbf{mpt}^{(t+1)} \mid \mathbf{mpt}^{(t)}\right). \quad (7)$$

The initial probability $p(\mathbf{mpt}^{(t^s)})$ is considered uniform over all the MPT's in an action. The transition probability $p(\mathbf{mpt}^{(t+1)} \mid \mathbf{mpt}^{(t)})$ includes two components:

1. Transition probability $p(\ell^{(t+1)} \mid \ell^{(t)})$ for the MPT labels $\ell^{(t)}, \ell^{(t+1)} \in \{1, \ldots, n\}$. This is the classic HMM model. As Fig. 7 shows, we use two types of HMM model. The first is an ergodic HMM allowing for flexible transitions between any MPTs, and in practice this matrix is sparse. The second is a left-right HMM. Fig. 8 shows two examples for the two HMMs, respectively. For repeated actions, like waving hands, there are some discrepancies between different people and, thus, end up with occasional irregular transitions. For punctual actions, like picking up, pushing button, which have a strict order, the probability of staying at the current state determines the speed of the action. This can be affected by the sampling rate.
2. Tracking probability $p(Z^{(t+1)} \mid Z^{(t)})$ for the movement of parts between frames. We assume that given the root template position and scale, the parts are conditionally independent, as Fig. 9 illustrates. Therefore, the probability is factorized:



(a) Egrodic transition matrix (b) Left-right transition weight

Fig. 8. Left: Transition matrix of the "waving hand" category which has six moving poses displayed alongside each row (state). For clarity, we only show nonzero transition probabilities. Right: The transition matrix for another action category: "picking up" in a left-right HMM.

$$p\big(Z^{(t+1)} \mid Z^{(t)}\big) = \prod_{j=0}^{n} p\big(Z^{(t+1)}_{\ell^{(t+1)}j} \mid Z^{(t)}_{\ell^{(t)}j}\big). \qquad (8)$$

The tracking process does not account for matching the appearance features $X^{(t)}_{ij}$ and $X^{(t+1)}_{ij}$ within the window, and this window patching from $Z^{(t)}$ to $Z^{(t+1)}$ may not be consistent with the motion flow in $\mathbf{MT}^{(t)}$. Because parts may either move too fast or have no reliable feature to track as we explained about the intrackability issue.

As $p(Z^{(t+1)}_{\ell^{(t+1)}j} | Z^{(t)}_{\ell^{(t)}j})$ is a Gaussian probability for the position (x, y) and scale s changes, we can express the quadratic function by a linear size over a six-dimensional vector $\phi^{(t)}_{ij} = (dx_{ij}, dx^2_{ij}, dy_{ij}, dy^2_{ij}, ds_{ij}, ds^2_{ij})$.

Then, the logarithm of the transition probability $p(\mathbf{mpt}^{(t+1)} \mid \mathbf{mpt}^{(t)})$ can be written as a score function:

$$S\big(\mathbf{mpt}^{(t+1)} \mid \mathbf{mpt}^{(t)}\big) = A\big(\ell^{(t+1)} \mid \ell^{(t)}\big) + \sum_{j=0}^{n} <\omega^T_{\ell^{(t)}j}, \phi^{(t)}_{\ell^{(t)}j}>. \qquad (9)$$

In the above definition, A is the logarithm of $p(\ell^{(t+1)} \mid \ell^{(t)})$, and $\omega^T_{\ell^{(t)}j}$ is the six-dimensional parameter for the quadratic function, which will be learned through SVM training.

In summary, in the video an animated pose template hypothesis $\mathbf{apt}[t^s, t^e]$ has the following score, following (5) and (9):

$$S(\mathbf{apt}[t^s, t^e]) = \sum_{t=t^s}^{t^e} S(\mathbf{mpt}^{(t)}) + \sum_{t=t^s}^{t^e-1} S(\mathbf{mpt}^{(t+1)} \mid \mathbf{mpt}^{(t)}). \qquad (10)$$

The inference algorithm will search for the **apt** that maximizes this score, which is equivalent to maximizing the posterior probability except that the parameters are discriminatively trained in SVM.

2.3 Animated Pose Templates with Contextual Object

We augment the MPTs by adding contextual objects as extra parts. Let $\Omega_{\text{obj}} = \{\text{``}\mathit{ground}\text{''}, \text{``}\mathit{button}\text{''}, \text{``}\mathit{cigarette}\text{''}, \text{``}\mathit{cup}\text{''}, \ldots\}$ be the set of possible objects involved in the human actions. These objects can be divided in two subsets depending on the information flow between the object and pose:

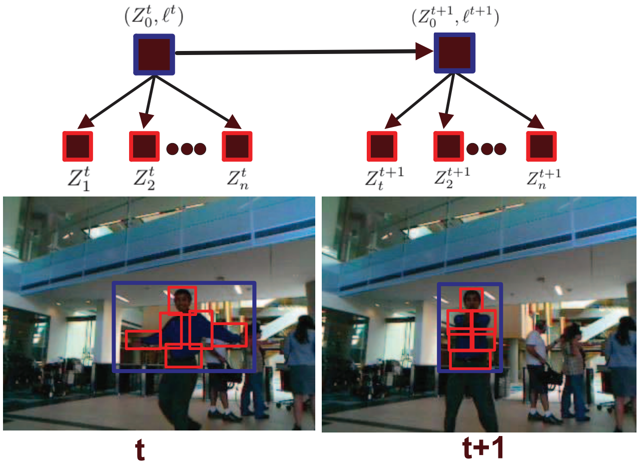


Fig. 9. The animated pose templates model assumes that parts are independent of each other given root template location and pose label. This assumption enables faster computation in our dynamic programming algorithm.

1. *Weak objects*. They are either too small or too diverse in appearance to be detected. Such as the cigarette, the ground. Therefore, action recognition provides contextual information for object recognition.
2. *Strong objects*. They are more distinguishable on their own, such as cup, torch when it is turned on, and trash can. Detecting such objects helps action recognition.

For example, Fig. 10a illustrates an instance of an action that consists of three MPTs sequentially: walking on the "ground" to approach the vending machine, pushing "button," and picking up the merchandise at the "outlet."

We treat these objects in the same way as the body parts in the MPT models, except that they do not have motion vectors. Suppose m' contextual objects are involved in a moving pose template $\mathbf{MPT}_i \in \Omega_{\text{MPT}}$. Then, these objects are added to the shape templates $\mathbf{ST}_i$ with new part templates:

$$\mathbf{CO} = (T_{im+1}, \ldots, T_{im+m'}). \qquad (11)$$

Each object template $T_{ij}, j = m+1, \ldots, m+m'$ has the following variables:

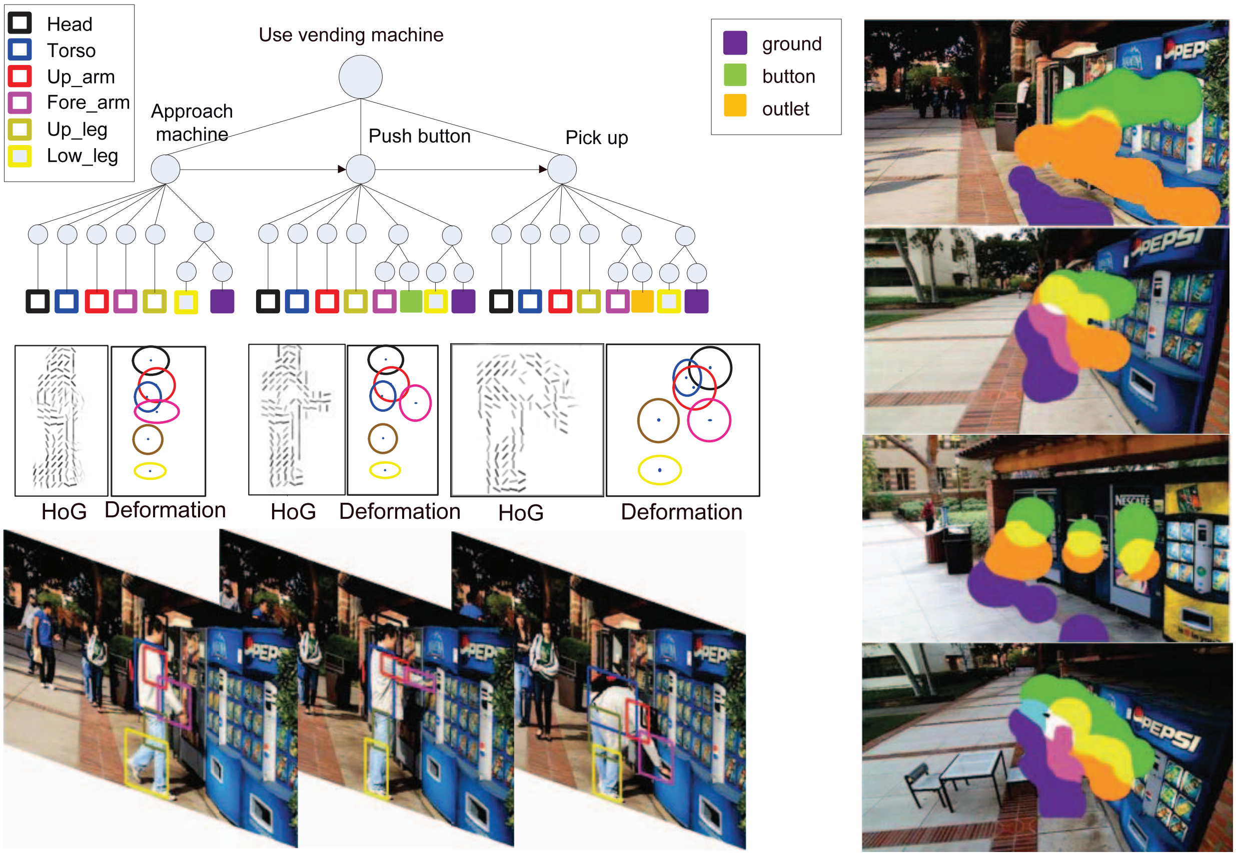
- $a_{ij} \in \Omega_{\text{obj}}$ for its name;
- Z_{ij} for the bounding box;
- X_{ij} for the HOG feature inside Z_{ij}; and
- d_{ij} for its deformation with respect to the interacting body part. For example, the button with respect to hand,

The labels and bounding boxes are annotated in keyframes of the training video. Thus, we add to the score function $S(\mathbf{mpt}^{(t)})$ by the following terms:

$$S(\mathbf{CO}^{(t)}) = \sum_{j=m+1}^{m+m'} <\omega^A_{ij}, X^{(t)}_{ij}> + \sum_{j=m+1}^{m+m'} <\omega^D_{ij}, d^{(t)}_{ij}>.$$

This score is added to (10) for inference and learning.

For weak objects, the inference process localizes them using the inferred body parts and the quadratic functions $<\omega^D_{ij}, d^{(t)}_{ij}>$ as constraints between the positions of the body part and object. Fig. 10b shows an example. After detecting



(a) Action recognition using contextual objects

(b) Semantic maps for contextual objects

Fig. 10. Contextual objects in action recognition. (a) The APT includes three sequential steps (or MPTs): walking on the "ground" to approach the vending machine, pushing "button" at the vending machine, and picking up the merchandise at the "outlet." The open squares are the body parts and we add three new nodes in solid squares for the contextual objects: ground (in purple), button area (green), and outlet (yellow). These objects have spatial relations with the low-leg, forearm nodes. In the second row, we show the learned HOG templates parts and their typical deformations by the ellipses. The bottom row of the figure shows the actual detection results of body parts on a video sequence. (b) The semantic maps for contextual objects generated by detected actions and person's body parts. Different colors indicate body parts and the objects that interact with them. The purple region is where feet are detected, and thus implies a standing point on the *ground*. The green region is where forearms are detected in a "pushing button" MPT, and thus implies a button. The yellow region is where forearms are detected while the person is in a picking-up MPT, and thus implies the outlet.

some purchasing actions in these scenes, we can derive the various contextual objects using different colors. The purple region is where feet are detected, and thus implies the *ground*. The green region is where forearms are detected in a pushing button MPT, and thus implies buttons. The yellow region is where forearms are detected while the person is in a picking-up MPT, and thus implies the outlets of the vending machines.

For strong objects, such as the cup in the drinking action in Fig. 11, the HOG feature is helpful in discriminating many other human activities.

3 Inference

3.1 The Objectives of Inference

Our inference integrates the following tasks in a single framework by dynamic programming in space and time:

1. Detecting the action snippets in every frame t by fitting the best moving pose templates $\mathbf{mpt}^{(t)}$. This includes localizing the parts $Z^{(t)}$ and classifying its class label $\ell^{(t)}$. As a by-product, the inference also predicts the locations of contextual objects and aligns the training examples in time.
2. Recognizing the animated pose templates in certain interval $[t^s, t^e]$ based on the moving pose template scores and the transition probability.

   All these variables are included in **apt**, and thus, the objective is to optimize the score functions in (10) for a given time interval $[t^s, t^e]$:

   $$\mathbf{apt}^*[t^s, t^e] = \arg\max S(\mathbf{apt}[t^s, t^e]). \qquad (12)$$

   One may interpret the SVM-trained score function S as a log-posterior probability, but it is different from a log-posterior probability in the Bayesian framework because it does not explain the whole video in $[0, N]$ or the images not covered by the bounding boxes of the root node.
3. Detecting multiple actions in a video. When a long video includes multiple and co-occurring actions, the inference is supposed to segment the actions in space and time. However, in most of the videos, such complex segmentation issues do not occur. To detect multiple actions, we use a sliding window in an image pyramid to calculate the MPTs and a sliding

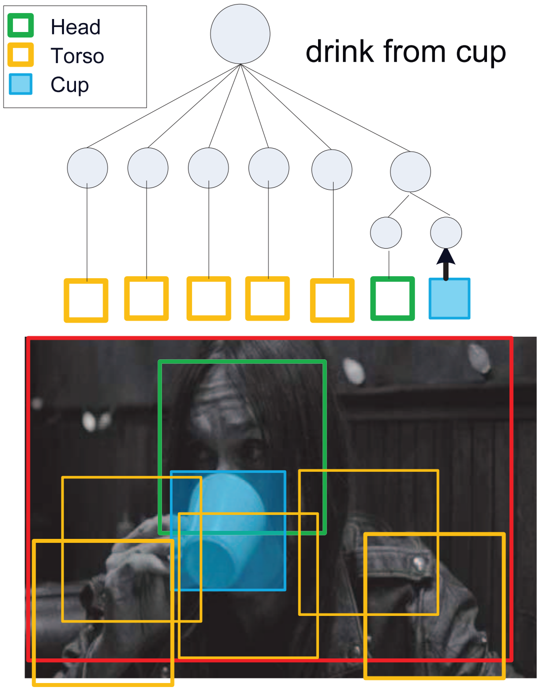


Fig. 11. A drinking action snippet (MPT) with a part for the head and five parts of the shoulder and chest. The cup is an additional part interacting with the head.

window in all time interval lengths, say between 3 and 60 frames, to calculate the APTs. Then, after selecting the APT with highest scores, we remove all the MPTs associated with it, and calculate the next best APT until no more APTs have scores above a certain threshold. Therefore, the approach is a greedy pursuit on top of the dynamic programming.

3.2 DP for Detecting MPTs in Space

We detect the action snippets for every three frames by maximizing the score functions in (5):

$$\mathbf{mpt}^{(t)} = \arg\ \max S(\mathbf{mpt}). \tag{13}$$

Given an image frame $I^{(t)}$, we build an image feature pyramid and extract the HOG features at 18 orientations and three octaves (10 scales per octave). This is similar to the deformable part-based model in the literature, and the difference is that we allow the part to rotate. We also need to calculate the HOF feature from three frames $I^{(t)}, I^{(t+1)}$, $I^{(t+2)}$ by the Lucas-Kanade [20]. We first compute two optical flow maps between frames $[I^{(t)}, I^{(t+1)}]$ and frames $[I^{(t+1)}, I^{(t+2)}]$, respectively, then we derive HOF features by pooling over an 8×8 pixels window in space and two optical-flow frames in time. Each HOF feature has eight bins representing eight directions, and the value of each bin equals to the projected sum of optical-flow vectors, modulated by a sigmoid function.

As the moving pose template is decomposed in an And-Or tree representation, it can be solved through standard Dynamic programming algorithm. We apply a cascade algorithm similar to the star-cascade detection method [29]. As this is standard in recent practices of detection with deformable parts model, we refer readers to the related literature [10] for additional details. The speed for detecting a single action class is about 2 frames per second (assuming that the optical flow maps are precomputed) for video resolution of (240×320) pixels using an i-7 PC.

3.3 DP for Detecting APTs in Time

At each frame, we output candidate MPTs through the dynamic programming algorithm sequentially. Each time the dynamic programming method finds a MPT with the highest score $S(\mathbf{mpt})$, it outputs the MPT as a candidate if its score is larger than τ_1. Then, we will block the window of the current candidate and find the next best MPT in the image until the best score is lower than τ_1. Threshold τ_1 is determined using the probably approximately admissible threshold strategy [29]. It is essentially the lowest score for detecting the presence of a person in the training set.

In a given interval $[t^{(s)}, t^{(e)}]$ scheduled by the sliding window detection process, suppose we have a series of MPTs detected over these image frames, and store them in an array of index:

$$\{\alpha(t) \in \{0, 1, \ldots, n^{(t)}\}, t \in [t^{(s)}, t^{(e)}]. \tag{14}$$

$n^{(t)}$ is the number of MPT candidates at frame t. These candidates are illustrated by the dots in Fig. 12 with color indicating the type of MPT.

Then, we compute the following two quantities:

1. The MPT score for candidate $\alpha(t)$:

$$s(\alpha(t)) = S(\mathbf{mpt}_{\alpha(t)}). \tag{15}$$

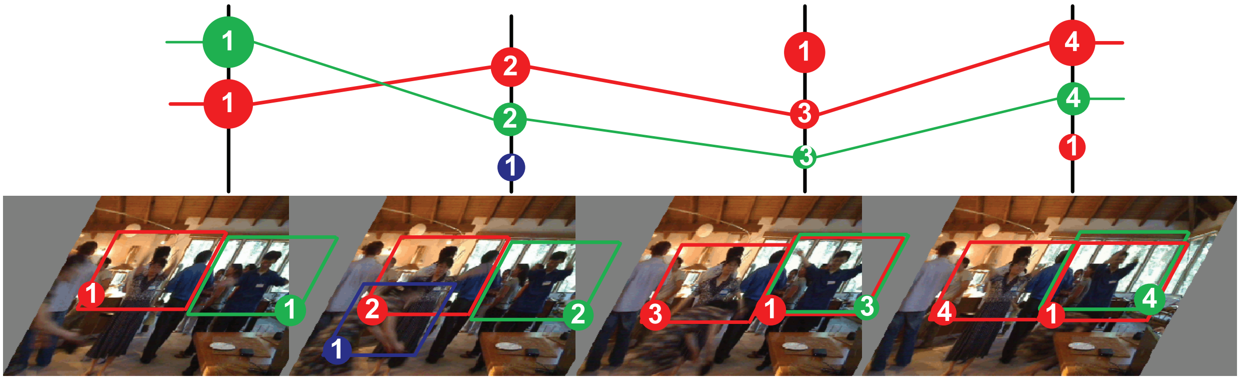


Fig. 12. Action detection with dynamic programming. Each frame has a number of action snippet candidates visualized with a number of colored balls. Green stands for "clapping," red stands for "hand waving," and blue is "boxing." Red and green lines represent two detected APTs.

If frame t has no detected MPTs (i.e., $n^{(t)} = 0$), we set $S(\alpha(t)) = -\inf$. We do not go back to adjust the moving pose templates based on temporal information, which could be very expensive and do not improve the results significantly.

2. The transition scores between any two candidates in consecutive frames:

$$s(\alpha(t), \alpha(t+1)) = S(\mathbf{mpt}_{\alpha(t+1)} \mid \mathbf{mpt}_{\alpha(t)}). \tag{16}$$

Thus, we can simplify the function in (12) by a standard DP program on finite state space:

$$\mathbf{apt}^*[t^s, t^e] = \arg\max \sum_{t=t^s}^{t^e} s(\alpha(t)) + \sum_{t=t^s}^{t^e-1} s(\alpha(t), \alpha(t+1)). \tag{17}$$

We outline the algorithm below.

Algorithm 1. Action detection algorithm.

```
Input: Video $I[t_i : t_j]$, thresholds $\tau_1$ and $\tau_2$;
Output: Detected actions in a list $List_A$;
  Perform MPT detection on each frame and add
  a candidate $\mathbf{mpt}_k$ to a candidate list $List_C$ if
  $S(\mathbf{mpt}_k) > \tau_1$ by (5)
  Connect all candidates at consecutive frames
  $\mathbf{mpt}_i^{(t)} \leftrightarrow \mathbf{mpt}_j^{(t+1)}$ of the same action type, and
  compute their transition cost $S(\mathbf{mpt}_j^{(t+1)} \mid \mathbf{mpt}_i^{(t)})$ by (9).
  repeat
    Find an optimal path apt using DP, and compute
    its $S(\mathbf{apt}) \Rightarrow s$ by (10).
    if $s < \tau_2$ then
      return $List_A$;
    else
      Remove all the MPTs in apt from $List_C$ and add
      them into $List_A$
    end if
until $List_C = \emptyset$.
return $List_A$;
```

To determine the optimal threshold for τ_2, we examine all positive training examples. For each action pose template $\mathbf{apt}_i$, we pick the highest value for threshold τ_2 so that it does not prune optimal configuration on the positive examples.

4 Learning

We adopt a semi-supervised Structured SVM method for learning the MPT and APT models. To be consistent with the SVM literature, we simplify the notation for clarity.

Suppose we have N training frames with n structured labels $y_i, i = 1, 2, \ldots, n$, where the first n frames are annotated with structured labels $y_i \in \mathcal{Y}$. y_i includes the MPT and APT labels and the bounding boxes for the roots and parts in these actions. The remaining frames have hidden labels $h_i \in \mathcal{Y}, i = n+1, \ldots, N$:

$$\mathcal{D} = \big(\{x_i, y_i\}_{i=1}^n, \{x_i, h_i\}_{i=n+1}^N\big). \tag{18}$$

x_i is the feature extracted from the ith example including the HOG and HOF features given the underlying window boxes Z_i.

The learning method proceeds in three steps in the following:

1. *Initializing the MPT and APT models.* We cluster the annotated frames into a dictionary of MPTs Δ_{mpt} (i.e., key poses) using EM. These MPTs correspond to different views and motion velocities of the moving pose templates. The algorithm is based on locations of annotated parts, and finds clusters in the joint space of HoG and HoF features.

   We also initialize the transition probabilities $A(\ell^{(t+1)} \mid \ell^{(t)})$ based on the cluster labels by counting the frequency of transitions.

2. *Training the MPT parameters by structured SVM.* Using the annotated frames and the MPT labels, we train the MPT parameters $\omega = (\omega^A, \omega^D, \omega^M)$ for appearance, deformation, and motion by Structured SVM. This is posed as a multiclass classification problem. Let function $\phi(x_i, y_i)$ denote a feature vector extracted from a frame i with MPT label y_i, and $\Delta(y_i, \hat{y}_i)$ a loss function, then the optimal parameters ω can be learned by minimizing the following function of Structured-SVM [30], [31]:

$$\begin{aligned} &\min_{\omega} \frac{1}{2}\|\omega\|_2 + \frac{C}{n}\sum_{i=1}^n \xi_i, \\ s.t.\quad &\max_{\hat{y}\in\mathcal{Y}} \omega^T(\phi(x_i, y_i) - \phi(x_i, \hat{y}_i)) \geq \Delta(y_i, \hat{y}_i) - \xi_i. \end{aligned} \tag{19}$$

   This optimization can be solved efficiently with a cutting-plane algorithm [30].

3. *Training the APT model by semi-supervised structural SVM (S^4VM).* In this step, we add unlabeled frames into the training process and incorporate the training of transition porbabilities between the MPTs. We define an upper bound function for the risk of the latent Structured SVM:

$$\begin{aligned} g(x, y; \omega) = max\{0, \Delta(y, \hat{y}) + \omega^T(\phi(x, \hat{y}) \\ - \max_{h^*} \phi(x, h^*))\}. \end{aligned}$$

   The latent SVM learning process optimizes the following objective function over ω:

$$\begin{aligned} \omega = \arg\min_{\omega\in\mathcal{R}^d} \Bigg(\frac{1}{2}\|\omega\|_2 + \sum_{i=1}^n g(x_i, y_i; \omega) \\ + \sum_{i=n+1}^N \max_{h\in\mathcal{Y}} g(x_i, h_i; \omega)\Bigg). \end{aligned} \tag{20}$$

   This function is a sum of convex and concave functions and can be optimized by the CCCP procedure iteratively.

In experiments, we find that if we add all unlabeled frames into the learning process at once, the results can go very bad. The difficult frames may have incorrect labels h_i that leads to wrong parameters ω and the algorithm can go into a downward spiral from here.

Inspired by the curriculum learning strategy introduced in [32], we add the unlabeled frames gradually. We add the frames with good scores using the current parameters ω, so that the structured labels h_i is close to the truth. This helps the algorithm converges smoothly to a good local minimum.

We now modify the above optimization problem by introducing binary variables v_i with $v_i = 1$ meaning that this sample will participate in the training. $v_i = 0$ will ban the frame in this iteration. Thus, the parameters ω is updated iteratively by the following mixed-integer program problem:

$$\omega_{t+1} = \arg\min_{\omega_t \in \mathcal{R}^d} \left[\frac{1}{2} \|\omega_t\|_2 + \sum_{i=1}^{n} g(x_i, y_i; \omega_t) + \sum_{i=n+1}^{N} v_i(t) \max_{h \in \mathcal{Y}} g(x_i, h_i; \omega_t) - \frac{1}{N} \sum_{i=n+1}^{N} v_i(t) \right].$$

Here, t indexes the iteration.

5 Experiment Results

5.1 Data Sets

We test our animated pose templates on five data sets, four of which are public data sets:

1. The *KTH* data set [1];
2. The *Microsoft Research Action II* (MSR) data set [33];
3. The *Coffee & Cigarette* data set [26]; and
4. The *CMU human-object interaction* data set [17].

The fifth data set is a contextual action detection data set we collected at UCLA campus.

Some of the public data sets have annotations, for example, the MSR and C&C data set have bounding boxes and labels for each action. But none of them include detailed annotations (i.e., location of body limbs at several keyframes) that are required by our algorithm. Therefore, we manually added annotations to all of the training data. The extra annotations and the UCLA data set used in this paper are available for download from our website: http://vcla.stat.ucla.edu/data set/animated_pose.html.

5.2 Action Classification on KTH Data Set

The KTH data set contains six types of human actions: walking, jogging, running, boxing, hand waving, and hand clapping. Each action is performed several times by 25 persons. We follow the standard experimental setting of KTH data set as in [1]. Among the 25 persons, 16 of them are used for training and the rest nine are used for testing. The training set contains 2,391 sequences. Since the data set has clean background and each video has one individual action from begin to end, it is easy to locate the actions of interest. Therefore, we only test the classification aspect of our model. To operationalize this, we learn APT for all action classes using the training set. On each testing video, we use Algorithm 1 to detect actions from the entire clip. There might be multiple detections in one clip because for certain action classes (e.g., jogging, running, etc.), the person goes outside of the image boundary between repetitions. The final classification of a video is the class that logs the longest time in all detections.

We use up to 40 annotated keyframes (evenly distributed in time) from each training video. Each keyframe has 10 annotated parts: head, torso, upper/lower arms, and upper/lower legs. For each action class, we cluster these keyframes into three poses, and the number of parts for the MPT model is 10, same as the numb of body parts. For comparison, we also test an APT model with latent parts (same as DPM [10]). The initialization procedure for this type of model is also down automatically in two steps: 1) clustering frames into poses with a k-means algorithm using motion cues; 2) initializing the parts using a method similar to DPM.

TABLE 1
Comparison on the KTH Data Set

Supervision	Work	Average
Weakly-supervised	Schuldt *et al.* [1]	71.71%
	Dollar *et al.* [8]	80.66%
	Niebles and Fei-Fei [34]	83.92%
	Laptev *et al.* [7]	91.81%
	Liu and Shah [35]	94.16%
	Kovashka and Grauman. [9]	94.53%
	Cao *et al.* [33]	95.02%
	Sadanand and Corso. [36]	**98.20%**
	APT with latent parts	84.70%
Semi-supervised	**APT, 10 annotated keyframes**	92.70%
	APT, 20 annotated key-frames	94.24%
	APT, 40 annotated key-frames	94.53%

Table 1 compares the accuracy of our method with the previous works on KTH data set using same experimental setting. The performance of APT with latent parts (and no additional annotation) is not very good, which we believe is mainly due to a bad local-minimum model caused by poor initializations. But with as few as 20 annotated keyframes, the full APT model yields performance that is among the best ones. It is also interesting to see that the effort of doubling the amount of annotated keyframes generates diminishing returns.

5.3 Action Detection on the MSR Data Set

The MSR data set includes 54 video sequences, each of which contains three types of actions, for example, hand waving, clapping, and boxing. These videos are taken with cluttered backgrounds, such as parties, outdoor traffic, and walking people. Actors are asked to walk into the scene, perform one of the three kinds of actions, and then walk out of the scene with these backgrounds. Each video clip is around 1 minute, while most action instances finish in 10 seconds. Throughout all the videos, people in the background are unconstrained, talking and walking. Unlike the KTH data set, there are multiple actions performed in each frame. Therefore, it is necessary to locate the action of interest from the scene. The original data set has annotations for bounding boxes and action classes.

A number of papers have reported action detection results on this data set such as [37], [33], [38]. Except [38], all the papers used cross-data set recognition setting where KTH data set was used for training while MSR data set was used for testing. Cao et al. [38] used a conventional setting where half of the videos were used as training and the other half were used for testing. They reported better results than the other papers since they do not use cross-data set setting. We follow the same experiment setting as Cao et al. [38], that is, we use half of the videos for training and the remaining half for testing. Since this data set is more interesting than the KTH data set, we present more detailed results here.

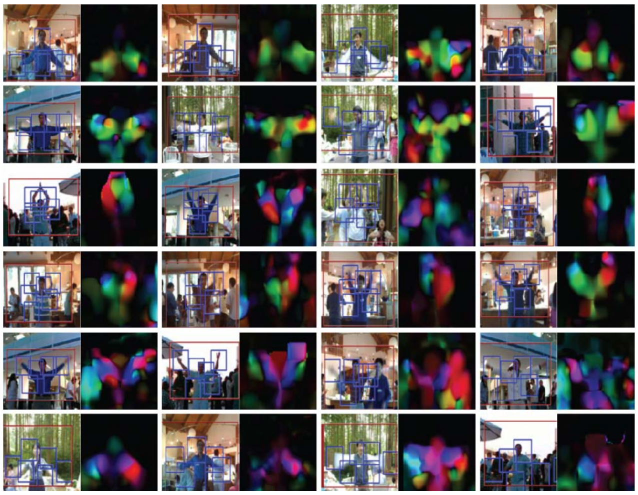


Fig. 13. Detection results using the moving pose templates on the "waving" category in the MSR action data set. Most of the poses are correctly identified with properly localized parts. Please see Fig. 2 for the illustration of optical flow directions.

For model initialization, we manually annotated all the frames of the training set (i.e., all the video clips with odd numbers) with body parts. Since all the actions in this data set are only upper-body related, we chose to annotate only six upper-body parts: head, torso, upper/lower left/right arm. The number of parts for the APT model is also set as 6.

Fig. 13 shows the detection results of moving pose templates on the "waving" category. Our method can correctly identify the pose, localize the parts, and estimate the velocity flows of each part. The motion is illustrated in colors in the same way as in Fig. 2. We take two criteria for quantitative performance comparison.

First, we treat the testing videos as action snippets and evaluate the testing performance using "False positive per image/Recall" criterion. The results are shown in Fig. 14a. The "boxing" class achieves the best performance because some poses of the "waving" and "clapping" classes are quite similar and, therefore, get confused with each other.

Second, we test action detection by searching over all possible combinations of starting and ending frames with 15 frames time step. For example, if the testing video has 60 frames, we test $[1:(2-60)], [2:(3-60)] \ldots, [58:(59-60)]$ frames, thus a total of 1,711 testing instances. The boxes bounding the action over many frames form a cuboid. Following the same criterion used by Cao et al. [38], we denote the cuboids of ground truth as $Q^g = \{Q^g_1, Q^g_2, \ldots, Q^g_m\}$, and the detected cuboids as $Q^d = \{Q^d_1, Q^d_2, \ldots, Q^d_n\}$. We use $HG(Q^g_i)$ to denote whether a ground-truth cuboid Q^g_i is detected, and $TD(Q^d_j)$ to denote whether a detected cuboid is correct:

$$HG(Q^g_i) = \begin{cases} 1, & \text{if } \exists Q^d_k,\ s.t. \dfrac{|Q^d_k \cap Q^g_i|}{|Q^g_i|} > \delta_1, \\ 0, & \text{otherwise}, \end{cases}$$
$$TD(Q^d_j) = \begin{cases} 1, & \text{if } \exists Q^g_k,\ s.t. \dfrac{|Q^d_k \cap Q^d_j|}{|Q^d_j|} > \delta_2, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where $|\cdot|$ denotes for the area of the cuboid, and δ_1, δ_2 are parameters to judge the overlapping ratio. Similar to [38], we set the δ_1 and δ_2 as $1/4$ in this paper.

Based on HG and TD, precision and recall can be defined as

$$\text{Precision} = \frac{1}{M}\sum_{i=1}^{M} HG(Q^g_i), \tag{22}$$

$$\text{Recall} = \frac{1}{N}\sum_{j=1}^{N} TD(Q^d_j), \tag{23}$$

where M is the number of ground-truth cuboids and N is the number of detected cuboids.

Given a collection of detected cuboids, we can compute the Precision-Recall curves based on (22)-(23). We then compute the area under curve (AUC) value for the three actions and compare our results with the best performing algorithm previously reported on this data set[1] [38]. The comparison results are shown in Fig. 14b. It is clear that APT model outperforms the previous work when full supervision is used. We further analyze the results by looking at the following aspects.

Amount of keyframes used. We test several different settings for the amount of annotated keyframes used for model learning. From the results, we find that more annotations lead to better detection performance, but the improvements are diminishing. The difference between 25 and 50 percent of annotations is much more significant than the differences between 50 and 100 percent.

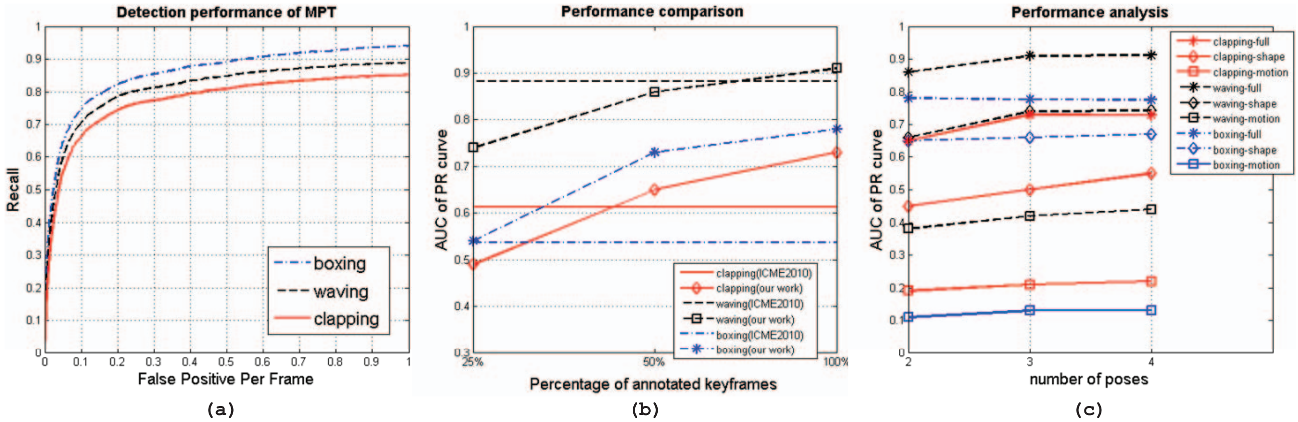
Number of poses. We investigate the sensitivity of APT method against the number of poses. As illustrated in Fig. 14c, reducing the number of poses to "2" hurts performances in both "waving" and "clapping' classes. But increasing the number to 4 or 5 (not illustrated in this figure) does not improve performances in all categories. It seems that "3" is a sweet point for the number of poses, therefore we use it as a default setting.

Contribution of "shape" and "motion" parts. We compares the performance of "full" APT model against models with only "shape" or "motion" parts in Fig. 14c. From the results, we notice that "shape" parts achieve better performance than "motion" parts, which is understandable because the resolution of HoF feature is rather low comparing with HoG features. It is also worth noting that for the "boxing" class "shape" parts are much more important than the "motion" ones. In fact with "shape" parts alone, our model achieves better performance on this class than previous methods.

Part localization accuracy. Since localization of body-parts is a by-product of the MPT model, we test its accuracy to shed some light into the performance of our model. For comparison, we choose a state-of-the-art method in [12], which has code online, as baseline. A few details for reproducibility:

1. For the baseline model, we use 17 parts (upper body) and five mixtures per part;
2. Because only half of the MSR data set has part annotations, we use a leave-one-out strategy for training and testing on the annotated videos;

[1] Courtesy of L. Cao, author of the previous work.



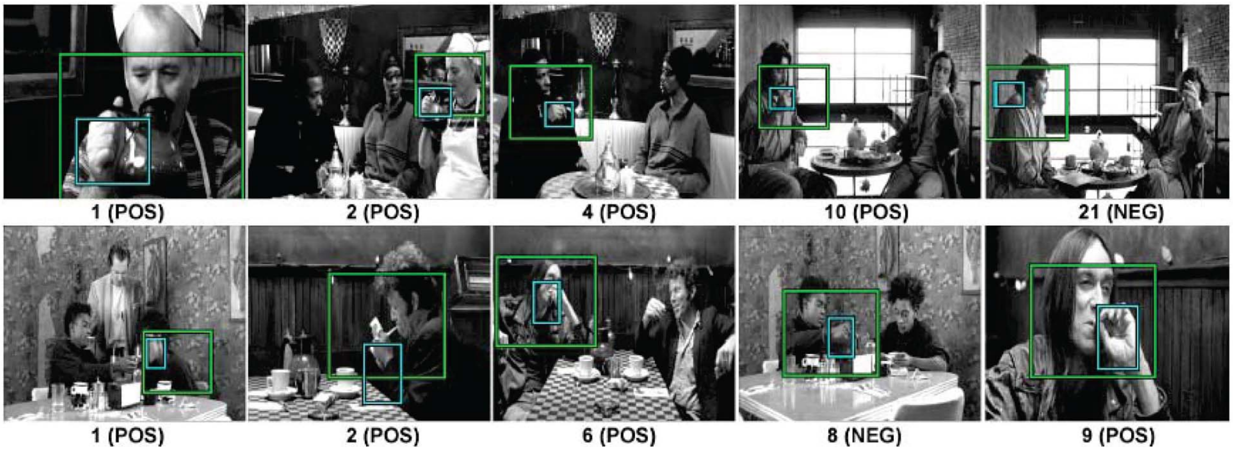
(a) (b) (c)

Fig. 14. Performance evaluations on the MSR data set. (a) Detection performance of three action snippets using MPT model. The “boxing” class is the best because some poses of the “waving” and “clapping” classes are easily confused with each other. (b) Performance comparison in terms of the area under Precision-Recall curve against the amount of annotated keyframes used to initialize training. Here, 100 percent means that all the training frames are keyframes. ICME2010 is the best previous method reported on this data set. (c) Performance against the number of poses used for each class. Comparisons between using full model and using only “shape” or “motion” parts are also included.

TABLE 2
Parts Detection Accuracy in Percent Using the Standard Criteria of PCP

Method	Head	Torso	U. arms	L. arms	Overall
baseline [12]	99.4, **99.6**, 96.2	**100**, **100**, **96.5**	87.6, 90.3, **73.5**	39.3, 48.7, **45.3**	75.6
MPT-shape	98.4, 98.6, 95.3	**100**, 96.5, 95.4	80.4, 85.2, 69.5	32.4, 38.5, 35.6	70.3
MPT-full	**99.6**, 99.4, **96.8**	**100**, **100**, 96.1	**89.5**, **91.2**, 70.4	**45.6**, **52.6**, 40.1	**76.2**

There are three numbers in each cell, which represent, from left to right, “clapping,” “waving,” and “boxing,” respectively.



1 (POS) 2 (POS) 4 (POS) 10 (POS) 21 (NEG)

1 (POS) 2 (POS) 6 (POS) 8 (NEG) 9 (POS)

Fig. 15. Detection performance of action snippets on the coffee and cigarette data set. We only show the bounding boxes for the human head and the boxes on the contextual objects: the hand holding a cup or cigarette.

3. When evaluating part localization accuracy, it is a convention to assume that person detection results (i.e., bounding boxes of persons) are given. In our case, since we are not interested in background persons, we use the cuboids from the ground truth as the bounding boxes;
4. The results are evaluated using a standard PCP criteria [39], which considers a part correct if its segment endpoints lie within 50 percent of the length of the ground-truth segment from their annotated locations.

The evaluation results are illustrated in Table 2. Each cell of the table shows three numbers, which are, from left to right, the performance of “clapping,” “waving,” and “boxing,” respectively. We test both MPT with “shape” parts and MPT-full. From the Table 2, we can see that the MPT-full model outperforms the baseline on average, and particularly in body limbs. The motion parts give the MPT model a noticeable boost. It is reasonable because that the limb parts have a lot of blurring, self-occlusion, and foreshortening. Therefore, they are very hard to detect with only shape information.

5.4 Coffee and Cigarette Data Set

The Coffee and Cigarette data set is collected mostly from the movie “Coffee and Cigarettes” and some training data are from a different movie named *Sea of Love* and a controlled lab video. It has 11 short stories each with different scenes and actors (See Fig. 15 for sample shots). This data set focuses on two action classes: “drinking” and “smoking.” For the drinking actions, there are 106 samples for training and 38 for testing. For the smoking action, there are 78 samples for training and 42 for testing. In each example,

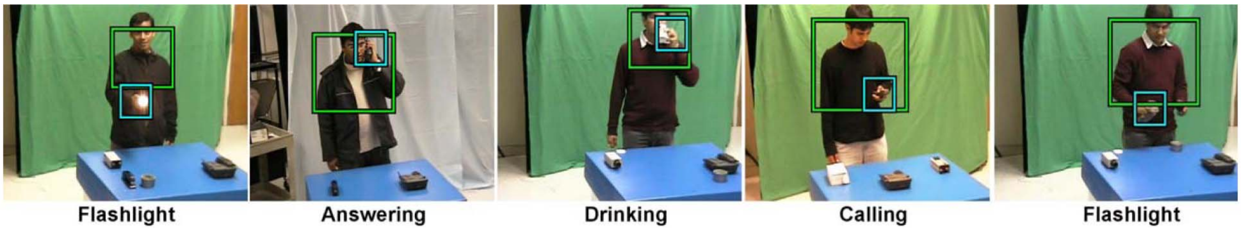


Fig. 16. Examples of detection results on the CMU data set.

we manually choose and annotate 40 keyframes (evenly spaced in time) with six upper-body parts and one contextual object (i.e., the hand holding a mug or a cigarette).

For this data set, we show some detection examples in Fig. 15. We adopt the evaluation protocol used in [26]: an action is correctly detected if the predicted spatiotemporal volume detection overlaps at least 20 percent with the ground-truth volume (or cuboid). Let $\mathcal{A}$ be the annotation cuboid for an event as ground truth. Our method outputs two cuboids $\mathcal{H}$ and $\mathcal{O}$, respectively. The overlap between $\mathcal{H}$ and $\mathcal{A}$ is given by $(\mathcal{A} \cap \mathcal{H})/(\mathcal{A} \cup \mathcal{H})$. We do not calculate the overlaps for the objects as the object cuboids often lay within the human cuboids. Table 3 reports the average precision (AP) of our methods and previous work [26], [40], [41].

We show three results: one uses the MPT without temporal information; one uses the APT with annotated contextual parts, and the other use the APT without manually selected parts (similar to DPM, locations of parts are treated as latent variables and are initialized with a heuristic procedure). The table shows that our method achieves better performance on this challenging task. Also, we observe that APT with latent parts performs worse than APT with contextual parts. The reason we believe is that with strong supervision on the contextual parts, the SVM algorithm is forced to learn a better template for the objects from roughly aligned examples. Without supervision, the information of contextual objects is likely to get lost during the learning process because these objects have bigger variations in terms of location and appearances than other body parts.

5.5 CMU Human-Object Interaction Data Set

The CMU human-object interaction is comprised of 60 videos with 10 actors performing six different actions, i.e., drinking from a cup, spraying from a spray bottle, answering a phone call, making a phone call, pouring from a cup, and lighting a flash light. Some examples are shown in Fig. 16. For each action, the videos are split into five training and five test videos. Unlike the C&C data set, these videos are shot in controlled conditions inside a laboratory with a static camera and a static background of uniform color. Like KTH, each video sequence has one individual action from beginning to end. We train an action classifier for each of the six actions using the training videos from the other classes as negative examples. Similarly to the C&C data set, we manually annotated 40 keyframes of each training example with six upper-body parts and one contextual object.

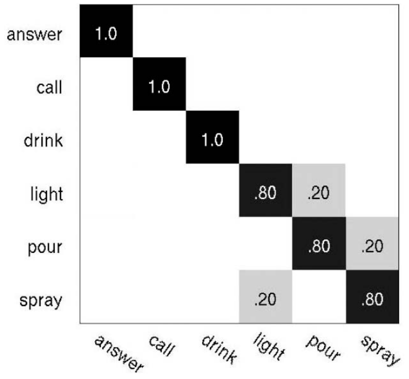
Given a test video, we evaluate the scores for the six actions and return as class label the one with the highest score. Note that the sliding window mechanism is not required, as the videos are already temporally segmented to the action extent. To minimize the effect of overfitting, we apply a fivefold cross validation and measure the average class accuracy as in [17].

Fig. 17 shows the confusion matrix for the 6-class classification results. The average classification accuracy is listed in Table 4. Interestingly, the moving pose template model with contextual objects already achieves 80 percent accuracy. The performance obtained with the *APT with contextual part* model is comparable to the result from [17]. The difference between 90 and 93 percent is actually just one misclassified test sample. This is an excellent result, considering that the method in [17] requires a static camera and background, rendering it unsuitable for realistic videos such as C&C. Moreover, our method needs substantially less manual annotation for training than [17]. For example, they need the location of the person's hand and a pixelwise segmentation of the object in each frame of the training videos. The confusion matrix in Fig. 15 reveals that most misclassifications are due to the similarity between the actions "lighting torch,", "spraying," and "pouring" water which were distinguished in [17] with a cue based on the

TABLE 3
Results on Coffee and Cigarettes

Supervision	Work	Drinking	Smoking
Semi-supervised	MPT w/ contextual parts	29%	14%
	APT w/ contextual parts	58%	31%
Weakly supervised	APT w/ latent parts	43%	26%
	Laptev *et al.* [26]	43%	-
	Willems *et al.* [40]	45%	-
	Klaeser *et al.* [41]	54%	25%

AP performance for spatiotemporal localizations in percent. The first two rows report the performance of our algorithm. The remaining results are from recent literature.



	answer	call	drink	light	pour	spray
answer	1.0					
call		1.0				
drink			1.0			
light				.80	.20	
pour					.80	.20
spray				.20		.80

Fig. 17. Confusion matrix on the CMU data set.

TABLE 4
Average Classification Accuracy on the CMU Data Set

	CMU Videos
MPT w/ contextual parts	80%
APT w/ contextual parts	90%
APT w/ latent parts	68%
Gupta *et al.* [17]	93%

color of the action-object, which requires manual pixelwise segmentation of the object at training time.

It is also interesting to see that *APT with latent parts* method performs very poorly on this data set (even worse than MPT). This is in fact understandable because, while C&C data set is mainly about localization, CMU data set is a classification test. Therefore, it is much more important to distinguish the subtle difference between classes. This confirms our intuition that modeling contextual objects is the key for solving such a problem.

5.6 UCLA Contextual Action Detection Data Set

Our data set consists of videos of 10 scenes taken from everyday living places such as campus plaza, food court, office, corridors, and so on. Each of these videos contains about a dozen instances from the following event list:

1. purchase from a vending machine,
2. use an elevator,
3. throw trash into a can,
4. use a water dispenser,
5. pick up newspapers from a paper-stand, and
6. sit down on a chair then get up and leave.

Most of these categories involve multiple action phases and involve contextual objects. Fig. 18 illustrate a snapshot of the data set. All the events are annotated with six body parts: "head," "torso," "upper arm," "lower arm," "upper leg," and "lower leg."

What made this data set different special is that its contextual objects are static in the background. Even though it is very hard to directly detect some contextual objects such as "vending machine button," we can exploit the fact that these objects can be represented as hot zones within the scene. Therefore, we do not directly annotate the contextual objects for this data. Instead, we divide actions in each video into two halves and use the first half to learn a semantic map of hot-zones, that is, to build a 2D histogram for each part of interest (in this paper, we consider two parts *lower arm* and *lower leg*). Some examples are shown in Fig. 10b. Since these semantic maps are learned from the "ground truth," they are very accurate. Even if without ground truth, we can imagine ways to automatically learn these hot-zones by accumulated over many instances of actions. This is, however, only feasible in a much larger data set and, hence, beyond the consideration of this paper.

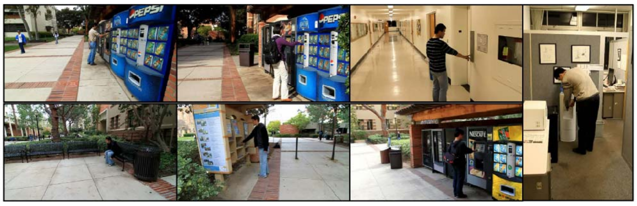


Fig. 18. Snapshots from the UCLA action data set.

TABLE 5
Detection Performance on the UCLA Data Set

Event	APT-full	APT w/ latent parts
vending machine	82%	43%
elevator	92%	67%
throw trash	86%	58%
water dispenser	87%	62%
news-stand	89%	74%
sit down then get up	90%	66%

Applying these semantic maps, we then use the second half of our data for testing. To minimize the effect of overfitting, we apply a fivefold cross validation by randomly choosing different combinations of training and testing actions. The average detection precision is measured for six event classes as shown in Table 5. Since the *APT with latent parts* method does not use the contextual information, it is much worse than the full APT model.

6 Discussion and Future Work

Human actions are complex patterns and most of the current data sets are quite constrained and there is still a long way to go before robust and general vision system can work on generic scenes.

Our model is limited and, thus, can be extended in the following aspects. First, it is two-dimensional and thus view-dependent. For different views, more pose templates are needed. Second, it does not have rich appearance model to account for human clothes at high resolution. The HOG feature for each body part needs more than one templates to account for the intraclass variations. We plan to address the above two problems by using the And-Or graph representation developed by Rothrock and Zhu [42], where different views are modeled by Or-nodes, and each node in the And-Or graph terminates in low resolution. Third, we should also learn the action and contextual objects in 3D model, for example, using Kinect as training data. This will help the action recognition to new scenes for robust performance. Fourth, we are connecting the action recognition with long-term event recognition with goal and intent reasoning as it was shown in [19].

Acknowledgments

This work was supported by the following grants at UCLA: US National Science Foundation (NSF) IIS 1018751, MURI N00014-10-1-0933 and US Defense Advanced Research Projects Agency (DARPA) MSEE project FA 8650-11-1-7149.

References

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," *Proc. IEEE Int'l Conf. Pattern Recognition (ICPR),* 2004.

[2] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2247-2253, Dec. 2007.

[3] K. Schindler and L.V. Gool, "Action Snippets: How Many Frames Does Human Action Recognition Require?" *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[4] I. Essa and A. Pentland, "Coding, Analysis, Interpretation, and Recognition of Facial Expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 757-763, July 1997.

[5] A. Bobick and J. Davis, "The Recognition of Human Movement Using Temporal Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 3, pp. 257-267, Mar. 2001.

[6] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing Action at a Distance," *Proc. Ninth IEEE Int'l Conf. Computer Vision (ICCV),* pp. 726-733, 2003.

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2008.

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proc. IEEE Int'l Conf. Computer Vision Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS),* 2005.

[9] A. Kovashka and K. Grauman, "Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2010.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 9, pp. 1627-1645, Sept. 2010.

[11] W. Yang, Y. Wang, and G. Mori, "Recognizing Human Actions from Still Images with Latent Poses," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 2030-2037, 2010.

[12] Y. Yang and D. Ramanan, "Articulated Pose Estimation with Flexible Mixtures-of-Parts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.

[13] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Pose Search: Retrieving People Using their Pose," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[14] S. Johnson and M. Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation," *Proc. British Machine Vision Conf. (BMVC),* 2010.

[15] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[16] T. Lan, Y. Wang, W. Yang, and G. Mori, "Beyond Actions: Discriminative Models for Contextual Group Activities," *Proc. Advances in Neural Information Processing Systems,* 2010.

[17] A. Gupta, A. Kembhavi, and L. Davis, "Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 10, pp. 1775-1789, Oct. 2009.

[18] B. Yao and L. Fei-Fei, "Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 17-24, 2010.

[19] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing Video Events with Goal Inference and Intent Prediction," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* 2011.

[20] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence,* 1981.

[21] E. Muybridge, *Animals in Motion.* Dover Publications, 1957.

[22] H.-F. Gong and S.-C. Zhu, "Intrackability: Characterizing Video Statistics and Pursuing Video Representations," *Int'l J. Computer Vision,* vol. 97, no. 3, pp. 255-275, 2012.

[23] B. Yao and S. Zhu, "Learning Deformable Action Templates from Cluttered Videos," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* pp. 1507-1514, 2010.

[24] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J. Mazziotta, and G. Rizzolatti, "Grasping the Intentions of Others with One's Own Mirror Neuron System," *PLoS Biology,* vol. 3, no. 3, 2005.

[25] A. Gupta, "Beyond Nouns and Verbs," PhD dissertation, Univ. of Maryland at College Park, 2009.

[26] I. Laptev and P. Pérez, "Retrieving Actions in Movies," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* 2007.

[27] S. Branson, P. Perona, and S. Belongie, "Strong Supervision from Weak Annotation: Interactive Training of Deformable Part Models," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* pp. 1832-1839, 2011.

[28] H. Azizpour and I. Laptev, "Object Detection Using Strongly-Supervised Deformable Part Models," *Proc. 12th European Conf. Computer Vision (ECCV),* 2012.

[29] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade Object Detection with Deformable Part Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 2241-2248, 2010.

[30] T. Joachims, T. Finley, and C. Yu, "Cutting-Plane Training of Structural SVMS," *Machine Learning,* vol. 77, no. 1, pp. 27-59, 2009.

[31] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables," *J. Machine Learning Research,* vol. 6, no. 2, pp. 1453-1484, 2006.

[32] M.P. Kumar, B. Packer, and D. Koller, "Curriculum Learning for Latent Structural SVM," *Proc. Advances in Neural Information Processing Systems (NIPS),* 2010.

[33] L. Cao, Z. Liu, and T. Huang, "Cross-Data Set Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1998-2005, 2010.

[34] J. Niebles, H. Wang, and L. Fei-fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int'l J. Computer Vision,* vol. 79, pp. 299-318, 2008.

[35] J. Liu, J. Luo, and M. Shah, "Recognizing Realistic Actions from Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[36] S. Sadanand and J. Corso, "Action Bank: A High-Level Representation of Activity in Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1234-1241, 2012.

[37] J. Yuan, Z. Liu, and Y. Wu, "Discriminative Subvolume Search for Efficient Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[38] L. Cao, Y. Tian, Z. Liu, B. Yao, Z. Zhang, and T. Huang, "Action Detection Using Multiple Spatial-Temporal Interest Point Features," *Proc. IEEE Int'l Conf. Multimedia and Expo,* 2010.

[39] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive Search Space Reduction for Human Pose Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1-8, 2008.

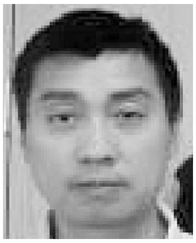
[40] G. Willems, J. Becker, T. Tuytelaars, and L.V. Gool, "Exemplar-Based Action Recognition in Video," *Proc. British Machine Vision Conf. (BMVC),* 2009.

[41] A. Klaser, M. Marszałek, C. Schmid, and A. Zisserman, "Human Focused Action Localization in Video," *Proc. ECCV Workshop Sign, Gesture, and Activity,* 2010.

[42] B. Rothrock and S.-C. Zhu, "Human Parsing Using Stochastic and or Grammar and Rich Appearance," *Proc. IEEE Int'l Conf. Computer Vision (ICCV) Workshop Stochastic Image Grammar,* 2013.

Benjamin Z. Yao received the BS degree from the University of Science and Technology of China, Hefei, China, in 2003 and the PhD degree in statistics from the Department of Statistics, University of California Los Angeles in 2012. Currently, he is a research scientist at Beijing University of Posts and Telecommunications. During 2006-2007, he was a research assistant at Lotus Hill Institute, Ezhou, China. His research interests include human action detection and recognition, human annotated image database, and video surveillance.

Bruce (Xiaohan) Nie received the BS degree in computer science from Zhengzhou University, Zhengzhou, China, in 2009, and the MS degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2012. Currently, he is working toward the PhD degree in the Department of Statistics at the University of California Los Angeles. He was a research assistant during 2010 to 2011 at the Lotus Hill Institute, Ezhou, China. His research interests include video surveillance, action detection and recognition, and object detection.



Zicheng Liu received the BS degree in mathematics from HuaZhong Normal University, Wuhan, China, in 1984, the MS degree in operation research from the Institute of Applied Mathematics, Chinese Academy of Sciences in 1989, and the PhD degree in computer science from Princeton University in 1996. He is a senior researcher at Microsoft Research Redmond. Before joining Microsoft Research, he was at Silicon Graphics Inc. His current research interests include human activity recognition, 3D face modeling and animation, and multimedia signal processing. He is a senior member of the IEEE.

Song-Chun Zhu received the BS degree from the University of Science and Technology of China in 1991 and the PhD degree from Harvard University in 1996. He is a professor with the Department of Statistics and the Department of Computer Science at the University of California Los Angeles. His research interests include computer vision statistical modeling and learning, cognition and AI, and visual arts. He has received a number of honors, including the Marr Prize in 2003 with Z. Tu et al. on image parsing, the Aggarwal prize from the IAPR in 2008, Marr Prize honorary nominations in 1999 for texture modeling and 2007 for object modeling with Y.N. Wu et al., the Sloan Fellowship in 2001, a US National Science Foundation (NSF) Career Award in 2001, and a US Office of Naval Research Young Investigator Award in 2001. He is a fellow of the IEEE.