Simultaneous Inference of Activity, Pose and Object

Furqan M Khan Vivek Kumar Singh Ram Nevatia
Institute of Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273
{furqankh|viveksin|nevatia}@usc.edu

Abstract

Human movements are important cues for recognizing human actions, which can be captured by explicit modeling and tracking of actor or through space-time low-level features. However, relying solely on human dynamics is not enough to discriminate between actions which have similar human dynamics, such as smoking and drinking, irrespective of the modeling method. Object perception plays an important role in such cases. Conversely, human movements are indicative of type of object used for the action. These two processes of object perception and action understanding are thus not independent. Consequently, action recognition improves when human movements and object perception are used in conjunction. Therefore, we propose a probabilistic approach to simultaneously infer what action was performed, what object was used and what poses the actor went through. This joint inference framework can better discriminate between actions and objects which are too similar and lack discriminative features.

1. Introduction

Human action recognition is important for its wide range of applicability from surveillance systems to entertainment industry. Our objective is to not only assign an action label but also provide a description which includes not only "what"' happened but also "how"' it happened by breaking it into component primitive actions, "what"' object was used if any, "where"' the actor and object were and "when"' the interaction took place. This requires action, pose and object recognition. In the past, one or more tasks were performed independently. Our hypothesis is that joint inference can improve accuracy of these tasks and in turn yield more meaningful descriptions.

Action recognition is a challenging task because of high variations in execution style of different actors, view-point changes, motion blur and occlusion, to name a few. Most of the previous work for human activity recognition has been based on modeling human dynamics alone. [6] shows that action recognition can benefit from object context, *e.g.*, presence of 'cup' or 'water bottle' may help in detecting 'drink' action. Further, human movements are not always discriminative enough and their meaning may depend on object type and/or location. For example, 'drinking' vs. 'smoking'. Similarly, actions can help object recognition. Therefore, modeling action-object context can improve both action and object recognition.

(a) Answer phone (b) Drink from a cup

Figure 1. Similarity in pose sequences for two actions

One way to model mutual context is to recognize actions and objects independently and learn co-occurrence statistics. Such an approach is limiting because not only that independent recognition is very challenging but also that same objects can be used to perform different actions. For example, a 'cup' for both 'drinking' and 'pouring'. Also, similar movements with different objects imply different actions, for instance, 'drinking' and 'answering a phone' have similar motion profiles. In such cases, detailed human representation, *e.g.* human pose model, can facilitate accurate human interaction modeling.

Methods have been proposed that use actions, poses and objects to help recognition of actions and/or objects [15, 6]. [15] recognize actions independently using pose observations to improve object detection, whereas, [6] calculates hand trajectories from pose estimates and use them for simultaneous action and object recognition. These methods estimate human pose independently; however, robust estimation of pose is a well-known difficult problem. Commonly, skin color is used to provide context when tracking hands, which limits applicability of the method. Simultaneous inference can improve estimation of all three entities.

Independent object and pose detection in our experiments gave detection rates of 57.8% and 34.96% for respectively.

Recently, [22] presented a method for simultaneous inference of action, pose and object for static images which does not take advantage of human dynamics. Due to absence of human dynamics, the approach cannot differentiate between 'make' and 'answer' a phone call. An intuitive way to extend the approach for videos is to map their model onto an HMM. Such extension will be computationally expensive as a large number of action-pose-object hypotheses need to be maintained in every frame because actions, such as 'drink', can be performed using one of many objects.

Our contribution through this paper is three fold:

i) We propose a novel framework to jointly model *action, pose dynamics* and *objects* for efficient inference. We represent action as a sequence of keyposes. Transition boundaries of keyposes divide actions into their primitive components. To aid video description, we map the recognition problem onto a dynamic graphical model. Inference on it yields likely label and segmentation of action into keyposes and hence into component actions. We also obtain human pose estimates in our inference to answer "where"' and "how"' when producing description. Segmentation by our method is more detailed than [6], which only segments reach and manipulation motion.

ii) We propose a novel two step algorithm in which fewer hypotheses are required to be maintained for every frame in comparison to extending [22] by adding temporal links and running Viterbi algorithm.

iii) Keypose instances may vary highly among actors and even for same actor. To deal with variability in keypose instances we model keypose as a mixture of Gaussian over poses and refer to it as *Mixture of Poses*.

To validate our hypothesis, we evaluate our approach on the dataset of [6] which contains videos of actions using small objects and obtain action recognition accuracy of 92.31%. We show significant improvement over the baseline method which does not use objects. We also observe that doing simultaneous inference of action, pose and object improves action recognition accuracy by 25% in comparison to using only action and pose. Finally, we evaluate the performance of our system for video description task.

2. Related Work

Human activity recognition in video has been studied rigorously in recent years in vision community. Both statistical [10] and graphical [3, 18, 13, 14] frameworks have been proposed in the past. However, most methods try to recognize actions based on human dynamics alone. These methods can be grouped on the basis of using 2D [8, 23, 10] or 3D [20, 11, 14] action models. 2D approaches work well only for the viewpoint they are trained on. [11, 20, 14] learn 3D action models for viewpoint invariance. [11, 14] use foreground blobs for recognition; therefore, their performance depend heavily on accurate foreground extraction. Also, human silhouettes are not discriminative enough, from certain viewpoints, for actions performed by movement of hand in front of the actor, *e.g. drink or call*. We use 3D action models learned from 2D data and do part-based pose evaluation, which does not require silhouettes.

[21, 12, 2] did some early work in modeling action and object context. [12] detects contact with objects to create a prior on HMMs for different actions. [15, 9] does indirect object detection using action recognition. [5] recognizes primitive human-object interactions, such as grasp, with small objects. The method is suited for actions that depend on the shape of the object. [7] tracks and recovers pose of hand interacting with an object. Most of these methods solve for either action or object independently to improve estimate of the other, with an assumption that either one is reliably detected. However, independent estimation is generally difficult for both. [6] uses hand tracks obtained from independent pose estimation for simultaneous inference of action and object; where interaction with objects is restricted to hands only.

Attempts have also been made to recognize actions in a single image using object and pose context [6, 22, 17]. These methods, however, do not use human dynamics cues.

3. Modeling Action Cycle

Graphical modeling of action, pose and object provides a natural way to impose spatial, temporal, functional and compositional constraints on the entities. Our model has separate nodes to represent action, object and actor's state. Actor's state is a combination of actor's pose, duration in the same pose and *micro-event*, which represent transition from one keypose to another. Variations in speed and execution of action are captured through transition potentials.

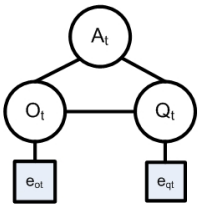
Figure 2 presents the graphical model to simultaneously estimate action A, object of interaction O, and actor's state Q[1] using their mutual context. Evidences for object and actor's state are collected as e_o and e_q respectively.

Let, $s_t = \langle a_t, o_t, q_t \rangle$ denote the model's state at time t, where a_t, o_t and q_t represent the action, the object of interaction and the actor's state at time t. Then the joint log likelihood for a state given an observation sequence of length T can be obtained by a sum of potentials similar to [1].

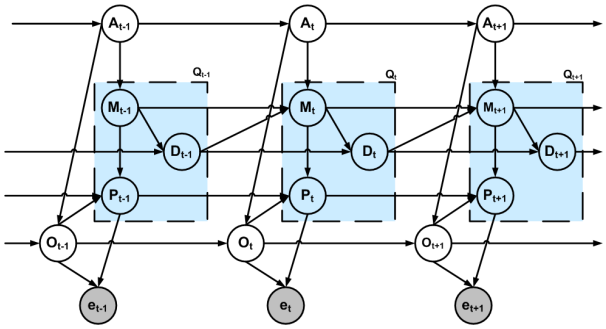
$$\mathcal{L}(s_{[1:T]}, I_{[1:T]}) = \sum_{t=1}^{T} \sum_{f} w_f \psi_f(s_{t-1}, s_t, I_t) \qquad (1)$$

where, $\psi_f(s_{t-1}, s_t, I_t)$ are potential functions that model interaction between nodes in our graphical model and w_f are the associated weights.

[1] We disambiguate states of complete model and state of an actor at an instant by calling them *model's state* and *actor's state* respectively



(a) Basic underlying model



(b) Time Unrolled model

Figure 2. Graphical Model for Actor Object Interaction

The most likely sequence of model's states, $s^*_{1:T}$, can be obtained by maximizing the log likelihood function in Eq (1) for all states

$$s^*_{[1:T]} = \arg\max_{\forall s_{[1:T]}} \mathcal{L}(s_{[1:T]}, I_{[1:T]}) \quad (2)$$

Optimal state sequence obtained using Eq. (2) provides keypose labels for each frame. These labels are used with action and object estimates to generate video description.

3.1. Actor's States

An actor goes through certain poses, called *Keyposes*, during an action which are representative of that action. We use Keyposes as part of our actor's state definition like [11] but represent them using *Mixture of Poses* to handle variation in keypose instances. A Mixture of Poses is a Gaussian mixture of *K* N-dimensional Gaussian distributions, in which, each component distribution corresponds to a 3D Pose with dimensions corresponding to 3D joint locations. We distinguish samples drawn from keypose distributions by referring to them as '*poses*'. Next, we define transitions from one keypose to a different keypose as *micro-events* and include them in our definition of actor's state. This way, pose transitions depend not only on the current keypose but also on the last (dissimilar) keypose. Finally, since probability of staying in the same state decreases exponentially with time, we add duration of being in the same keypose to complete actor's state definition. Therefore, the state of actor at time t, q_t, in our model is represented by the tuple $\langle m_t, kp_t, p_t, d_t \rangle$, where m_t, kp_t, p_t and d_t represent previous micro-event, keypose, pose and duration spent in current keypose respectively.

[14] presented a similar model for actor's state but uses *primitive events* to explicitly model intermediate poses between two 3D keyposes. This does not account for all the pose variations that may occur between two keyposes, *e.g.*, reach motion can be categorized by start and end pose and does not depend on variations in poses in between. We address this by *not* explicitly modeling the transformation of poses. Also, linear transformation assumption suggests that more keyposes would be required for accurate action modeling. [11] uses similar approach to ours but does not model duration of keyposes and will have problem dealing with actions in which actor stays in a keypose for long duration. We permit arbitrary connections for *skipping* and *repeatability* of actor's states within an action.

3.2. Human Object Interaction

A scene may have a number of objects present, requiring us to identify the type and location of the object with which an actor interacts. We have an estimate of actor's joints via pose as part of actor's state. Given an action, joint locations can help us better estimate the type and the location of the object. For example, we expect a cup to be close to the hand when *drink* action is being performed. Conversely, action and object constraint joint locations, *e.g.* when a person is sitting on a chair, his/her hip is usually closer to the chair than the head. We call the joint which is closest to the object during the interaction as the *joint of interaction*.

Objects often get highly occluded during interaction when they are small. For small objects, the pose does not depend on the location of the object but on its type and the action. This fact is also used by Gupta *et al.* in their work [6], where manipulation motion depends only on object's type. Therefore, using location of small object to provide context to the pose during interaction may degrade action recognition performance. In our experiments, objects were detected only 5% of the time after being grabbed. Consequently, a binding of pose and objects at each frame, as presented in [22], may not be optimal because objects don't provide useful context at every frame. On contrary, when objects are large and not occluded, they are easier to detect, however, body parts get occluded during interaction with them. In that case, objects provide useful context.

Based on above observations, we model human object interaction for those portions of the action where objects are more likely to be detected and influence the pose. To capture this notion, we use segmentation provided by micro-events. The micro-event with likelihood of observing the object around it near the joint of interaction greater than some threshold, is called the *binding event*.

3.3. Action Model Learning

First, we select keyposes to represent our actions. Next, we learn keyposes from a small number of pose annotations for each video. Finally, action models and binding events are learned from keypose transition boundary labels.

[Keypose Selection] Keyposes can be selected using local optima of motion energy as explained in [11] which requires MOCAP data. To avoid use of MOCAP data, poses for whole video sequences can be marked in 2D and be lifted to 3D [19, 14]. However, keypose selection using motion energy may not have semantic significance. Since, one of our goals is to explain different phases of an action, we manually choose a set of keyposes based on the semantics for training. This makes our action description task straight forward after most likely keypose sequence labeling.

[Keypose Learning] To learn Mixtures of Poses, we annotate the joints of the actor in only few frames of training videos and assign them to one of the Keyposes. These 2D poses are then lifted to 3D and *normalized* for scale and orientation using a method similar to [19] and [14]. We cluster instances of a keypose using K-means and use each cluster as a component in the mixture with equal weight.

[Actor's State Transition Model Learning] We mark keypose transition boundaries for each video. Using these annotations, mean and variance of duration a person spends in a keypose for every action and a valid set of micro-events and their transitions are learned from these annotations.

[Binding Event Learning] We annotate objects in training data for all frames. Then an object detector for object of interest is run and only true detections are kept. The micro-event around which an object is detected with likelihood above some threshold, is selected as binding event. We collect statistics for all the joints for both starting and ending pose of a micro-event. The joint closest to the object on average is selected as joint of interaction.

[Potential Weights Learning] In principle, weights (w_f) can be learned as in [1] for improved performance. In our experiments, we set the weights to 1.0.

4. Simultaneous Inference of Action, Pose and Object

Obtaining most likely state sequence by exhaustive search is computationally prohibitive due to large state space. A particle filter based approach can be used to do inference efficiently in $\mathcal{O}(K * T)$, where K is the number of particles in a frame (beam size) and T is number of frames.

We propose a novel two step inference algorithm, to reduce the minimum beam size and hence the complexity of inference. We break down Eq. (1) into two parts according to the types of potentials:

$$\begin{aligned}\mathcal{L}(s_{[1:T]}, I_{[1:T]}) =& w_i \sum_{t=1}^{T} \psi_i(s_{t-1}, s_t, I_t) \\ &+ \sum_{t=1}^{T} \sum_{q_j} w_{q_j} \psi_{q_j}(s_{t-1}, s_t, I_t) \end{aligned} \quad (3)$$

where the first term involving $\psi_i()$ models human object interaction and the last term involving $\psi_{q_j}()$ models transitions and observations of actor's states during the action.

First, we obtain samples of object tracks and pose sequences for the video. Finding likely actor's state sequence, $\widehat{s}^*_{[1:T]}$, is equivalent to maximizing the last term of eq (3). This gives us a distribution of poses in every frame. In the second step, we obtain the most likely sequence of action, object and actor's state by computing interaction potentials between all possible pairs of object tracks and actor's states. Pseudo code for inference is given in Algorithm 1.

Algorithm 1 Pseudo Code for Inference

1: Use particle filter based algorithm to obtain pose distribution: $\widehat{S}^* = \left\{ \widehat{s}^*_{i_{[1:T]}} \right\}_{i=1}^{K}$.

$$\widehat{s}^*_{i_{[1:T]}} = \arg \max_{\forall s_{[1:T]}}{}^{i} \left(\sum_{t=1}^{T} \sum_{q_j} w_{q_j} \psi_{q_j}(s_{t-1}, s_t, I_t) \right) \quad (4)$$

where, $\max^i(.) = i^{th}$ best solution of (.)

2: Obtain distribution of objects using window based detection and tracking

3: Obtain most likely sequence of states, $s^*_{[1:T]}$ by maximizing Eq. (1) over $\widehat{S}^*$

$$s^*_{[1:T]} = \arg \max_{s_{[1:T]} \in \widehat{S}^*} (\mathcal{L}(s_{[1:T]}, I_{[1:T]})) \quad (5)$$

[Complexity Analysis] Let $\mathbf{A}$ be the set of actions and $\mathbf{O}_a$ and $\mathbf{B}_a$ be the set of objects and binding events for action a, respectively. To do joint inference in single pass, we need to maintain at least $K_{min} = \sum_{a \in \mathbf{A}} |\mathbf{O}_a|$ hypotheses to represent all actions at least once. Therefore, the complexity of a single pass algorithm for the smallest beam becomes:

$$\mathcal{T}_1 = \mathcal{O}(T * \sum_{a \in \mathbf{A}} |\mathbf{O}_a|) = \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * T)$$

For our two pass algorithm, we evaluate interaction potentials only for frames where binding events occur, *i.e.* $|\mathbf{B_a}| < T$. Since we bind objects in second pass, we only need to maintain at least $K_{min} = |\mathbf{A}|$ hypotheses during first step. In second step, we evaluate interaction potential $|\mathbf{O}_a| * |\mathbf{B}_a|$ times for each action a.

$$\mathcal{T}_{2a} = \mathcal{O}(|\mathbf{A}| * T) \leq \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * T)$$

$$\mathcal{T}_{2b} = \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * |\mathbf{B}_a|) < \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * T)$$

$$\mathcal{T}_2 = \mathcal{O}(|\mathbf{A}| * T) + \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * |\mathbf{B}_a|) \leq \sum_{a \in \mathbf{A}} \mathcal{O}(|\mathbf{O}_a| * T)$$

If at least one action can be performed using more than one object, the above inequality is strict.

4.1. Pose Tracking

We use a top-down approach to obtain $top-K$ likely sequences of poses, independent of the object. At any instant t, the actor state transition models are used to predict 3D poses for instant $t+1$. The likelihood for these poses is then computed by evaluating them against the video evidence. For computational efficiency, we use a particle filter based approach for inference similar to [14].

We keep a total of K states $\{s_t^i\}_{i=1:K}$ at each frame t. In order to have a good representation of all actions when we bind pose tracks and objects at later stage, we ensure that we keep at least $M = floor(K/|A|)$ states for each action at every frame. For each state s_t, we sample the next micro-event m_t given the action a_t using *Event Transition Potential* $\psi_m(s_{t-1}, a_t, m_t)$. Next, using the predicted micro-event we sample pose (p_t) for next state from the Mixture of Poses corresponding to ending keypose of the micro-event. Finally, observation potential $\psi_o(p_t)$ is computed for the next predicted pose. The weighted sum of potentials is accumulated to find $top-K$ state sequences. Pseudo code and further details of our algorithm are provided as supplemental reading.

Observation and transition potentials can be any function depending on the domain. We explain our choices below.

4.1.1 Transition Potential

In our implementation we allow transition between actions only after last micro-event of the current action. Therefore, the action label is carried from previous state to the next except for the last micro-event. We want the probability of staying in the keypose to decay near mean duration. [14] models this using a *log of signum* function. We define *Event Transition Potential*, $\psi_m(.)$, like them as follows:

$$
\begin{aligned}
&if\ a_{t-1} = a_t: \\
&\quad \psi_m(s_{t-1}, a_t, m_t) = \\
&\quad \begin{cases} -ln\left(1 + e^{\frac{d_t - \mu(m_t) - \sigma(m_t)}{\sigma(m_t)}}\right) & m_{t-1} = m_t \\ -ln\left(1 + e^{-\frac{d_t - \mu(m_t) + \sigma(m_t)}{\sigma(m_t)}}\right) & m_{t-1} \neq m_t \end{cases} \qquad (6) \\
&if\ a_{t-1} \neq a_t: \\
&\quad \psi_m(s_{t-1}, a_t, m_t) = \\
&\quad \begin{cases} -ln\left(1 + e^{-\frac{d_t - \mu(m_t) + \sigma(m_t)}{\sigma(m_t)}}\right) & m_{t-1} = lm(a_t) \\ -\infty & otherwise \end{cases} \qquad (7)
\end{aligned}
$$

where, $lm(a_t)$ is the last micro-event for action, and $\mu(m_t)$ and $\sigma(m_t)$ are the mean and variance of duration an actor spend in the same keypose after micro-event m_t.

4.1.2 Observation Potential

We compute the observation potential of each keypose by matching its shape with the edges we extract from the video. To obtain the shape model, each keypose is scaled and rotated (in pan) appropriately based on the tracking information and then projected orthographically. For robustness, we use two different shape potentials - Hausdorff distance of the shape contour and the localization error of a 2D part based model (pictorial structure [4]).

[Hausdorff Distance] Given a keypose kp, we obtain its shape contour by collecting the points on the boundary of the projected pose. The shape contour is then matched with the canny edges in the image I using the Hausdorff distance.

$$\psi_H(S, I) = \max_{p \in kp_{cont}} \min_{e \in E(I)} ||a - e|| \qquad (8)$$

where kp_{cont} is the pose shape contour, $E(I)$ are canny edges of image I and $||.||$ is any norm. We used the euclidean norm between the model points and edge points, as it can be computed efficiently by computing the distance transform of the edge image.

[Part Localization Error] To handle the variations in pose across different action instances, we use a 2D part model to accurately fit the projected 3D pose to image observations. The body part model used in the work is similar to the *Pictorial Structures* [4] which is widely used for estimating human pose in an image [16]. The model represents each part as a node in a graphical model and the edges between nodes represent the kinematic relation between the corresponding parts. Note that unlike [4, 16], which assume the parts to be unoccluded, our body model is defined over only the observable parts; note that a part may not be observable either due to inter-part occlusion or 3D-2D projection (projected length may be too small). Furthermore, the estimate of the adjusted 3D pose imposes a strong constraint on the orientation and position of body parts and localization in our case does not require a dense search [4, 16].

To determine which parts are observable, we first compute the visibility of each part by computing the fraction of overlap between that part and other body parts, and using their depth order w.r.t. the camera. In this work, we consider a part to be occluded if the fraction of part occluded is greater than 0.5.

For localization, we first apply a template for each part over the expected position and orientation of the part and a small neighborhood around it. In this work, we used the boundary templates provided by Ramanan et al [16]. Pose estimate is then obtained by maximizing the joint log likelihood of the visible parts and the kinematic constraints between the parts (similar to [4]). After localization, we normalize the total potential by number of visible parts to remove bias towards poses with fewer visible parts.

4.2. Object Recognition and Tracking

We train an off-the-shelf window based object detector that does not use color for each type of object used. Our

Figure 3. Object visibility for different keyposes. Even for static camera, appearance of spray bottle and flashlight change significantly between frames.

method does not depend on the specifics of the object detector. We apply these detectors to obtain a set of candidates for each object of interest. We associate object candidates in each frame, by running an off-the-shelf detection based object tracker. Running an object tracker gets rid of intermittent false alarms and miss detections. Still, each frame may have more than one type of object and more than one candidate of each object type present.

4.3. Pose-Object Binding

Object detection and tracking in full generality is a challenging problem, therefore, we do not expect reliable object detections to be available throughout the action (Fig. 3). Instead, we bind the objects with the action when we have the best chance of detecting them. For a *pick up* action, it is before the object is removed from its location, while for a *put down* action it is after the object is put down. We learn these facts from data as explained earlier.

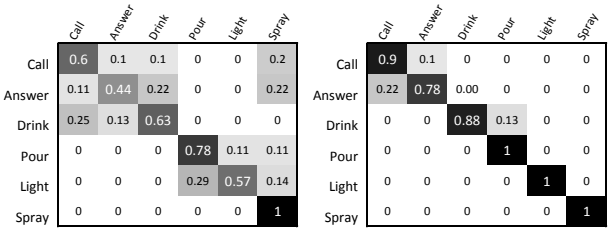
Once we have pose and object tracks, we compute interaction potential among poses, objects and action labels. For each pose track given an action, we compute interaction potential for object hypotheses (all detection across all types) and the *joint of interaction* of either the source or destination pose at each binding event for that action. We define two binding functions depending on object visibility before or after the occurrence of the binding event. Let j^a be the joint of interaction for action a. For frame t, $l(p_t^j)$ be the location of joint j for pose p and $l(o_t)$ be the location of object o, the binding functions are defined as:

$$\begin{aligned}\mathcal{B}_0(s_{t-1}, s_t) &= -dist(l(o_{t-1}), l(p_t^{j^a}))/r \\ \mathcal{B}_1(s_{t-1}, s_t) &= -dist(l(o_t), l(p_{t-1}^{j^a}))/r \end{aligned} \tag{9}$$

where, r is used to normalize the distance. Now, for the binding event b^a for action a, the interaction potential is defined as:

$$\psi_i(s_{t-1}, s_t, I_t) = \begin{cases} max(\mathcal{B}_0(.), \mathcal{B}_1(.), i_{min}) & m_t = b^a \neq m_{t-1} \\ 0 & otherwise \end{cases} \tag{10}$$

Due to uncertainty in estimates of location of object and joint of interaction in one frame, we in practice, use mean location over n frames before or after t by replacing $l(x_t)$ with $\widehat{l}(x_{t:t+m})$ and $l(x_{t-1})$ to $\widehat{l}(x_{t-1-m:t-1})$.



(a) Using Partial Model (b) Using Full Model

Figure 4. Confusion Matrix

5. Experiments

We evaluated performance of our system for video description task on the dataset of [6]. The dataset has 10 actors performing 6 different actions using 4 objects. The actions in the datasets are drinking, pouring, spraying, lighting a flashlight, making a phone call and answering a phone call. These actions are performed using one of the four objects, cup, phone, spray bottle or flashlight. The videos, however, contain other objects such as a stapler and a masking tape. Only 52 of 60 videos used for testing in [6] were made available by the authors for evaluation.

[Object Detection] To evaluate the performance of our object detector for recognition of objects before they are grabbed we ran all the detectors at different scales. Locations for which no class had a likelihood of more than 0.5 were classified as *background*. Otherwise, the class with the highest likelihood was assigned to that location. When using object detectors without action context, we achieved 63.43% recognition rate for localized objects of interest before being grabbed. Figure 5 demontstrates the effect of using object context.

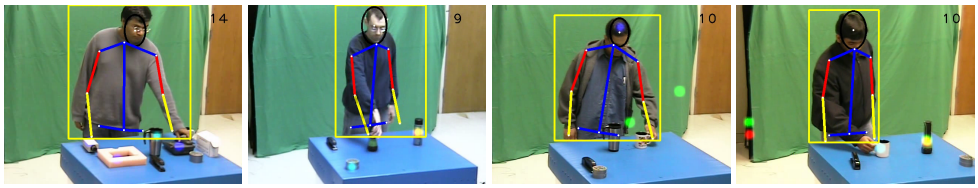
We similarly evaluated our object detector for recognition of objects after being grabbed. As expected, only 5% of the objects were detected correctly, indicating that small objects cannot be reliably detected during interaction phase.

[Baseline] For action recognition task, we first established a baseline using a method which does not use object context. We used space-time interest points based method of [10]. Code provided by authors was used to obtain interest points and SVM-Light was used for classification. Our setup produced results similar to the authors on KTH dataset. On our dataset, we ran experiments for single channel with variety of configurations and obtained best accuracy of *53.8%* using HOF-313. Same configuration is reported to give best performance for single channel on KTH dataset in [10].

[Action Recognition without Object] Next, we evaluated the performance of our action recognition system. We first

Method	Accuracy
Space-Time Interest Points [10]	53.8%
Keypose based Action Model	67.31%
Full model with context(Section 3)	**92.31%**

Table 1. Action Recognition Accuracy



(a) (b) (c) (d)

Figure 5. Effect of using object context. Likelihoods of cup, phone, flashlight and spray bottle at each location of interaction frames are shown in cyan, blue, green and red respectively. (a) Phone, 'calling' action and pose were correctly recognized in presence of multiple object detections. (b) Object context helped to disambiguate 'lighting' from 'pouring' action. (c) 'Pouring' action was recognized in absence of 'cup'. (d) Detection of 'cup' confused 'drinking' with 'pouring' action.

Pose Classification	Accuracy
Without action context	34.96%
With action context	66.43%
With action and object context	**73.29%**

Table 2. Pose Recognition Accuracy

performed action recognition without using object knowledge. This is equivalent to using (3) without the $\psi_i(.)$ term and the most likely action was reported. We used *leave-one-out validation* method for testing *i.e.* we trained our models using 9 actors and tested it on the remaining actor. Recognition rate of *67.31%* was achieved using the limited model. Significant confusion occurs between *pouring from a cup* and *lighting a flashlight* and *drinking* and *answering* actions because of similarity in limb movements. Some of *call* and *answer* actions are confused with *spraying* actions, because in 2D, initial few poses of a spraying action are similar to that of calling and answering actions.

[Pose Classification] We setup our pose classification task as to classify the human pose at each frame into one of 20 Keypose classes. Approximately, 5300 frames were evaluated. As reported in Table 2, recognition improves as more contextual information is provided. Accuracy is low without action context because we chose keyposes based on their semantic meanings, therefore, their joint locations appear to be very similar from certain viewpoints. Note that accuracy for no context is still better than random guess of 5%.

[Action Recognition with Object]To study the impact of using object context on action recognition, we performed evaluation of our full model as explained in Section 3. This improved recognition accuracy from 67.31% to **92.31%** and only 4 out of 52 videos were mis-classified. Our method also demonstrates significant improvement over the baseline method. We, however, can't compare our results directly with [6] who reports accuracy of 93.34% because a) the authors used a training set separate from the test set and b) report results for 60 videos. Both the training set and unavailable videos were not provided by the authors for they got corrupt/lost. Our experiments show that use of object context in our proposed framework increases recognition rate significantly. Note that for generality of application, we do not use skin color model for better alignment of hand

```
<Action Name="Call">
    <Description>
    <Element Verb="Stand" Object="NULL" StartFrame="1" EndFrame="13"/>
    <Element Verb="Grab" Object="Phone" StartFrame="14" EndFrame="23"/>
    <Element Verb="Dial" Object="Phone" StartFrame="24" EndFrame="55"/>
    <Element Verb="Listen" Object="Phone" StartFrame="56" EndFrame="83"/>
    <Element Verb="Hold-Midway" Object="Phone" StartFrame="84" EndFrame="89"/>
    <Element Verb="Relinquish" Object="Phone" StartFrame="90" EndFrame="99"/>
    </Description>
</Action>
```

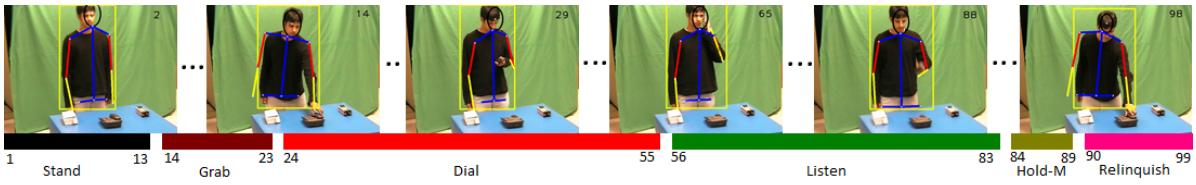
Figure 7. XML description generated by our framework

locations, which might be useful in this case.

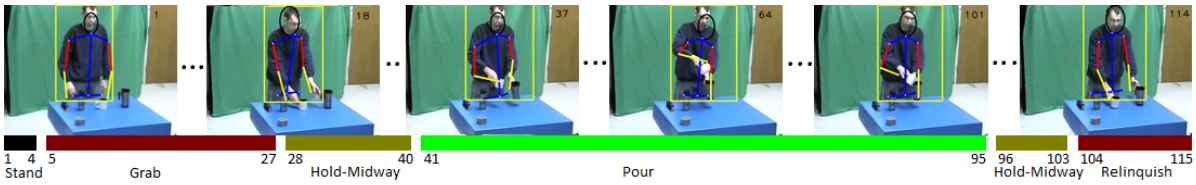
[Video Description]Finally, we evaluated performance of our system for video description task. We assign a "verb"' to every keypose in our system. The scene description is then generated using the keypose segmentation provided by our inference algorithm; we represent description as a set of tuples $\langle$*verb, object, start frame, end frame*$\rangle$, segmenting the video into component actions. Each tuple represents the verb assigned to the duration of video marked by start and end frames and the object associated with the verb. For verbs relating to non-interaction phase, like 'Stand', we report 'NULL' for object. An example output XML file is shown in Fig 5 for 'making a call' action. We quantitatively evaluate the accuracy of our description results by comparing the segment (component action) boundaries with the ground truth segment boundaries, which is available from keypose boundary annotations. When the actions were correctly recognized, our description sequence exactly matched the ground truth and verb durations overlapped with **73.29%** accuracy.

6. Conclusion

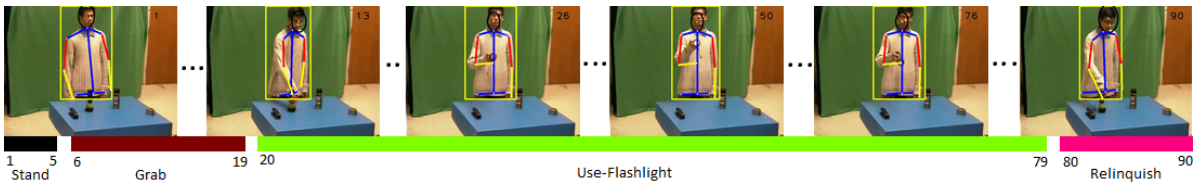
Object identification plays an important role in discrimination of actions that involve similar human movements, whereas knowing the action can help resolve disambiguities between objects on the basis of their normal use. We presented a probabilistic approach to utilize the mutual context that action and objects provide to each other. We represented an action as a sequence of *Mixture of Poses* that captures pose variations across different action instances in a compact manner. By combining human pose and object information in the same probabilistic model and performing joint inference, we were able to better discriminate between actions which have similar poses. We applied our approach to a dataset of human actions that involve interaction with



(a) Making Phone Call



(b) Pouring from Cup



(c) Using Flashlight

Figure 6. Pose Tracking results for actions that were correctly identified.

objects and showed that action recognition improves when pose and object are used together.

References

[1] M. Collins. Discrminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP*, 2002. 282, 284

[2] J. Davis, H. Gao, and V. Kannappan. A three-mode expressive feature model on action effort. In *WMVC*, 2002. 282

[3] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR*, 2005. 282

[4] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. In *IJCV*, 2005. 285

[5] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *CVPR*, 2008. 282

[6] A. Gupta and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. In *PAMI*, 2009. 281, 282, 283, 286, 287

[7] H. Hamer, K. Schindler, E. Koller-Meier, and L. V. Gool. Tracking a hand manipulating an object. In *ICCV*, 2009. 282

[8] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005. 282

[9] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 115(1):81–90, 2011. 282

[10] I. Laptev, M. M. nad C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 282, 286

[11] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007. 282, 283, 284

[12] D. Moore, I. Essa, and M. Hayes. Exploiting human action and object context for recognition taks. In *ICCV*, 1999. 282

[13] L.-P. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*, 2007. 282

[14] P. Natarajan, V. K. Singh, and R. Nevatia. Learning 3d action models from a few 2d videos for view invariant action recognition. In *CVPR*, 2010. 282, 283, 284, 285

[15] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *ICCV*, 2005. 281, 282

[16] D. Ramanan. Learning to parse images of articulated objects. In *NIPS*, 2006. 285

[17] V. K. Singh, F. M. Khan, and R. Nevatia. Multiple pose context trees for estimating human pose in object context. In *IEEE Workshop on Structured Models in Computer Vision in conjunction with CVPR*, 2010. 282

[18] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV*, 2005. 282

[19] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *CVPR*, 2000. 284

[20] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. In *CVIU*, 2006. 282

[21] A. Wilson and A. Bobick. Parametric hidden markov model. In *IEEE PAMI*, 1999. 282

[22] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-pose interaction activities. In *CVPR*, 2010. 282, 283

[23] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In *CVPR*, 2005. 282