Modeling 4D Human-Object Interactions for Event and Object Recognition

Ping Wei\textsuperscript{1,2}, Yibiao Zhao\textsuperscript{2}, Nanning Zheng\textsuperscript{1}, and Song-Chun Zhu\textsuperscript{2}
\textsuperscript{1}Xi’an Jiaotong University, China \textsuperscript{2}University of California, Los Angeles, USA
pingwei.pw@gmail.com,nnzheng@mail.xjtu.edu.cn \{yibiao.zhao,sczhu\}@stat.ucla.edu

Abstract

Recognizing the events and objects in the video sequence are two challenging tasks due to the complex temporal structures and the large appearance variations. In this paper, we propose a 4D human-object interaction model, where the two tasks jointly boost each other. Our human-object interaction is defined in 4D space: i) the co-occurrence and geometric constraints of human pose and object in 3D space; ii) the sub-events transition and objects coherence in 1D temporal dimension. We represent the structure of events, sub-events and objects in a hierarchical graph. For an input RGB-depth video, we design a dynamic programming beam search algorithm to: i) segment the video, ii) recognize the events, and iii) detect the objects simultaneously. For evaluation, we built a large-scale multiview 3D event dataset which contains 3815 video sequences and 383,036 RGBD frames captured by the Kinect cameras. The experiment results on this dataset show the effectiveness of our method.

1. Introduction

The past decade has seen remarkable progress in event understanding\cite{5,8,12,22}. An event usually exhibits complex temporal structures. It can be decomposed into several sequential sub-events or atomic events in the temporal domain\cite{12}. In addition to recognizing the entire event, modeling and recognizing these atomic events are also important, especially in the real applications, like predicting agent’s goal and intention in actions\cite{12}.

The man-made indoor objects are always involved in the human action. It is usually hard to recognize and localize them by appearance, due to the motion and occlusion caused by human action. Actually, the man-made indoor objects are mainly defined by function rather than appearance, like the \textit{cellphone} for making a call. A cellphone is a cellphone because of its ability to allow the agent to perform the action \textit{make a call}. This ability is known as affordance\cite{3,4,25}. When someone is making a call, it is hard to detect the occluded cellphone in the hand. But we can reasonably predict that there is a cellphone in the hand according to the action. This is like a ‘pantomime’. When someone is performing actions in a scene, even if without seeing the objects themselves, we can guess and predict the classes, locations, and even the sizes of the objects, according to the actions. Furthermore, in the progress of an event, the location of an object is coherent. For example, in the event \textit{fetch water from dispenser}, the \textit{dispenser} almost stays still, and the \textit{mug} smoothly moves with the hand.

An event is a sequence of time-varying interactions between human and objects with hierarchical structures in 3D spatial domain and 1D temporal domain.

In this paper, we propose a 4D human-object interaction model (4DHOI) for event recognition and object detection. The framework is shown in Figure 1. The human-object interaction relation is embedded in 4D space: i) the semantic co-occurrence and geometric compatibility of human pose and object in 3D spatial domain; ii) the atomic event transition and object coherence in 1D temporal domain. We model the 4D human-object interaction with a hierarchical graph, as Figure 2 shows. An event is decomposed into several sequential atomic events. The atomic event is decomposed into human pose and objects.

Given the RGBD video and the human pose from the Kinect camera\cite{18}, we design an online dynamic programming beam search algorithm to segment the video, recognize the events, and detect the objects in each video frame. In each frame, the human pose predicts possible object classes and their 3D locations where the objects are searched. The possible interpretations to this frame are jointly proposed according to the human pose, the objects, and the 3D spatial relations between them. The temporal relations between frames are incorporated to optimize those proposals for each frame. In this way, the algorithm generates the hierarchical event interpretation and correspondent object labeling. This framework is illustrated by Figure 1.

Dataset. To evaluate our method, we built a large-scale 3D event dataset with human-object interactions. It is captured by three stationary Kinect cameras from different viewpoints simultaneously. It includes 8 event categories and 11 interacting object classes. It has totally 3815 even-
t video sequences and 383,036 RGBD frames. Each event category includes about 477 video sequence instances. We test our model on this dataset and the experiment result demonstrates the strength of our model.

1.1. Related Work

**Human-object Context.** In recent years, many work applied the human-object mutual context to event and object recognition [2, 5, 7, 9, 11, 14, 15, 23, 24]. Gupta et al. [5] combined the spatial and functional constraint between human and objects to recognize action and object. Yao and Fei-Fei [24] modeled the relations between actions, objects, and poses in still image for detecting objects. These work define the human-object interaction in 2D image. Such contextual cues are often compromised due to their sensitivity to viewpoint changes and temporal variations.

Koppula et al. [7] used Markov random field to model the relations between human activity and object affordance, as well as their changes over time. This method needs the video to be pre-segmented, and all the relations are defined between these small segments. Such strategy makes it hard to understand the contents of object and human action in each frame. And it detects and tracks objects independently of the contextual feedback from human action. Different from it, our model defines the relations within each frame or between frames. And our model incorporates the object detection, tracking, and human action modeling into a unified framework, under which these tasks mutually facilitate each other.

**Event Recognition.** Event is usually recognized by combining the human body features and the temporal relation [3]. Some work took the event recognition as a classification problem. They represented the pre-segmented video as a feature vector, and classified it to an event category. Such methods are advantageous in the computation efficiency. But they can not interpret the inner structure of the video, like the actions and objects in each frame. Also, they are ineffective in real applications like video surveillance. In addition to event classification, our model can segment the video, recognize the atomic events and objects in each frame.

**Temporal Structure of Event.** The hidden Markov model [16] is usually used to model the transition between video frame states [8]. Tang et al. [20] introduced duration variables to the HMM and modeled them with multinomial distribution. Sung et al. [19] decomposed the human activity into sub-activities and model the hierarchical structure with maximum entropy Markov model. They solved this model by graph structure selection in the dynamic programming framework. However, this work do not consider the object interactions and the duration of the sub-activity. Pei et al. [12] represented an action with several atomic actions and employed a temporal filter embedded in an And-or graph for video parsing. Inspired by these work, our model integrates human action, object, and their 4D interaction relations into a unified framework.

2. Hierarchical Graph Model of Event

In the 1D temporal domain, an event is decomposed into multiple ordered smaller atomic events. For example, the event *fetch water from dispenser* in Figure 2 is decomposed into three sequential atomic events - *approach the dispenser*, *fetch water*, and *leave the dispenser*.

In the 3D spatial domain, each atomic event is decomposed into human pose, interacting objects, and the geometric relations between them. An atomic event integrates a specific type of human pose and one or more objects. The semantic relation between the object class and a specific atomic event is a hard constraint. For example, the atomic event *fetch water* consists of the pose *fetch* and the objects...
Suppose \( V = (I_1, ..., I_T) \) is an event video sequence in the time interval \([1, T]\), where \( I_t \) is the RGBD frame at time \( t \). The sequence \( V \) is interpreted by a hierarchical graph \( G = (E, L) \):

i) \( E = \{ e_i | i = 1, ..., |E| \} \) is the event category like “fetch water from dispenser.” \( E \) is the set of event categories.

ii) \( L = (l_1, ..., l_T) \) is a sequence of frame labels. \( l_t = (h_t, o_t, a_t) \) is the interpretation to the frame \( I_t \). \( h_t \) is the human pose, \( o_t = (o_1^t, ..., o_n^t) \) are the objects interacting with human, where \( n_t \) is the number of objects. Each object includes the attributes of class label and 3D location.

The energy that the video \( V \) is interpreted by graph \( G \) is defined as

\[
\Phi(l, l_t) = \Phi_1(a_t, h_t) + \Phi_2(a_t, o_t, l_t) + \Phi_3(a_t, h_t, o_t)
\]

The energy that the video \( V \) is interpreted by graph \( G \) is defined as

\[
\Phi(l_t, l_t) = \Phi_1(a_t, h_t) + \Phi_2(a_t, o_t, l_t) + \Phi_3(a_t, h_t, o_t)
\]

We define \( \Phi(I_t, l_t) \) as:

\[
\Phi(I_t, l_t) = \phi_1(a_t, h_t) + \phi_2(a_t, o_t, l_t) + \phi_3(a_t, h_t, o_t)
\]

\[
\Phi(l_t, l_t) = \phi_1(a_t, h_t) + \phi_2(a_t, o_t, l_t) + \phi_3(a_t, h_t, o_t)
\]

**Pose Model.** \( \phi_1(a_t, h_t) \) is the human pose model. The human pose with 20 3D joints are estimated by the Kinect [8]. To normalize the data, we align all the skeletons to a reference pose so that the torso and shoulder of all poses have the same location, size, and direction.

The feature of each joint is defined as the 3D coordinate concatenating the motion vector which is the difference of joint coordinates in two successive frames. We extract a feature vector containing the features of joints on arms and apply the PCA to the feature vector to reduce the correlation and noise. \( h_t \) is the vector of the PC parameters. We assume that \( h_t \) follows a Gaussian distribution, and then \( \phi_1(a_t, h_t) = -\ln N(h_t; \mu_{a_t}, \Sigma_{a_t}) \), where \( \mu_{a_t} \) is the mean and \( \Sigma_{a_t} \) the covariance.

**Object Model.** \( \phi_2(a_t, o_t, l_t) \) is the object detection model. Suppose \( z_i^t \) is the 3D bounding box center of the object \( o_i^t \) in the 3D space. The 3D box is projected into the RGB and depth images to form 2D bounding box, in which the RGB and depth HOG features \( z_i^t \) are extracted. The probability of object \( o_i^t \) at \( z_i^t \) is obtained by normalizing the SVM detector with Platt scaling \( p(o_i^t | z_i^t) = 1 / (1 + \exp(\mu s(z_i^t) + \nu)) \) [6, 13], where \( s(z_i^t) \) is the score of linear SVM object detector with the RGBD HOG features at location \( z_i^t \). \( \phi_2(a_t, o_t, l_t) \) is formulated as

\[
\phi_2(a_t, o_t, l_t) = -\frac{1}{n_t} \sum_{i=1}^{n_t} \ln p(o_i^t | z_i^t)
\]

We use a sliding window detection strategy to search the objects. But different from [6] where the sliding window is defined on the 2D image plane, we slide the 3D window box in the 3D space where the point cloud is not empty. We then project the 3D window into the 2D image to extract the appearance feature, as Figure 1 shows. Since the instances of the same object class usually have similar sizes in 3D space, we define a prior 3D size for each object class.

Our model defines object location and scale in the 3D space, and appearance in the 2D image, which are more robust to the viewpoint and scale changes. It also provides a natural way to define human-object relations in 3D space.

**3D Geometric Compatibility.** \( \phi_3(a_t, h_t, o_t) \) measures the human-object geometric relations. As Figure 3 shows, the geometric relation in 2D image is not applicable in different viewpoints. We model this relation in 3D space.

In an atomic event, the location of an object is closely related to the locations and directions of some body parts, which we call the key parts, as the arm to the dispenser in Figure 3. Suppose \( y_{o_i} \) is the difference vector from the key...
parts center to the object bounding box center. \( x_{o1} \) is the difference vector between the end points of the key parts. \( y_{o1} \) is closely related to \( x_{o1} \). We define \( n_{o1} = y_{o1} - W_{o1}^T x_{o1} \), where \( W_{o1}^T \) is a similarity transformation matrix. We assume \( n_{o1} \) follows the Gaussian distribution. The 3D geometric relation is modeled as:

\[
\phi_3(a_t, h_t, o_t) = -\frac{1}{nt} \sum_{i=1}^{nt} \ln N(n_{o1}; \mu^R_{o1,a1}, \Sigma^R_{o1,a1}) \tag{4}
\]

where \( \mu^R_{o1,a1} \) is the mean and \( \Sigma^R_{o1,a1} \) is the covariance. The superscript \( R \) is a sign which is used to differentiate the 3D relation Gaussian parameters from others. The subscript \( (o_1, a_t) \) indicates that the human-object geometric relation varies in different atomic events and objects.

The key body parts vector \( x_{o1} \) is like a local reference system, by which we can estimate \( y_{o1} \), and therefore predict the locations of related objects.

2.2. Temporal Relation

The temporal relation \( \Psi(l_{t-1}, l_t) \) is decomposed as:

\[
\Psi(l_{t-1}, l_t) = \psi_1(a_{t-1}, o_t) + \psi_2(o_{t-1}, o_t) \tag{5}
\]

where \( a_{t-1} \) are the atomic event labels of the frames from the time 1 to \( t-1 \). The first term encodes the atomic event transition, and the second term encodes the object tracking.

Atomic Event Transition. In an event, the transition probability from the current atomic event to the next atomic event is related to the duration of current atomic event. We propose to model the time-varying transition probability with the logistic sigmoid function.

Suppose \( \omega_{k-1} \) and \( \omega_k \) are two neighboring atomic events of event \( E \). Given \( E \) and \( a_{t-1} = \omega_{k-1} \), the next frame’s atomic event \( a_t \) can be \( \omega_{k-1} \) (repeat the same atomic event) or \( \omega_k \) (start a new atomic event). \( d_{k-1} \) is the continuous duration of \( \omega_{k-1} \) up to time \( t-1 \). The time-varying transition probability \( p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1}) \) is modeled as:

\[
p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1}) = \sigma(\beta d_{k-1} + \gamma) \tag{6}
\]

\( \sigma(v) = 1/(1 + e^{-v}) \) is the logistic sigmoid function. \( \beta \) and \( \gamma \) are the function parameters. We simplify \( p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1}) \) as \( p(\omega_k | \omega_{k-1}, d_{k-1}) \). The transition probability to \( \omega_{k-1} \) is \( p(\omega_{k-1} | \omega_{k-1}, d_{k-1}) = 1 - p(\omega_k | \omega_{k-1}, d_{k-1}) \). Then \( \psi_1(a_{t-1}, a_t) \) is modeled as

\[
-\ln p(\omega_{k-1} | \omega_{k-1}, d_{k-1}) \text{ or } -\ln p(\omega_{k-1} | \omega_{k-1}, d_{k-1}) \text{, up to the value of } a_t.
\]

Figure 4 shows two kinds of transition probability. To the duration-dependent transition, at the preliminary stage of \textit{approach the dispenser} when the hand is still far from the dispenser, the probability of transition from \textit{approach the dispenser} to \textit{approach the dispenser} is much larger than the possibility to the next atomic event \textit{fetch water}, as the interval 1 in Figure 4 (a). If \textit{approach the dispenser} has been lasting a long time, as in the interval 3, then the probability of transition to \textit{approach the dispenser} will be much smaller than the probability to \textit{fetch water}. In interval 2, the transition choice is indeterminate. The interval 1 and 3 describe the common duration distribution of the atomic event, and the interval 2 reflects the variance. To the duration-independent transition, the probability is constant regardless of the duration, as Figure 4 (b) shows.

Object tracking. \( \psi_2(o_{t-1}, o_t) \) describes the object location tracking. In an event, the locations of some objects like dispenser are rare to be changed. Some objects like mug can move when human action is applied. To the \textit{movable} objects, we assume the location follows a Gaussian distribution \( p(z_t^i | z_{t-1}^i) = N(z_t^i - z_{t-1}^i; \mu^Z_{o1,a1}, \Sigma^Z_{o1,a1}) \). To the \textit{non-movable} objects, we set a hard threshold. If the difference of proposed location in the current frame and the last frame is smaller than the threshold, \( p(z_t^i | z_{t-1}^i) = 1 \), otherwise 0. The tracking energy is

\[
\psi_2(o_{t-1}, o_t) = -\frac{1}{nt} \sum_{i=1}^{nt} \ln p(z_t^i | z_{t-1}^i) \tag{7}
\]

2.3. Learning Atomic Events

We use the manually labeled video sequences (detailed in section 4.1) of event category \( E \) to learn its atomic events. Each sequence contains one instance of the event \( E \) from the beginning to the end. First, we use EM algorithm to cluster the pose feature and time order in all video frames of \( E \) so that each sequence is grouped into \( K_E \) segments.
Based on the $K_E$ segments, we obtain $K_E$ atomic events for $E$. The pose model of the $k$th atomic event is the $k$th component of the mixture Gaussian. The co-occurrence object categories in all the frames of the $k$-th segment are set as the interacting object classes for the $k$th atomic event. The parameters of the 3D geometric compatibility model (Eq.(4)) are learned using maximum-likelihood estimation with samples of the $k$-th segment. Figure 5 shows some samples of the learned atomic events.

### 3. Inference

Given a video $V$ in the time interval $\land = [1, T]$ which contains multiple events, the goal of inference is to interpret it with a graph list $G = (G_1, G_2, ..., G_Q)$. $G_q$ is the graph interpretation of video clip $V_{\land_q}$ in the time interval $\land_q$, which satisfies $\bigcup_{q=1}^Q \land_q = \land$ and $\bigcap_{q=1}^Q \land_q = \emptyset$. With graph list $G$, $V$ is segmented into multiple video clips $V = (V_1, V_2, ..., V_Q)$. The posterior probability is $p(G|V) = \prod_{q=1}^Q p(G_q|V_q)$. The energy of the video $V$ is interpreted by the graph list $G$ as

$$E(G|V) = \sum_{q=1}^Q \text{En}(G_q|V_q)$$  \hspace{1cm} (8)

$\text{En}(G_q|V_q)$ is the energy of each video clip, as defined in Eq.(1). The most likely interpretation to $V$ is computed as

$$G^* = \arg \min E(G|V)$$  \hspace{1cm} (9)

### 3.1. Dynamic Programming Beam Search

The general framework to solve Eq.(9) includes three procedures: i) in each frame, detect objects by sliding the window in 3D space and produce multiple hypothesized object detections; ii) propose multiple possible interpretations to this frame according to the human pose feature, the object detection, and the 3D spatial relations between them; iii) the temporal relations between frames are applied to optimize these proposals, and finally output the hierarchical interpretations to the video sequence. However, it is impossible to search the entire solution space of optimization because it has an exponential complexity of the video length.

We use a dynamic programming beam search algorithm (DPBS) to solve Eq.(9). The DPBS was previously used in the machine language translation [12]. We extend it to the video interpretation and exploit the characteristic of the event graph structure to accelerate the computation. The general idea is that based on the interpretations to the past video frames, we compute all the interpretations to the current frame. Then we keep part of all the current interpretations with the highest probabilities. This process iterates forward frame by frame until the video sequence ends. The DPBS is illustrated in Figure 6.

Suppose $G_{t-1}^1, ..., G_{t-1}^J$ are $J$ possible interpretation graph lists to the video sequence in the time interval $[1, t - 1]$, with the energy $E_{t-1}^1, ..., E_{t-1}^J$. They are shown as the paths from time 1 to $t - 1$ in Figure 6. We now want to compute an interpretation to the current frame at $t$, based on one of the $J$ paths, like the $j$th path (the green path in Figure 6). Suppose $a_{t-1}$ and $a_t$ are the atomic event labels of frame $I_{t-1}$ and $I_t$, respectively. Given the $j$th path $G_{t-1}^j$, there are three types of interpretation to the current frame $I_t$ (shown in the right side of Figure 6):

1. $a_t$ repeats the same atomic event with $a_{t-1}$;
2. $a_t$ is the next atomic event of $a_{t-1}$ in the same event;
3. $a_t$ is the atomic event of a new event.

In the third case, $a_t$ can be any atomic event in the given set, which makes our model able to handle the cases of event insertion, interruption, and repetition.

We append all the possible values of the node $a_t$ to $G_{t-1}^j$ according to the three types of interpretations, which generates $m_j$ new graph lists $G_1^j, G_2^j, ..., G_{m_j}^j$ to $G_{t-1}^j$. Their energy is $E_{t-1} + \Phi(I_t, l_t) + \Psi(I_{t-1}, l_{t-1})$. For all $G_{t-1}^1, ..., G_{t-1}^J$, we obtain $m_1 + ... + m_J$ possible solutions. We keep $J$ solutions $G_1^j, ..., G_J^j$ with the lowest energies $E_{t-1}^1, ..., E_{t-1}^J$ as the interpretations to the video in the interval $[1, t]$. Figure 6 illustrates the algorithm with a simplified example.

Our DPBS algorithm is an online algorithm. It interprets each frame from the beginning of the video to the end. Additional to recognize the event and the atomic event, it also detect and label the objects in each frame.

### 4. Experiment

#### 4.1. Multiview 3D Event Dataset

To evaluate our algorithm, we collect a large-scale multiview 3D event dataset. The dataset is captured using three stationary Kinect cameras simultaneously at different viewpoints around the human, which records the RGB, depth, and 3D human pose for each video frame. The events are performed by about 8 subjects in the natural indoor scenes, like hallway and library. Each subject repeats an event for
Figure 6. The dynamic programming beam search algorithm. The graph list in the dash green box is the interpretation to the video in the time interval \([1, t - 1]\). The dot edges and nodes in the right side of the figure are the interpretations to the frame \(I_{t-1}\).

Table 1. Dataset comparison. MV: multiview; SN: the number of total video sequences; ASN: the average number of sequences for each event category; AVL: the average length (frames) of each video sequence.

<table>
<thead>
<tr>
<th>Subject</th>
<th>MV</th>
<th>3D</th>
<th>SN</th>
<th>ASN</th>
<th>AVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMUH [5]</td>
<td>54</td>
<td>9</td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSRA3D [22]</td>
<td>✓</td>
<td>356</td>
<td>28</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Our Dataset</td>
<td>✓</td>
<td>3815</td>
<td>✓</td>
<td>477</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2. Event recognition accuracy comparison.

<table>
<thead>
<tr>
<th>Event</th>
<th>MT</th>
<th>HMM</th>
<th>4DH</th>
<th>4DHOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>drink with mug</td>
<td>0.51</td>
<td>0.62</td>
<td>0.72</td>
<td>0.83</td>
</tr>
<tr>
<td>call with cellphone</td>
<td>0.32</td>
<td>0.41</td>
<td>0.43</td>
<td>0.46</td>
</tr>
<tr>
<td>read book</td>
<td>0.83</td>
<td>0.73</td>
<td>0.93</td>
<td>0.95</td>
</tr>
<tr>
<td>use mouse</td>
<td>0.84</td>
<td>0.87</td>
<td>0.96</td>
<td>0.88</td>
</tr>
<tr>
<td>type on keyboard</td>
<td>0.77</td>
<td>0.89</td>
<td>0.96</td>
<td>0.97</td>
</tr>
<tr>
<td>fetch water from dispenser</td>
<td>0.82</td>
<td>0.76</td>
<td>0.90</td>
<td>0.93</td>
</tr>
<tr>
<td>pour water from kettle</td>
<td>0.68</td>
<td>0.67</td>
<td>0.89</td>
<td>1.00</td>
</tr>
<tr>
<td>press button</td>
<td>0.73</td>
<td>0.99</td>
<td>0.97</td>
<td>0.90</td>
</tr>
<tr>
<td>Overall</td>
<td>0.69</td>
<td>0.74</td>
<td>0.85</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Figure 7. Confusion matrix of 4DH and 4DHOI.

4.2. Event Recognition

Event recognition is to predict an event label for each video sequence which contains one event from the beginning to the end. To label the sequence, in the inference, we set \(Q = 1\) and use the dynamic programming beam search algorithm to compute its graph interpretation. The root of the graph is its event label.

We use two classical event recognition method as baselines - motion template (MT) [10] and traditional hidden Markov model (HMM) [16]. Similar to our pose model in Section 2, we use the 3D joint points on the arms as the input frame feature for the MT and HMM methods. All the original data is aligned with the same method as our model. We also compute the recognition accuracy of the 4DH method, which is the same algorithm as the 4DHOI except that it only uses the human pose information as input and omits the information of object interaction.

Table 2 shows that the performance of our model is better than other three methods. It outperform other methods in 6 categories of all 8 event categories, and improves the overall accuracy greatly, which demonstrates the strength of our method.
Figure 7 shows the confusion matrix of 4DH and 4D-HOI. The comparison between 4DH and 4D-HOI demonstrates the effect of human-object interaction on event recognition. For example, the human body movement in the event drink with mug and call with cellphone are highly similar. It is hard to distinguish them only by the human pose information. Incorporating the object information of mug and cellphone, the two events are better distinguished. Consider another event - pour water from kettle, it is complex in the temporal structure and human body movement because it involves the movement of both two arms and the coordination between the two arms. The object kettle has distinct appearance and only exists in the event pour water from kettle, which makes it provide strong support to this event. So when incorporating the information of kettle, the performance is significantly improved.

### 4.3. Sequence Segmentation

Sequence segmentation is to segment a long video sequence into coherent clips that each clip contains one event. Simultaneously segmenting a sequence and recognizing the events is a challenging problem. Our inference algorithm can interpret the current frame as a new event. The new event interpretation segments the video into clips which correspond to different events.

We use 10 unsegmented long event sequences to test the segmentation. Each sequence contains multiple events. Our segmentation data is challenging because many of the highly similar events successively occur in one sequence, and some events occur many times in one sequence.

We compare our 4DH model with the nearest neighbor classification (NN), which recognizes each frame independently without temporal context. We evaluate the accuracy in terms of frames compared to the ground truth. The accuracy of our 4DH is 0.783, and the accuracy of NN is 0.641. Figure 8 visualizes some segmentation results. Recognizing each frame independently produces many small incoherent clips, as the NN method shown in the Figure 8. Our 4DH incorporates the prior temporal structures of the events, which

<table>
<thead>
<tr>
<th>Object</th>
<th>mug</th>
<th>cellphone</th>
<th>book</th>
<th>mouse</th>
<th>keyboard</th>
<th>dispenser</th>
<th>kettle</th>
<th>button</th>
<th>monitor</th>
<th>chair</th>
<th>desk</th>
</tr>
</thead>
<tbody>
<tr>
<td>4DH</td>
<td>59</td>
<td>23</td>
<td>18</td>
<td>25</td>
<td>84</td>
<td>84</td>
<td>41</td>
<td>82</td>
<td>73</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>65</td>
<td>48</td>
<td>24</td>
<td>92</td>
<td>81</td>
<td>39</td>
<td>18</td>
<td>77</td>
<td>72</td>
<td>92</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Object localization accuracy (%).

Figure 9 shows some examples of object recognition and localization. (a) Ground truth. (b) HOG. (c) RDH. (d) Our 4D-HOI. The results of RDH are visualized by projecting the areas on the depth images into the 3D point cloud.

provide the contextual and duration information among successive frames. So it produces coherent segmentation and achieves better performance than NN.

### 4.4. Object Recognition and Localization

In video, object recognition is to determine the object class, which is related to the event recognition since the connection between object class and event category is hard constraint. So in this section, we mainly focus on the object localization. Different from the previous work which only localized objects in one video frame, or just recognized the pre-detected object motion, we localize the object in each video frame of the 3D point cloud (with it, the 2D location on image is available by projection). In each RGB frame, an object localization box is considered correct if it overlaps more than 0.5 with ground truth bounding box. The localization accuracy is defined as the ratio between the number of frames with correct object localization and the number of frames where the object appears in ground truth. We compare our method with method HOG [1] and RDH which uses the RGBD HOG feature [6] in a sliding window way to detect objects. We choose the detection with the maximum score as the final detection. The HOG and RDH detectors are trained for each object class. Table 3 shows the localization accuracy. Figure 9 shows some examples of object
localization.

The objects involved in the event present large appearance variance. Some objects have non-rigid structures, like books. Some objects move with the human action and present different directions, scales, and views in the motion, like mug. Some small objects are always occluded by the human body in the action, like cellphone and mouse. The HOG and RDH methods localize objects with appearance information in each frame. However, non-rigid structure, movement, occlusion, and low resolution make it hard to localize these objects by appearance. The human action information can facilitate the localization by using the temporal and human body context. So for those objects, our model significantly improves the accuracy. For those big and still objects which have regular appearance, like dispenser, though the improvement is not remarkable, our method still outperforms the baseline methods.

5. Conclusion

We proposed a 4D human-object interaction model for event and object recognition. The human-object interactions defined in 3D spatial domain boost the reliability on atomic event recognition. Ambiguities in interpreting the video frames are resolved by integrating temporal relation between frames. Through the dynamic programming beam search algorithm, we can efficiently segment the video, recognize events, and localize objects simultaneously. The experiment on our large scale multiview 3D event dataset proves the effectiveness of our method. The future work will focus on using the 4D human-object relations to estimate human pose in regular surveillance video.

Acknowledgement

The authors thank the support of grant: ONR MURI N00014-10-1-0933, DARPA MSEE project FA 8650-11-1-7149, 973 Program 2012CB316402.

References