Modeling 4D Human-Object Interactions for Joint Event Segmentation, Recognition, and Object Localization

Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu

Abstract—In this paper, we present a 4D human-object interaction (4DHOI) model for solving three vision tasks jointly: i) event segmentation from a video sequence; ii) event recognition and parsing; and iii) contextual object localization. The 4DHOI model represents the geometric, temporal and semantic relations in daily events involving human object interactions. In 3D space, the interactions of human poses and contextual objects are modeled by semantic co-occurrence and geometric compatibility. In 1D time axis, the interactions are represented as a sequence of atomic event transition with coherent objects. The 4DHOI model is a hierarchical spatial-temporal graph representation which can be further used for reasoning scene functionality and object affordance. The graph structures and parameters are learned using an ordered expectation maximization algorithm which mines the spatial-temporal structures of events from RGBD video samples. Given an input RGBD video, the inference is performed by a dynamic programming beam search algorithm which simultaneously carries out the event segmentation, recognition, and object localization. We collect and release a large multi-view RGBD event dataset which contains 3,815 video sequences and 383,036 RGBD frames captured by three RGBD cameras. The experimental results on three challenging datasets demonstrate the strength of the proposed method.

Index Terms—human-object interaction, object affordance, event recognition, sequence segmentation, object localization.

1 Introduction

In this paper, we present a 4D human-object interaction (4DHOI) model for solving three vision tasks jointly: i) event segmentation from a video sequence; ii) event recognition and parsing; and iii) contextual object localization in the scenes. Learned from RGBD videos, our 4DHOI model is a hierarchical spatial-temporal graph. It represents the geometric, temporal and semantic relations in daily events involving human object interactions. In comparison with the extensive research which often study these tasks in separation [1], [2], [3], [4], [5], [6], our method for joint modeling and inference is motivated by the following observations.

Firstly, many objects in daily scenes, such as the hand-hold objects in Fig. 1, are hardly detected or recognized due to heavy occlusions and appearance variations. They need high level contextual information of human-object interactions. An extreme example is the capability of human vision in understanding a 'pantomime' where the actors pretend to use objects which do not appear and are hallucinated by the observers. Recognizing other functional objects, like tables, chairs etc., also relies on the contexts of human actions.

Secondly, daily events can often be hierarchically divided into sequences of atomic events defined by human poses and contextual objects. The objects and relative time duration of the atomic events also contribute to the recognition of events. For example, the events *drink with mug* and *call with cellphone* are hardly distinguished by poses and motion features, because they are both performed by the upper body parts and have similar motion patterns. They can be told apart by the subtle difference between the appearance of *mug* and *cellphone*, and the subtle time duration of the action, i.e. a phone call often takes longer time than a sip.

Fig. 1: Examples of objects in video frames. These objects are hardly recognized by features inside the bounding boxes for heavy occlusions and large appearance variations, but can be recognized in the context of actions.

Thirdly, there is an increasing interest for reasoning object functionality by affordance in the recent literatures [7], [8], [9]. The concept of affordance has not been formally formulated. Our 4DHOI representation is an attempt to describe affordance by the hierarchical spatial-temporal graph involving 3D human poses and contextual objects in events.

In this paper, we will demonstrate that the 4DHOI model improves, significantly, the performance of each individual task through joint inference. As Fig. 2 shows, the input is a RGBD video with 3D human skeletons from Kinect camera [10]. The output includes three parts: i) hierarchical graphs segmenting the video into events and atomic events; ii) event category labels; and iii) the contextual objects localized in the RGBD frames.

We design a dynamic programming beam search algorithm for the joint inference. Based on the human pose, the searched objects, the relations between them, and the interpretations to the past frames, all the possible interpretations

- *Ping Wei and Nanning Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Shaanxi, 710049 China. E-mail: pingwei.pw@gmail.com, nnzheng@mail.xjtu.edu.cn.*
- *Yibiao Zhao and Song-Chun Zhu are with the Department of Statistics, University of California, Los Angeles, CA, 90095 USA. E-mail: {yibiao.zhao,sczhu}@stat.ucla.edu.*

Manuscript received XX, 2015; revised XX, 2015.

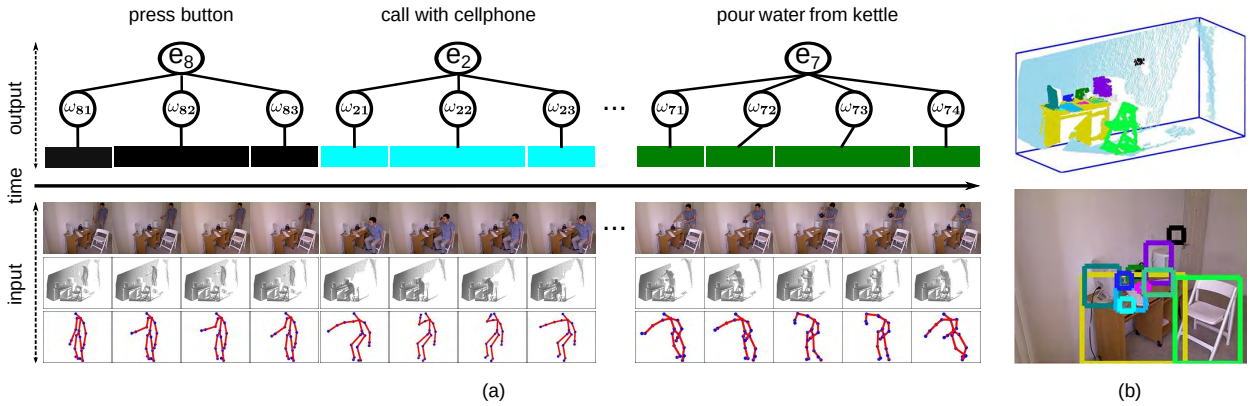

Fig. 2: The 4DHOI model. (a) The framework of the model. The inputs are the RGBD videos and human skeletons, and the outputs are the hierarchical interpretations to the video sequence, including the event recognition, segmentation, and object localization. (b) Object recognition and localization in the 3D point cloud (upper) in colored segmentation and RGB image (lower) in 2D colored bounding boxes through the 4DHOI model after analyzing the video events.

to the current frame are proposed. The interpretations with small probability are pruned. This process iterates forward frame by frame until the video ends.

To learn the spatial-temporal graph structure and the model parameters, we propose an ordered expectation maximization algorithm (OEM). Different from the conventional EM [11], OEM incorporates temporal orders of video frames and temporal alignments of atomic events into clustering. It therefore produces temporally-continuous clusters.

We collect a large-scale multiview RGBD event dataset and has released it to public[1]. It is captured by three stationary Kinect cameras from different viewpoints simultaneously. It includes 8 event categories, 11 object classes, 3,815 event videos, and 383,036 RGBD frames. We test our method on this dataset and other two challenging datasets by the event recognition, segmentation, and object localization. The experimental results prove the strength of our method.

This paper is an extension of a previous conference paper in ICCV 2013 [12]. The extension includes two main aspects: i) in methodology, we have reformulated the 4DHOI model in Section 3, introduced the learning algorithm in Section 5, added implementation details in Section 4.2, and discussed new strategies for scene alignment and object search in Section 6; ii) in experiments, we re-trained and tested the model on more data, added more comparison results, and obtained improved performances with new figures.

2 Related Work and Our Contributions

In this section, we briefly review related work from four streams of research, and discuss our contributions compared with the existing work.

2.1 Action Modeling in RGBD Data

In recent years, portable RGBD cameras, like Kinect with 3D pose estimation [10], have motivated a new wave of studying 3D human poses, actions and affordance from RGBD videos [13], [14], [15], [16], and utterly changed the landscape of action modeling and recognition. In this wave, action recognition has usually been posed as a classification problem [13], [1], [17], where the feature vector is extracted from pre-segmented videos and then classified into an action category. In a most related work of this stream, Wang *et al* [13] designed features to represent 3D pose sequence and mined the actionlet ensemble to classify actions. Different from this method, our work represents skeleton features in each frame and overcomes the noise and ambiguity by incorporating object interactions and temporal relations. The methods in [1] and [17] matched action templates with video sequence for recognition. These methods need the event clips to be given or pre-segmented, which is ineffective in real applications like video surveillance. Moreover, these methods do not interpret the objects involved in actions. In addition to recognizing actions, our method segments the video sequence, and recognizes the objects in each frame.

To understand long unsegmented video sequences, some existing work combined action recognition and segmentation or detection [2], [18], [19], [20], [21], [22]. Lv *et al* [2] used hidden Markov model (HMM) to describe each action class and a dynamic programming method to segment and recognize actions. Shi *et al* [21] described temporal boundaries and addressed action segmentation and recognition in a discriminative way. These work modeled actions with simple temporal structures, like *walk* and *run*. They did not model the interactions between human and objects.

Explicitly modeling the inner structure of actions contributes to the action recognition [23], [24], [2], [25], [26], [27], [28]. HMM [23] is usually applied to describing the state transitions [2], [25] between frames or action snippets. Tang *et al.* [24] introduced duration to HMM. Pei *et al.* [26] represented an action with atomic actions and employed a temporal filter embedded on an And-Or graph for video parsing. Sung *et al.* [28] decomposed the human activity into sub-activities and solved the model under the dynamic programming framework. Inspired by those work, our model integrates the human action, objects, and their interaction relations into a unified framework.

1. http://vcla.stat.ucla.edu/download.html

2.2 Object Modeling and Localization

Most existing work for object detection represents objects with appearance [29], [5], [6], [30], such as HOG [5] features. Lai *et al.* [6] extended the HOG feature to RGBD images. Such features are often compromised by the low resolution, heavy occlusion, and large appearance variation. To solve such problems, contextual information was introduced into the object modeling. The methods in [31] and [32] incorporated the relations with other objects to improve detection or recognition. Zhao *et al.* [8] defined objects by integrating the function, geometry, and appearance. Gupta *et al.* [9] extended the object localization and recognition to the human-centric understanding by reasoning the human and 3D scene interaction. These methods aim at object detection or recognition in still images.

Some work aimed at recognizing and localizing objects in video sequences [33], [14]. Gupta et al. [33] labeled the object according to the human actions in videos. The method in [14] tracked the location of object in each RGB video frame. In comparison, our method localizes the object both in 3D point cloud and 2D images, and does not need the accurate initialization of object locations like in [14].

2.3 Human-object Interaction and Affordance

Many researchers have recently applied human-object mutual contexts to the event, object, and scene modeling [9], [33], [34], [14], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45]. Gupta *et al.* [33] combined spatial and functional constraints between human and objects to recognize actions and objects. Prest *et al.* [38] inferred the spatial information of objects by modeling the 2D geometric relations between human body and objects. Yao *et al.* [40] modeled the relations between actions, objects, and poses in still image for detecting objects. These methods define the human-object interactions on 2D image. Such contextual cues are often compromised by the viewpoint changes and occlusions.

Koppula *et al.* [14] modeled the relations between human activity and object affordance, and their changes over time. This method needs the video to be pre-segmented. And the object detection is independent of the contextual feedback from human actions. Differently, our model incorporates the event recognition, sequence segmentation, and object localization into a unified framework, under which these tasks mutually facilitate each other.

Human-object interactions were also applied in the robotics [46], [47], [48]. Aksoy *et al.* [46] recognized manipulations by learning object-action semantics. The work [47] built ontology to model manipulation types which were applied to robot execution. This stream of research demonstrates the significance of human-object interactions from the perspective of robot learning and execution.

2.4 Action Structure Learning

Many existing approaches mined the action structures by explicitly modeling the latent structures [4], [2], [24], [26], [49]. HMM [2] learned the hidden states and transition probabilities with maximum likelihood estimation. Hidden conditional random field (HCRF) [49] learned the hidden structures of action in a discriminative way. These methods define the temporal structures on frames or segmented windows of fixed size, which can not effectively characterize and utilize the duration information of hidden structures.

Yao *et al.* [40] defined atomic poses in still images and learned them by clustering the pose samples. Pei *et al.* [26] defined relations in video clips, which were clustered to learn the event And-Or grammar. Zhou *et al.* [4] segmented an action sequence into motion primitives with hierarchical cluster analysis. Those clustering methods are under the framework similar to the expectation-maximization (EM) clustering [11]. Conventional EM clustering does not consider the temporal orders of the frames in the sequence, which may produce undesirable clustering results. For example, as is shown in Fig. 3, the poses of *approach the dispenser* and *leave the dispenser* are very similar. Without considering the temporal order, these two poses may be clustered into the same cluster. Though the work [4] introduced the temporal order into clustering, it did not consider the mutual constraints among the sequences of the same category and carried out the frame clustering for each independent sequence.

2.5 Our Contributions

In comparison with the previous work, this paper makes four main contributions.

1. It presents a 4D human-object interaction model as a stochastic hierarchical spatial-temporal graph, which represents the 3D human-object relations and the temporal relations between atomic events in RGBD videos.
2. It develops a unified framework for joint inference of event recognition, sequence segmentation, and object localization.
3. It proposes an unsupervised algorithm to learn the latent temporal structures of events and model parameters from the sequence samples.
4. It tests the model on three challenging datasets and the performances outperform the existing methods.

3 4D Human-Object Interaction Model

As Fig. 3 illustrates, the 4DHOI model is a hierarchical graph for an event. In the time axis, an event is decomposed into several ordered atomic events. For example, the event *fetch water from dispenser* is decomposed into three sequential atomic events - *approach the dispenser*, *fetch water*, and *leave the dispenser*. An atomic event corresponds to a continuous segment in the video sequence which contains similar human poses and object interactions. These atomic events are treated as hidden variables, which will be learned through mining and clustering the sequence samples.

In 3D space, an atomic event is decomposed into a specific pose, one or multiple objects involved, and the semantic and 3D geometric relations between the pose and objects. The semantic relation between the object class and a specific atomic event is treated as hard constraint. For example, the atomic event *fetch water* consists of the pose *fetch* and the objects *dispenser, mug*, as is shown in Fig. 3.

We formulate the 4DHOI with a stochastic hierarchical graph, which is similar to the And-Or Graph [50]. Suppose $V = (f_1, ..., f_\tau)$ is an video sequence in the time interval

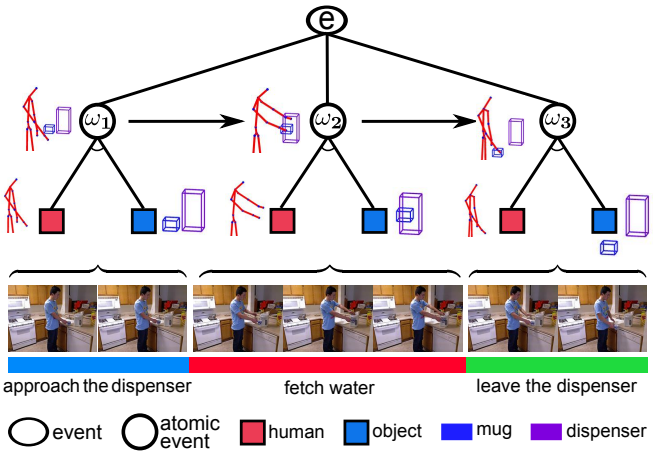

Fig. 3: A hierarchical graph of the 4D human-object interactions for an example event *fetch water from dispenser*.

$[1, \tau]$, where $f_t = (I_t, h_t)$ is the frame at time t. I_t is the RGBD data. h_t is the human pose feature defined on the 3D skeletons estimated by the motion capture technology [10].

The sequence V is interpreted by the hierarchical graph $G =< E, L >$, as follows.

i) $E \in \Delta = \{e_i | i = 1, ..., |\Delta|\}$ is the event category like *fetch water from dispenser*. Δ is the set of event categories.

ii) $L = (l_1, ..., l_\tau)$ is a sequence of frame labels. $l_t = (a_t, o_t)$ is the interpretation to the frame f_t. $a_t \in \Omega_E = \{\omega_i | i = 1, ..., K_E\}$ is the atomic event label like *fetch water*.

Ω_E is the atomic event set of E. Each event category e_i has its distinct atomic event set Ω_{e_i}, i.e. the relations of an event and its atomic events are hard constraints.

$o_t = (o_t^1, ..., o_t^{n_t})$ are the objects interacting with human at time t, where n_t is the number of objects. Each object has a class label and 3D location.

Similar to the graphical formulation in [50], the energy that the video V is interpreted by the graph G is defined as

$$\text{En}(G|V) = \sum_{t=1}^{\tau} \Phi(f_t, l_t) + \sum_{t=2}^{\tau} \Psi(l_{1:t-1}, l_t) \quad (1)$$

$\Phi(\cdot)$ is the spatial energy term of single frame, encoding the human-object interactions in 3D space.

$\Psi(\cdot)$ is the temporal energy term encoding the temporal relation between the current frame l_t and all previous frames $l_{1:t-1}$. This is different from the conventional HMM. The variable E is omitted in the right side of Eq. (1) for each event has its own distinct atomic event set.

In the following subsections, we will elaborate on the two energy terms.

3.1 Human-object Interactions in 3D Space

$\Phi(f_t, l_t)$ in Eq.(1) describes the human-object interactions in 3D space, which includes the two aspects:

1) semantic co-occurrence between a specific type of human pose and the object classes; and
2) geometric compatibility describing the 3D spatial constraint between the human body and objects.

Thus, $\Phi(f_t, l_t)$ is further decomposed into three terms which will be defined in the remaining of the subsection.

$$\Phi(f_t, l_t) = \phi_1(a_t, h_t) + \phi_2(a_t, o_t, I_t) + \phi_3(a_t, h_t, o_t) \quad (2)$$

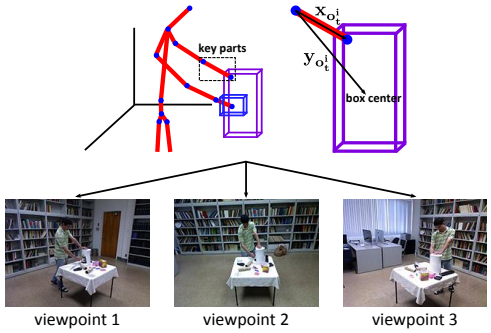

Fig. 4: Human-object geometric relations in 3D space.

3.1.1 Pose Model

$\phi_1(a_t, h_t)$ is the human pose model. A human skeleton consists of multiple 3D joints estimated by the motion capture technology, like Kinect camera [10]. To normalize the data, we align all the skeletons to a reference skeleton so that the torsos and shoulders of all the skeletons have the same locations, scales, and directions.

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. A feature vector containing the features of joints on human body is extracted and processed with PCA to reduce the correlation and noise. h_t is the PC parameter vector. It is assumed to follow a Gaussian distribution for each atomic event. Then

$$\phi_1(a_t, h_t) = -\ln N(h_t; \mu_{a_t}, \Sigma_{a_t}),$$

where μ_{a_t} and Σ_{a_t} are respectively the mean and the covariance in the atomic event a_t.

3.1.2 Contextual Object Model

$\phi_2(a_t, o_t, I_t)$ is the term for fitting contextual objects o_t to the RGBD data I_t. Each object o_t^i includes a class label, e.g. *mug*, and a 3D bounding box located at z_t^i in 3D space. The 3D box is projected into the RGB and depth images to form 2D bounding boxes, in which the RGBD HOG features [5], [6] are extracted. Let $s(z_t^i)$ be the score of linear SVM object detector using the RGBD HOG features at location z_t^i. We convert this score function into a probability using Platt scaling [6], [51], i.e. $p(z_t^i|I_t) = 1/\{1 + \exp\{\mu s(z_t^i) + \nu\}\}$, where u, v are parameters. Then

$$\phi_2(a_t, o_t, I_t) = -\frac{1}{n_t} \sum\nolimits_{i=1}^{n_t} \ln p(z_t^i|I_t) \quad (3)$$

n_t is the number of objects. $1/n_t$ offsets the influence of different object numbers. Because the relation between a_t and the object label is hard constraint, we omit a_t in the right side of Eq. (3) for clarity.

Our model defines object location in the 3D space, and appearance in the 2D image, which are more robust to the viewpoint and scale changes. The 3D boxes also provide a natural way for defining the 3D geometric relations between objects and the human poses.

3.1.3 3D Geometric Compatibility and Object Prediction

The third energy term $\phi_3(a_t, h_t, o_t)$ measures the geometric relations between human and objects. Geometric relations

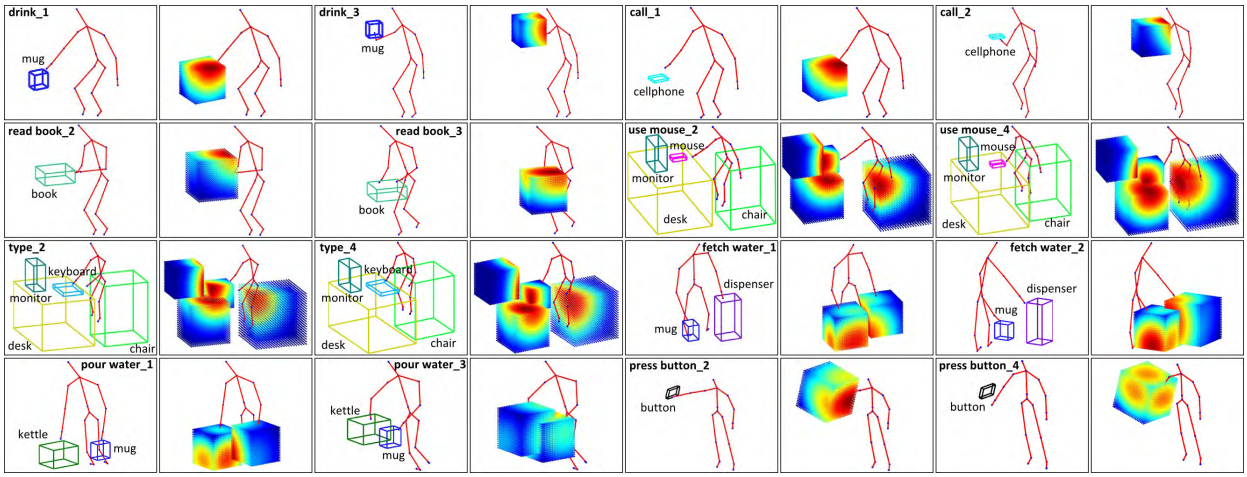

Fig. 5: Examples of learned geometric relations in atomic events. The odd number columns are the instances of learned atomic events. The indexes denote the atomic event number in each event. The even number columns are the probability maps of object prediction, where warmer colors indicate larger probabilities of the locations where objects appear.

are mostly defined on the 2D image plane [38], [40], and not suitable for different viewpoints, as Fig. 4 illustrates. We model the geometric relations in 3D space.

Geometric relations between human and objects are time-varying in different atomic events. For the example in Fig. 3, the human hand is far from the dispenser in the atomic event *approach the dispenser* while touches it in *fetch water*. So each atomic event has different geometric relations.

In an atomic event, an object interacts with some body part, which we call the key part. For example, in Fig. 4, the left arm interacts with the dispenser and the right arm interacts with the mug. The locations of objects in 3D space are closely related to and largely revealed by the locations and orientations of the key parts.

As is shown in Fig. 4, suppose $y_{o_t^i}$ is the difference vector from one joint of the key parts to the object bounding box center. $x_{o_t^i}$ is the difference vector between the end points of the key parts. $y_{o_t^i}$ is closely related to $x_{o_t^i}$. We define $\eta_{o_t^i} = y_{o_t^i} - W_{o_t^i}^{a_t} x_{o_t^i}$, where $W_{o_t^i}^{a_t}$ is a similarity transformation matrix. $\eta_{o_t^i}$ describes the location of the object relative to the key parts. We assume $\eta_{o_t^i}$ follows the Gaussian distribution. This formulation is motivated by the observation that for an atomic event the instances of an object category have similar locations relative to the key parts. The 3D geometric relation is modeled as:

$$\phi_3(a_t, h_t, o_t) = -\frac{1}{n_t} \sum_{i=1}^{n_t} \ln N(\eta_{o_t^i}; \mu_{o_t^i, a_t}^R, \Sigma_{o_t^i, a_t}^R) \quad (4)$$

where $\mu_{o_t^i, a_t}^R$ is the mean and $\Sigma_{o_t^i, a_t}^R$ is the covariance. The sign R is used to differentiate the 3D relation parameters from others. The subscript (o_t^i, a_t) indicates the geometric relation varies for different atomic events and objects.

The vector $x_{o_t^i}$ is like a local reference, by which we can estimate $y_{o_t^i}$, and therefore predict the locations of related objects. Fig. 5 illustrates some examples of atomic events and the probability maps of the learned geometric relations. As Fig. 5 shows, according to the key parts, the probability that an object appears at a 3D location can be evaluated.

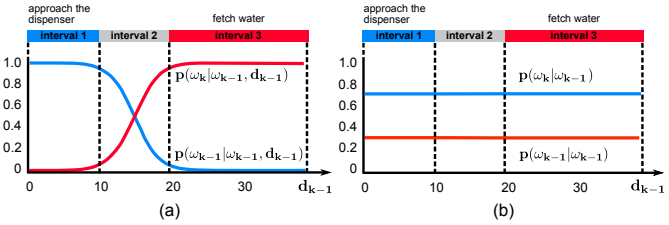

Fig. 6: The atomic event transition probability. (a) Duration-dependent transition. (b) Duration-independent transition.

3.2 Temporal Relation

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

$$\Psi(l_{1:t-1}, l_t) = \psi_1(a_{1:t-1}, a_t) + \psi_2(o_{t-1}, o_t) \quad (5)$$

$a_{1:t-1}$ are atomic event labels of the frames from time 1 to $t-1$. The first term encodes the atomic event transition, and the second term encodes the temporal coherence of objects.

3.2.1 *Atomic Event Transition*

In an event, the transition probability from the current atomic event ω_{k-1} to the next atomic event ω_k depends on the duration of the current atomic event denoted by d_{k-1}. We propose to model the time-varying transition probability with a logistic sigmoid function.

Fig. 6 compares the two kinds of transition probability. Fig. 6 (a) is a duration-dependent transition, in the interval 1 when the hand is still far from the dispenser, the probability of staying at *approach the dispenser* is much larger than the possibility of changing to the next atomic event *fetch water*. As the duration of *approach the dispenser* becomes long, in the interval 3, the probability of staying at *approach the dispenser* will be close to zero, and the transition to *fetch water* is almost 1. During the interval 2, the transition will most likely happen. In contrast, if we use a duration-independent transition probability, they will be constant, regardless of the duration, as Fig. 6 (b) shows.

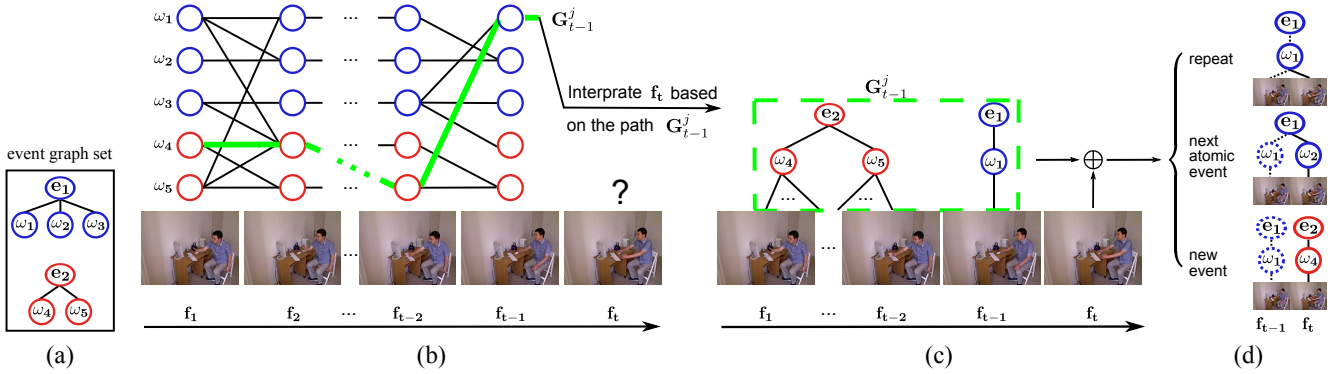

Fig. 7: The dynamic programming beam search inference algorithm. (a) Toy examples of given graph set. The goal is to interpret the input video sequences with the given graphs in this set. (b) Dynamic programming process to interpret each frame. Each path denotes one possible interpretation for the video. (c) The jth interpretation $\mathbf{G}^j_{t-1}$ for the video in the interval $[1, t-1]$ is shown in green. (d) Three types of interpretation to the frame f_t based on $\mathbf{G}^j_{t-1}$.

Denote ω_{k-1} and ω_k the two consecutive atomic events of an event E and d_{k-1} the duration of ω_{k-1} up to time $t-1$. We define the time-varying transition probability as

$$p(a_t = \omega_k | a_{t-1} = \omega_{k-1}, d_{k-1}) = \sigma(\beta d_{k-1} + \gamma). \quad (6)$$

$\sigma(v) = 1/(1+e^{-v})$ is a logistic sigmoid function with parameters β and γ. 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 ω_{k-1} is $p(\omega_{k-1}|\omega_{k-1}, d_{k-1}) = 1 - p(\omega_k|\omega_{k-1}, d_{k-1})$. Then the energy term $\psi_1(a_{1:t-1}, a_t)$ is $-\ln p(\omega_k|\omega_{k-1}, d_{k-1})$ or $-\ln p(\omega_{k-1}|\omega_{k-1}, d_{k-1})$, up to the value of a_t.

3.2.2 Temporal Coherence of Objects

$\psi_2(o_{t-1}, o_t)$ describes the temporal coherence of objects. In an event, the locations of some objects, like dispenser, are almost static, while other objects, like mugs, move with hands. For *moveable* objects, we assume the location follows a Gaussian distribution $p(z^i_t | z^i_{t-1}) = N(z^i_t - z^i_{t-1}; \mu^Z_{o^i_t, a_t}, \Sigma^Z_{o^i_t, a_t})$. For *static* objects, we set a hard threshold. If the difference of the proposed location z^i_t in the current frame and the location at the last frame z^i_{t-1} is smaller than the threshold, $p(z^i_t | z^i_{t-1})$ is 1, otherwise 0. The energy is

$$\psi_2(o_{t-1}, o_t) = -\frac{1}{n_t} \sum\nolimits_{i=1}^{n_t} \ln p(z^i_t | z^i_{t-1}) \quad (7)$$

4 Inference

Given a video $\mathbf{V}$ in the time interval $\wedge = [1, T]$ which contains Q ($Q \geq 1$) events, the goal of inference is to interpret $\mathbf{V}$ with a graph list $\mathbf{G} = (G_1, G_2, ..., G_Q)$. The graph G_q is the interpretation to the video clip $V_{\wedge_q}$ in the interval $\wedge_q$, where $\bigcup_{q=1}^Q \wedge_q = \wedge$ and $\bigcap_{q=1}^Q \wedge_q = \varnothing$. We segment $\mathbf{V}$ into Q disjoint segment $(V_1, V_2, ..., V_Q)$ and interpret $V_{\wedge_q}$ with G_q by optimizing a posterior probability,

$$p(\mathbf{G}|\mathbf{V}) = \prod\nolimits_{q=1}^Q p(G_q|V_q).$$

Or equivalently, minimizing the total energy,

$$\mathcal{E}(\mathbf{G}|\mathbf{V}) = \sum\nolimits_{q=1}^Q \mathrm{En}(G_q|V_q). \quad (8)$$

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

$$\mathbf{G}^* = \arg\min \mathcal{E}(\mathbf{G}|\mathbf{V}) \quad (9)$$

4.1 Dynamic Programming Beam Search

We use a dynamic programming beam search algorithm (DPBS) to solve Eq. (9). The DPBS was previously used in the machine language translation [52]. We improve and extend it for the hierarchical interpretation of videos. The general framework of our DPBS includes four processes:

1) searching for objects and producing multiple hypothesized object detections in the current frame;
2) proposing the possible interpretations to the current frame according to the pose feature, object detection, and 3D relations between them;
3) computing multiple graph lists and their energies of the current video with the interpretations to the past video and the current frame;
4) keeping those graph lists with larger probabilities and continuing to interpret the next frame.

The above processes iterate forward frame by frame until the video ends. The graph list with the largest probability will be the final interpretation to the video. Fig. 7 illustrates the DPBS. In the following, we will elaborate on how to compute the graph lists and their energies.

Suppose $\mathbf{G}^1_{t-1}, ..., \mathbf{G}^J_{t-1}$ are J interpretive graph lists for the video in $[1, t-1]$, with the energies $\mathcal{E}^1_{t-1}, ..., \mathcal{E}^J_{t-1}$ respectively. They are shown as the paths from time 1 to $t-1$ in Fig. 7(b). At time t, we need to compute an interpretation to the current frame f_t, based on each of the J paths, for example, the jth path (the green path in Fig. 7(b)). Suppose a_{t-1} and a_t are the atomic event labels of frame f_{t-1} and f_t, respectively. Given the jth path $\mathbf{G}^j_{t-1}$, there are three types of interpretations to the frame f_t (shown in Fig. 7(d)):
1) a_t repeats the same atomic event with a_{t-1};
2) a_t trasits to the next atomic event in the same event;
3) a_t is the atomic event of a new event.

All the possible values of a_t are appended to $\mathbf{G}^j_{t-1}$ according to the three types of interpretations, which generates m_j new graph lists $\mathbf{G}^1_t(\mathbf{G}^j_{t-1}), ..., \mathbf{G}^{m_j}_t(\mathbf{G}^j_{t-1})$ with energy $\mathcal{E}^j_{t-1} + \Phi(f_t, l_t) + \Psi(l_{1:t-1}, l_t)$. With all $\mathbf{G}^1_{t-1}, ..., \mathbf{G}^J_{t-1}$, we obtain $m_1 + ... + m_J$ possible solutions. We keep J solutions $\mathbf{G}^1_t, ..., \mathbf{G}^J_t$ with the lowest energies $\mathcal{E}^1_t, ..., \mathcal{E}^J_t$ as the possible interpretations to the video in the interval $[1, t]$.

We use a simplified example shown in Fig. 7 to illustrate the algorithm. $\mathbf{G}^j_{t-1}$ is the jth interpretation to the video

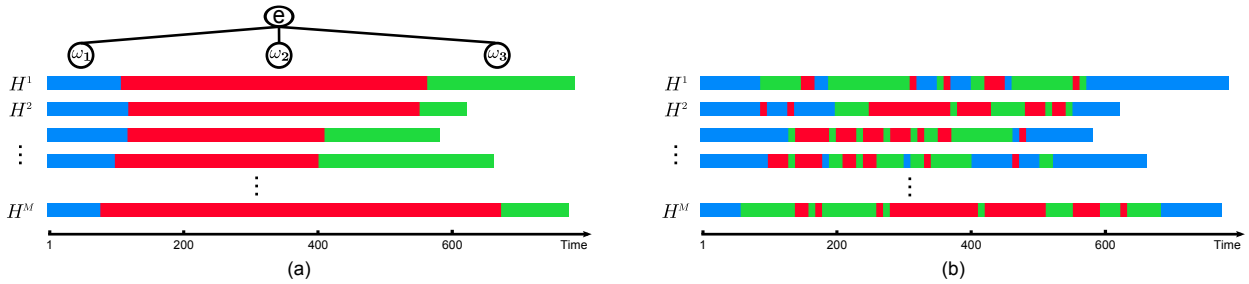

Fig. 8: Comparison between OEM and EM. (a) Our OEM. Each color denotes an atomic event. (b) Conventional EM.

in the interval $[1, t-1]$. Based on $\mathbf{G}^j_{t-1}$, there exist three interpretations to the current frame f_t. Appending them to $\mathbf{G}^j_{t-1}$ produces three possible interpretations to the video in $[1, t]$, i.e. $\mathbf{G}^1_t(\mathbf{G}^j_{t-1}), \mathbf{G}^2_t(\mathbf{G}^j_{t-1}), \mathbf{G}^3_t(\mathbf{G}^j_{t-1})$, and $m_j = 3$. With all $\mathbf{G}^1_{t-1}, ..., \mathbf{G}^J_{t-1}$, there totally exist $m_1 + ... + m_J$ possible interpretations to the sequence in $[1, t]$.

4.2 Implementation Details

Suppose N_a is the number of all atomic events. Without any constraint, there are totally ${N_a}^T$ interpretations to a sequence of the length T. It is incalculable considering T is often larger than 100. With the beam search number J, the complexity of our algorithm is $O(JTN_a)$. For example, if $J = 200$ and each event contains 4 atomic events, there are about 6400 interpretations to a video from the dataset with 8 event categories. It can be computed efficiently.

Event recognition is to predict an event category for a video. By setting $Q = 1$, DPBS computes an interpretive graph for the video. The graph root is the event label. Sequence segmentation is to cut a long video sequence into coherent segments that each segment corresponds to an event. DPBS can interpret a frame as a *new event*, at which the sequence is cut into segments of different events.

Object localization is to determine the object location both in the 3D point cloud and 2D image of each video frame. SVM-trained detectors (as defined in Section 3.1.2) firstly scan each frame to generate multiple hypothesized detections. Then the object detection score, human pose context, and temporal coherence are incorporated to maintain the locations with large probabilities.

5 Learning

In this section, we will elaborate on how to learn the hierarchical structure of the event, atomic events, temporal relations, and model parameters.

5.1 Ordered Expectation-Maximization Learning

To learn the hierarchical structure of an event category, each sample video of the category should be cut into disjointed segments so that each segment corresponds to an atomic event, as is shown in Fig. 8(a). However, the event structure is hidden. Though EM is widely-used for unsupervised clustering, it clusters frames without considering the temporal orders, and thus can not produce continuous segments, as is shown in Fig. 8(b). We propose an ordered expectation maximization algorithm (OEM)to learn the event structures and model parameters, as is shown in Fig. 8(a).

As Section 3.1.1 states, the human pose feature of each frame follows Gaussian distribution. From another perspective, we can assume each frame pose feature in a video follows Gaussian mixtures and each component of the mixtures corresponds to an atomic event. This assumption is based on and characterizes the fact that human poses of an atomic event in a video and different videos are similar.

Suppose $\{H^m | m = 1, 2, ..., M\}$ are M video samples of an event category , as the color bars in Fig. 8. Each sequence is composed of T_m frames $H^m = (h^m_1, h^m_2, ..., h^m_{T_m})$, where h^m_t is the frame pose feature at time t in the mth sequence. With these samples, the goal is to cut each sequence H^m into K disjointed segments so that the frames in the kth segment belongs to the kth atomic event, as shown in Fig. 8(a). The number K is decided empirically and an event typically has $K = 3, 4$ atomic events in our experiments.

To achieve this goal, we introduce latent variables $\mathbf{s}^m = (s^m_2, ..., s^m_K)$, and two constants $s^m_1 = 1, s^m_{K+1} = T_m + 1$ for each sequence H^m, where $s^m_k \in \{1, ..., T_m, T_m + 1\}$ is the time boundary of the segment, and $s^m_k < s^m_{k+1}$. These latent variables cut the sequence H^m into K disjointed segments. The kth segment starts at the time s^m_k and ends at the time $s^m_{k+1} - 1$. The frame pose feature in the kth segment follows the kth component distribution $\mathcal{N}(h^m_t | \mu_k, \Sigma_k)$ of the Gaussian mixtures. For all the sequence samples $\{H^m | m = 1, 2, ..., M\}$, the likelihood function is

$$L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{S}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \prod_{t=s^m_k}^{s^m_{k+1}-1} \tau_k \mathcal{N}(h^m_t | \mu_k, \Sigma_k) \qquad (10)$$

where $\boldsymbol{\tau} = (\tau_1, ..., \tau_K)$, $\boldsymbol{\mu} = (\mu_1, ..., \mu_K)$, $\boldsymbol{\Sigma} = (\Sigma_1, ..., \Sigma_K)$. $\mathbf{S} = (\mathbf{s}^1, ..., \mathbf{s}^M)$ is the latent variable set of all sequences. The optimal parameters are computed as

$$(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{S})^* = \arg\max L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{S}) \qquad (11)$$

5.1.1 Optimization

It is hard to solve Eq. (11) for its exponential complexity. Our OEM optimizes Eq. (11) under the framework similar to EM. Different from EM, OEM introduces the latent segment variables to determine the sample assignment. These segment variables are unknown and should be also optimized. So in the E step, we compute the optimal temporal segmentation for each sequence to cluster the samples, instead of computing responsibilities as in EM. Given $\mathbf{S}$, we can optimize Eq. (11) with respect to $\boldsymbol{\tau}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ by Lagrange multiplier method. The OEM optimization is summarized as:

1) **Initialization.** Initialize $\mathbf{S}$ by uniformly segmenting each video sequence into K segments. With $\mathbf{S}$, initialize $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\tau}$ with Eq. (13).
2) **E step**. Compute optimal segmentation for each sequence $H^m(m = 1, ..., m)$ with current $\boldsymbol{\tau}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$.

$$(\mathbf{s}^m)^* = \arg\max_{\mathbf{s}^m} \prod_{k=1}^{K} \prod_{t=s_k^m}^{s_{k+1}^m-1} \tau_k N(h_t^m|\mu_k, \Sigma_k) \quad (12)$$

In Section 5.1.2, we will detail how to optimize Eq. (12).
3) **M step**. Re-estimate the parameters with new $\mathbf{S}$.

$$\begin{aligned} \mu_k &= \frac{1}{N_k} \sum_{m=1}^{M} \sum_{t=s_k^m}^{s_{k+1}^m-1} h_t^m \\ \Sigma_k &= \frac{1}{N_k} \sum_{m=1}^{M} \sum_{t=s_k^m}^{s_{k+1}^m-1} (h_t^m - \mu_k)(h_t^m - \mu_k)^T \\ \tau_k &= \frac{N_k}{N} \end{aligned} \quad (13)$$

where $N_k = \sum_{m=1}^{M} (s_{k+1}^m - s_k^m)$ is the frame number of the kth cluster.
4) **Evaluate** the log likelihood

$$\ln L = \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{t=s_k^m}^{s_{k+1}^m-1} \{\ln \tau_k - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2}(h_t^m - \mu_k)^T {\Sigma_k}^{-1}(h_t^m - \mu_k)\} \quad (14)$$

and check for convergence. If the convergence criterion is satisfied, stop and output the optimization results; else, return to 2) E step.

5.1.2 Computing Optimal Temporal Segmentation

The optimization space size of Eq. (12) is exponentially related to the sequence length. So it is hard to get the global optimization by exhaustive search.

We propose to optimize it with an iterative approximation programming (IAP). In an iteration, each component of $\mathbf{s}^m$ is sequentially optimized conditioned on its other components. For clarity, we denote $\mathbf{s}^m = (s_2^m, ..., s_K^m)$ as $\mathbf{s} = (s_2, ..., s_K)$, and define $\mathbf{s}^{(i)} = (s_2^{(i)}, ..., s_K^{(i)})$ as the value in the iteration step i. We denote

$$\lambda(\mathbf{s}) = \prod_{k=1}^{K} \prod_{t=s_k}^{s_{k+1}-1} \tau_k N(h_t|\mu_k, \Sigma_k) \quad (15)$$

Eq.(12) is rewritten as

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} \lambda(\mathbf{s}) \quad (16)$$

The IAP algorithm is summarized as:
1) **Initialize** $\mathbf{s}^{(0)} = (s_2^{(0)}, ..., s_K^{(0)})$ as the values in the last E step.
2) **Iterate** $i = i + 1$, optimize

$$\begin{aligned} s_2^{(i)} &= \arg\max_{s_2} \lambda(s_2, s_3^{(i-1)}, ..., s_K^{(i-1)}) \\ s_3^{(i)} &= \arg\max_{s_3} \lambda(s_2^{(i)}, s_3, s_4^{(i-1)} ..., s_K^{(i-1)}) \\ &\vdots \\ s_K^{(i)} &= \arg\max_{s_K} \lambda(s_2^{(i)}, ..., s_{K-1}^{(i)}, s_K) \end{aligned} \quad (17)$$

3) **Check** for convergence. If the convergence condition is satisfied, stop and output the $\mathbf{s}^{(i)}$ as the optimal segmentation; else, return to step 2).

5.1.3 Estimating Parameters

The pose model of the kth atomic event is the kth component of mixture Gaussian. The co-occurred object categories in all the frames of the kth segment are set as the interacting object categories for the kth atomic event. The 3D geometric relation parameters are computed by maximum-likelihood estimation with the samples of the kth segment.

5.2 Learning Temporal Relation

While learning atomic events with OEM, each sample sequence of the event E is cut into K segments. The number of frames in each segment suggests the duration of each atomic event. We use these duration samples to learn the parameters of the atomic event transition, i.e. β and γ in Eq. (6) for two neighboring atomic events ω_{k-1} and ω_k.

We use the logistic function learning strategy [11]. Given $a_{t-1} = \omega_{k-1}$, a_t can be ω_{k-1} or ω_k. Suppose d is the continuous duration of ω_{k-1} in previous frames. We introduce a $0-1$ variable b corresponding to each d. Given d, $b = 1$ if the atomic event in the next frame is ω_k; $b = 0$ if it is ω_{k-1}.

s_{k-1}^m, s_k^m, and s_{k+1}^m are the segment boundaries of ω_{k-1} and ω_k in the mth sequence. In the interval $[s_{k-1}^m, s_{k+1}^m]$, the duration d_{k-1} of the atomic event ω_{k-1} can be $1, 2, ..., s_k^m - s_{k-1}^m, ..., s_{k+1}^m - s_{k-1}^m + 1$. When $d \in \{1, 2, ..., s_k^m - s_{k-1}^m - 1\}$, the atomic event in the next frame will be ω_{k-1}, therefore $b = 0$; when $d \in \{s_m^k - s_m^{k-1}, ..., s_m^{k+1} - s_m^{k-1}\}$, the atomic event in the next frame will be ω_k, therefore $b = 1$. In this way, we obtain $s_{k+1}^m - s_{k-1}^m$ pair samples from the mth sequence: $\{(d_i, b_i)|i = 1, 2, ..., s_{k+1}^m - s_{k-1}^m\}$. Suppose we totally obtain D pair samples from all the M sequences. $\mathbf{b} = (b_1, ..., b_D)$ is the label set. The likelihood function is

$$p(\mathbf{b}|\beta, \gamma) = \prod_{n=1}^{D} c_n^{b_n} (1 - c_n)^{1-b_n} \quad (18)$$

where $c_n = p(\omega_k|\omega_{k-1}, d_{k-1})$, as defined in Eq. (6).

The parameters β and γ in Eq.(18) can be computed by the maximum likelihood estimation.

6 Scene Alignment and Object Search

6.1 Scene Alignment

Due to the variations of camera positions and view angles, the geometric relations between human body and objects diverge in different scenes. To learn the geometric relations, all the original scenes of point cloud should be aligned. We implement it in two steps:
1) transforming the scene from the camera coordinate to the world coordinate; and
2) rotating the scene so that all the scenes have the same directions relative to the human body.

We assume the planes of wall and floor are orthogonal to each other. In the first step, the original scene is transformed so that the planes of wall and floor are parallel to the world coordinate planes. We adopt a Manhattan world method [53], [54]. We create a Delaunay triangulation of the scene

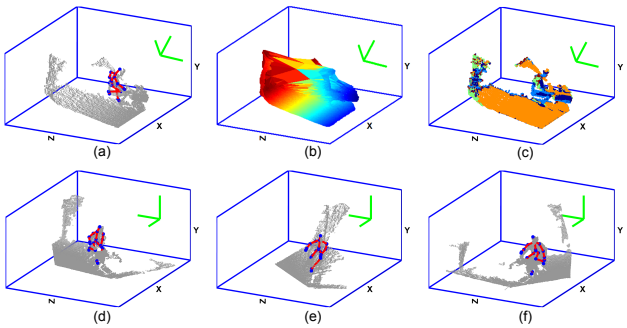

Fig. 9: Scene alignment. (a) Original scene. (b) Delaunay triangles surfaces. (c) Norm clustering. (d) Transformed scene. (e) Aligned scene. (f) Reference scene.

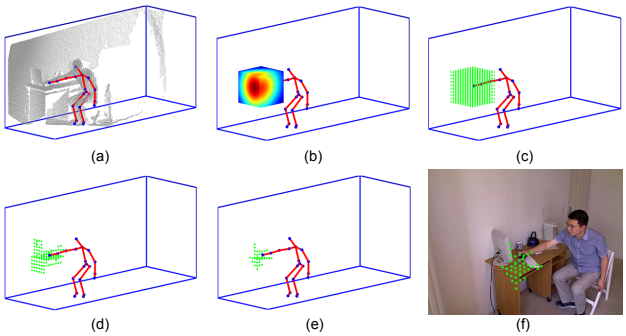

Fig. 10: Object search. (a) Point cloud. (b) Object prediction probability. (c) Potential locations. (d) Non-void locations. (e) Refined locations. (f) Final locations on 2D image.

point cloud and compute the norms of all the triangles, as shown in Fig. 9(b). These norms are clustered into several clusters with k-means in Fig. 9(c). The top three clusters with maximum samples correspond to the planes of floor and two neighboring walls. This assumption is based on the observation that the floor and wall directions are dominant in all the plane directions in the data of an indoor scene.

The three mean directions of the three clusters are orthogonalized using the Gram-Schmidt method. The three orthogonal directions are the norms of the floor and wall planes, which are aligned to the world coordinate to obtain the transformation matrix. With this matrix, the scene is transformed from the camera coordinate to the world coordinate in Fig. 9(d).

In the second step, we rotate all the scenes and human skeletons in them so that the skeletons have the same directions as the reference skeleton, as Fig. 9 (e) and (f) show.

6.2 Object Search

As Section 4.1 states, to interpret a video, the objects in each video frame should firstly be searched. We search the objects in each video frame with a sliding window strategy. On the 2D image, the conventional sliding window strategy often densely searches a great deal of invalid locations, where there is no object, like the background. It is inefficient considering the massive frames in videos. Differently, we slide the 3D window box at valid locations in the 3D space. Then the 3D window is projected into the 2D image to extract the appearance feature. This strategy largely reduces the search space and thus increases the efficiency. We implement this in three steps.

Step 1: proposing potential locations. In 3D space, we propose potential object locations inside a 3D box near the human body parts, as the green cubic area shows in Fig. 10 (c). The 3D box size and the step between locations are determined in experiments empirically.

Step 2: removing void locations. We further remove the locations at which the 3D spatial occupancy is void. Given the 3D locations from Step 1, we put a 3D box at each location and check to see if the box contains cloud points. If the number of the points is smaller than a threshold, the location is considered void and discarded. The box sizes and threshold are set empirically. After this step, the locations with 'real object entity' remains and are shown in Fig. 10(d).

Step 3: refining valid locations. In this step, the object locations are refined using the human-object geometric compatibility probabilities shown in Fig. 10 (b). Those locations with small compatibility probability are pruned. The result is shown in Fig. 10 (e). The resulted 3D locations are transformed to 2D locations on the image plane, where the objects are searched as the green area shown in Fig. 10 (f).

Through the three steps, most invalid locations are removed and the object search space is greatly reduced.

7 Experiments

7.1 Experimental Dataset

We test the proposed methods on three challenging dataset.

Multiview RGBD Event Dataset. We collect a large multi-view RGBD event dataset. The videos are captured by 3 stationary Kinect cameras simultaneously at different viewpoints around the human. Each video frame includes a RGB image at a resolution of 640×480 pixels, a depth image, and a 3D human skeleton. The events are performed by 8 actors in indoor scenes, like hallway and library. Each actor performs the events with different object instances and various styles. It includes 8 event categories: *drink with mug*, *call with cellphone*, *read book*, *use mouse*, *type on keyboard*, *fetch water from dispenser*, *pour water from kettle*, and *press button*, which involve 11 object classes: *mug*, *cellphone*, *book*, *mouse*, *keyboard*, *dispenser*, *kettle*, *button*, *monitor*, *chair*, and *desk*. Fig. 11 shows some RGB and depth frames.

We manually segment the long videos into short sequences with each segment containing one event from the beginning to the end. The labeled dataset contains 3,815 event videos and 383,036 RGBD frames. Each event category has about 477 video instances on the average.

Our dataset has several characteristics which make it challenging. Firstly, it is multiview. We use three cameras to capture the video. But due to various styles of actor's action, the viewpoint of each event is much larger than three. Secondly, the event involves various objects and the human skeletons are very noisy. Thirdly, the data has large variance due to styles of actors to perform the event. Table 1 shows the comparison of several well-known event datasets.

DailyActivity3D Dataset (Daily3D) [13] contains 320 RGBD videos and 3D human joint sequences of 16 daily activity classes: *drink*, *eat*, *read book*, *call cellphone*, *write on*

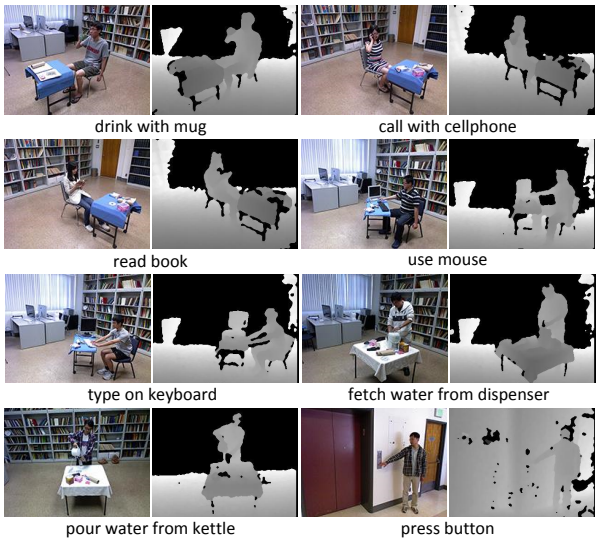

Fig. 11: Samples of Multiview RGBD Event Dataset

Subject	MV	3D	TN	AN	AL
CMUHOI [33]			54	9	110
Daily3D [13]		✓	320	20	195
MSRA3D [55]		✓	567	28	42
Our Dataset	✓	✓	3815	477	100

TABLE 1: Dataset comparison. MV, multiview; TN, total video number; AN, average video number of each event category; AL, average length (frame) of each video.

a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, and *sit down*. These activities involve interactive objects, like *mug, food package, cellphone, laptop*, etc. This dataset did not provide object labels of ground truth. To evaluate our method, we manually label the object classes and regions on each frame of all the videos with the contextual objects.

This dataset is very challenging. Firstly, the data is very noisy for occlusion and low resolution. Secondly, the activities have huge variances for the subjects perform activities in various ways.

MSR-Action3D Dataset (MSR3D) [55] contains 567 sequences performed by 10 subjects of 20 action classes : *high arm wave, horizontal arm wave,hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up & throw*. It contains depth frames from which the 3D joints are extracted.

This dataset has large number of action classes, and includes many subtle and highly-similar actions, like *draw x, draw tick, draw circle, tennis swing*, and *tennis serve*. These make the dataset very challenging.

7.2 Event Recognition

Multiview RGBD Event Dataset. We use two classical event recognition method as baselines-motion template (MT) [1] and hidden Markov model (HMM) [23]. Since the data captured by the Kinect is noisy, the feature proposed in work [1] is not available in our input. We use 3D joint points on the arms as the input frame features for MT and HMM. For MT, we train a motion template for each event category, and match the testing samples with the templates with dynamic temporal warping distance. For HMM, we train a hidden Markov model for each event category. We also compute the recognition accuracy of 4DH, which is the same framework as 4DHOI except that it only uses the human pose information as input and omits the information of contextual object interaction.

Event	MT [1]	HMM [23]	4DH	4DHOI
drink with mug	0.51	0.62	0.64	**0.85**
call with cellphone	0.32	0.41	**0.64**	0.63
read book	0.83	0.73	0.98	**1.00**
use mouse	0.84	0.87	0.98	**1.00**
type on keyboard	0.77	0.89	**0.98**	**0.98**
fetch water from dispenser	0.82	0.76	0.90	**0.95**
pour water from kettle	0.68	0.67	0.86	**0.98**
press button	0.73	**0.99**	0.97	0.95
Overall	0.69	0.74	0.87	**0.92**

TABLE 2: Event recognition accuracy comparison on Multiview RGBD Event Dataset.

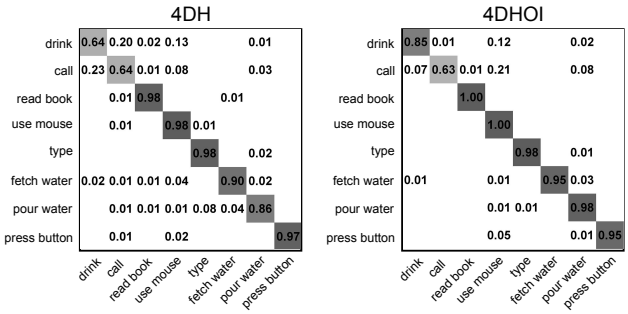

Fig. 12: Confusion matrix of 4DH and 4DHOI on Multiview RGBD Event Dataset. The event names are simplified.

Tabel 2 shows that the performance of our method is better than other three methods. It outperforms other methods in 6 categories of all 8 event categories, and improves the overall accuracy, which demonstrates the strength of our joint modeling and inference.

Fig. 12 shows the confusion matrices of 4DH and 4DHOI. The comparison between 4DH and 4DHOI proves 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 with 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 movements because it involves the movement of both two arms and the coordination between them. The object *kettle* has special appearance information 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.

DailyActivity3D Dataset. On this dataset, we compare our method with three other recent methods - dynamic temporal warping (DTW) [1], random occupancy pattern (ROP) [56], and actionlet ensemble on joint features (ALEJ) [13].

Method	DTW [1]	ROP [56]	ALEJ [13]	4DH	4DHOI
Accuracy	0.54	0.64	0.74	0.74	**0.80**

TABLE 3: Event recognition accuracy comparison on DailyActivity3D Dataset.

Method	DTW [1]	HMM [2]	AG [55]	MIJ [57]	ALEJ [13]	4DH
Accuracy	0.54	0.63	0.75	0.47	0.69	**0.83**

TABLE 4: Event recognition accuracy comparison on MSRAction3D Dataaset.

ALEJ is proposed in the same work with the dataset [13]. Table 3 shows the comparison of the recognition accuracy.

Our 4DH and 4DHOI achieve accuracies of 0.74 and 0.80 respectively, while DTW is 0.54, ROP is 0.64, and ALEJ is 0.74. This demonstrates the strength of our method, especially considering that our method can not only recognize the event, but also segment the sequence and localize objects simultaneously, while the other three methods were particularly designed for action recognition.

MSR-Action3D Dataset. Since this dataset contains no object interactions, we compare our 4DH method with five other recently proposed methods which use the skeleton information. Table 4 shows the comparison of the overall recognition accuracy.

Our method achieves an accuracy of 0.83 on this dataset. The work action graph (AG) [55], which released the MSR-Action3D Dataset, obtains an accuracy of 0.75. The other latest methods based on skeleton joints, like Dynamic Temporal Warping (DTW) [1], Hidden Markov Models (HMM) [2], Most Informative Joints (MIJ) [57], and Actionlet Ensemble on Joint Feature (ALEJ) [13] achieve accuracy of 0.54, 0.63, 0.47, and 0.69, respectively. This comparison proves the strength of our method. It also demonstrates that our model can effectively handle the actions with subtle structures.

7.3 Sequence Segmentation

We test our sequence segmentation method on 300 long video sequences which are generated by concatenating the videos from the Multiview RGBD Event Dataset. The event number, categories, and actors in each sequence are random. Each sequence contains one to ten event instances, and the length ranges from 29 to 1409 at a frame rate of 15 fps.

Our segmentation data is challenging for several reasons. Firstly, the event instances have complex temporal structures, and the human motion in an event is not coherent. For example, the human motions in the initial and middle steps of the event *pour water from kettle* are very different, which may lead to fake segmentation boundary between the two phases. Secondly, some similar events occur in one consecutive sequence, and some events occur many times in one sequence. Thirdly, some sequences contain only one event instance, which tests the generality of the segmentation algorithm and increases the data difficulty.

To comprehensively test the segmentation algorithms, we use two criteria for evaluation. The first is the frame accuracy, which is defined as the ratio between the number of correctly labeled frames and the number of all testing frames. The second is the segment accuracy defined in

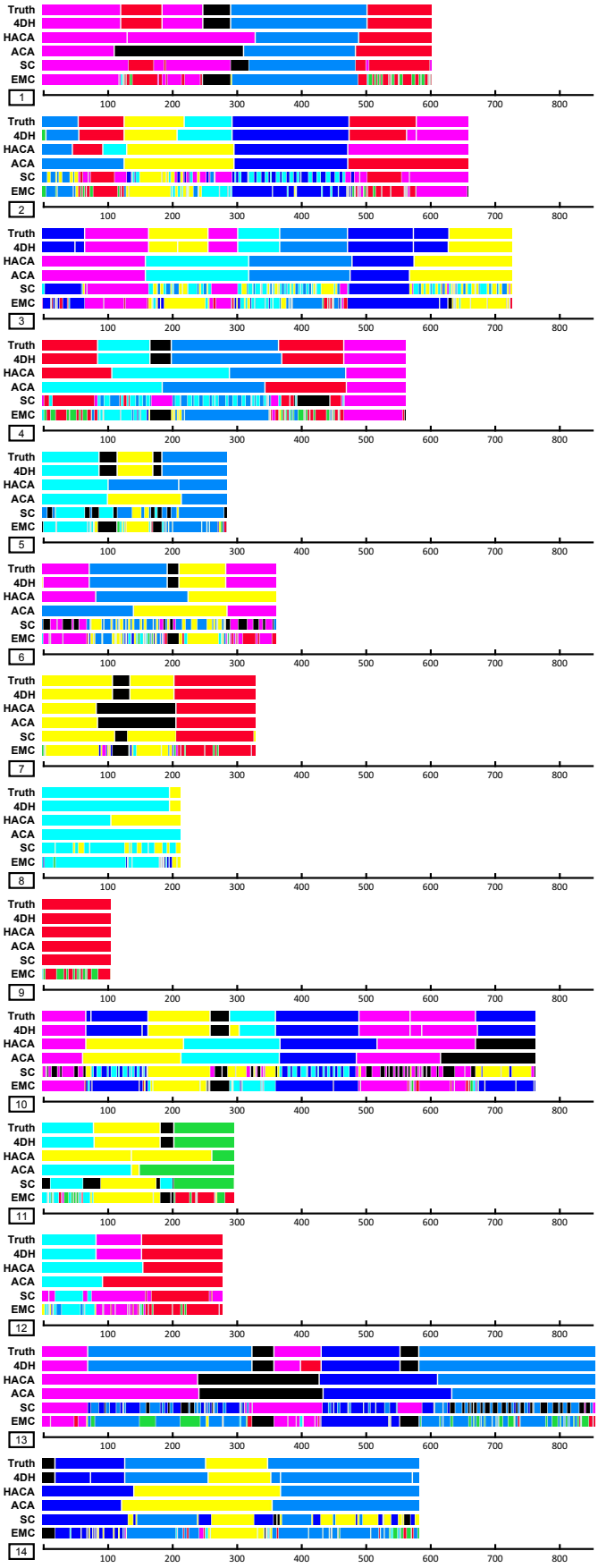

Fig. 13: Sequence segmentation comparison. Each row is a video sequence. Each color denotes an event.

Metric	EMC	SC [3]	ACA [4]	HACA [4]	Our 4DH
Segment Accuracy	0.75	0.61	0.69	0.69	**0.84**
Frame Accuracy	0.74	0.52	0.63	0.64	**0.76**

TABLE 5: Comparison of sequence segmentation algorithms with two accuracy metrics.

	mug	cellphone	book	mouse	keyboard	dispenser	kettle	button	monitor	chair	desk
HOG[5]	0.22	0.44	0.18	0.12	0.30	0.58	0.25	0.37	0.43	0.88	0.97
RDH[6]	0.29	0.48	0.19	0.13	**0.75**	0.69	0.38	**0.43**	0.69	0.96	0.97
4DHOI	**0.45**	**0.51**	**0.26**	**0.20**	**0.75**	**0.79**	**0.40**	0.39	**0.76**	**0.97**	**0.98**

TABLE 6: Average precision (AP) comparison.

[4] and is used commonly in the literature. It evaluates the algorithm by computing the segment correspondence between the testing result and the ground truth. The frame accuracy measures the local similarity between the testing result and the ground truth while the segment accuracy measures the global similarity.

We compare our 4DH model with four methods - EM Classification, Spectral Clustering (SC) [3], Aligned Cluster Analysis (ACA) [4], and Hierarchical Aligned Cluster Analysis (HACA) [4]. We use 4DH not 4DHOI because these four baselines used only human motion information, not contextual objects. To have fair comparison, we drop the object information. EMC recognizes each frame independently without temporal context. It trains the model parameters with EM algorithm and recognizes the event by Bayesian classification, where the prior distribution of each event category is assumed uniform. SC segments the sequence by maximizing the inter-cluster distance and minimizing the intra-cluster distance. It is one of the most widely used clustering algorithm. ACA incorporates the temporal order of action frames into clustering and HACA [4] is similar to ACA but solves the model in a hierarchical manner. We use the implementations of SC, ACA and HACA released in [4].

Table 5 shows the segmentation accuracy of different methods. Fig. 13 visualizes some segmentation results of the five methods. Recognizing each frame independently produces many small incoherent clips, as the EMC shown in Fig. 13. Our 4DH incorporates the temporal structures of events, which provides the contextual and duration information among successive frames. So it produces coherent segmentation and better performance than EMC. SC [3], ACA [4], and HACA [4] are unsupervised clustering methods, where the real event number and category number in each sequence should be given. Compared them, our method do not need these parameters to be given, which is advantageous in real applications.

7.4 Object Localization

Multiview RGBD Event Dataset. We use the average precision (AP) as the criterion for evaluating object localization. In each frame, we obtain many proposal positive locations, with which we compute the precision, recall and AP. This criterion uses more localized instances than the localization accuracy criterion in the conference paper [12]. We compare our method with the detection method using HOG

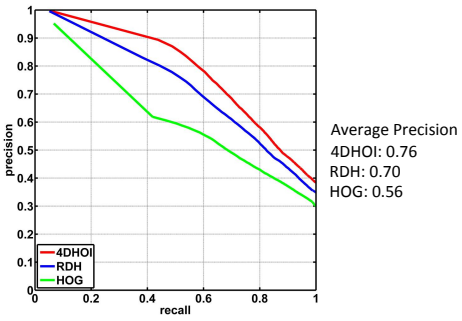

Fig. 14: Overall precision-recall comparison.

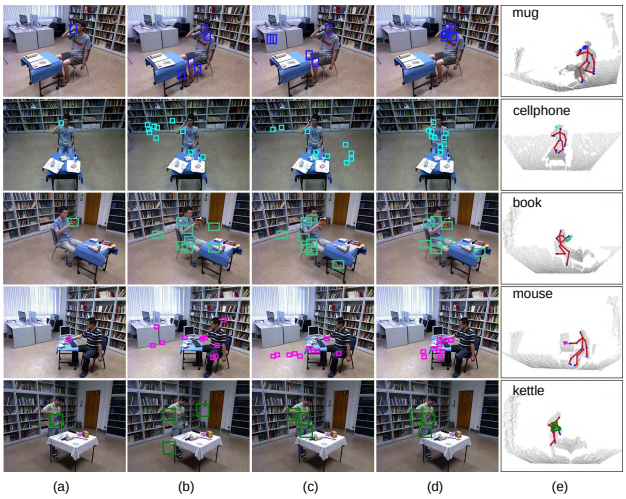

Fig. 15: Examples of object localization on Multiview RGBD Event Dataset. For each object category, we show the same number of localizations with top scores. (a) Ground truth. (b) HOG. (c) RDH. (d) 4DHOI on RGB image. (e) 4DHOI in 3D point cloud. We show one instance for clarity.

feature [5] and RDH using the RGBD HOG feature [6] which detect objects in a sliding window way. The originally detected boxes of these methods are processed with the non-maxima suppression. Table 6 shows the AP comparison of each object class. Fig. 14 shows the overall AP and precision-recall curves. Fig. 15 visualize some examples.

Those results demonstrate the strength of our model. Our method achieves the highest AP in 10 of all 11 object categories. It also largely improve the overall AP compared to other two methods. The objects involved in the event present large appearance variance. Some objects have non-rigid structures, like book. Some objects move with the human action and present different directions, scales, and views in the motion, like mug. Some small objects are often occluded by the human body in the action, like cellphone and mouse. The HOG and RDH methods localize objects with appearance information. Non-rigid structure, movement, occlusion, view variation, and low resolution lead to the low AP in Table 6. The human action information facilitates object localization by using temporal and human body contexts, and thus improves the accuracy.

For the objects keyboard and button, the performance of 4DHOI is equal to or lower than the appearance-based methods. This is for two reasons. Firstly, keyboard and

Method	**HOG** [5]	**RDH** [6]	**4DHOI**
Average Accuracy	0.41	0.51	0.70

TABLE 7: The overall average precision (AP) comparison on DailyActivity3D Dataset.

button are so thin that Kinect camera did not capture their depth. Thus the 3D geometric compatibility between human and object is not accuracy. Secondly, these objects are nearly still in the video, and thus do not fit well to the action model when the human body is constantly moving.

DailyActivity3D Dataset. We compare our method with HOG [5] and RDH [6] on the interacting objects of 9 classes - *mug*, *food packet*, *book*, *cellphone*, *laptop*, *cleaner*, *paper*, *gamebox*, and *guitar*. Table 7 shows the overall AP comparison.

The resolutions of the object instances on this dataset are low, and many instances are corrupted by the noise. Thus, the appearance-based method of HOG and RDH obtain APs of 0.41 and 0.51, respectively. When incorporating the human interactions, our 4DHOI method achieves an average prevision of 0.70. It significantly improves the performance.

8 Discussion and Future Work

In this paper, we present a 4D human-object interaction model for joint event recognition, sequence segmentation, and contextual object localization from RGBD videos. The 4DHOI model represents the geometric, temporal and semantic relations in daily events involving human object interactions. The experiments demonstrate improved performance on challenging datasets for all three tasks.

Several issues still need to be investigated in the future work. Firstly, the overall precision in object localization is still unsatisfactory. This is mainly due to the low quality of 3D human skeletons and depth data from the Kinect camera. The large noise and holes in the depth data may compromise the HOG feature and therefore weaken the object detections.

Secondly, modeling the human-object interaction in concurrent actions [15] is another interesting but challenging problem. Our 4DHOI model can potentially be applied to the concurrent actions for its part-based definition of features and relations. However, in the concurrent action, in addition to the human-object interaction, complex interactions often exist among different actions [15]. Such complicated interactions of multi-types need to be further studied.

Thirdly, how to define objects and scenes with multi-source information is a challenging problem [7], [8], [9]. This paper discusses several types of information, like appearance, affordance, and coherence. Quantitatively describing and measuring the weights of different factors are significant tasks of our future work.

The 4DHOI model is a hierarchically spatial-temporal graph representation which can be further used for reasoning scene functionality [8] and object affordance [14]. It can potentially be used for recovering 3D poses from RGB videos. These tasks will be the topics of our future work.

Acknowledgments

Ping Wei and Nanning Zheng thank the support of grants: Key Program of NSFC 61231018, NSFC 61503297, and 973 Program of China 2012CB316402. Yibiao Zhao and Song-Chun Zhu thank the support of ONR MURI N00014-10-1-0933, and DARPA MSEE FA 8650-11-1-7149.

References

[1] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Animat.*, 2006, pp. 137–146.

[2] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 359–372.

[3] J. Shi and J. Malik, "Normalized cuts and tmage segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[4] F. Zhou, F. D. la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, 2013.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[6] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3d scenes," in *Proc. Int. Conf. Robot. Autom.*, 2012, pp. 1330–1337.

[7] H. Grabner, J. Gall, and L. V. Gool, "What makes a chair a chair?" in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1529–1536.

[8] Y. Zhao and S.-C. Zhu, "Scene parsing by integrating function, geometry and appearance models," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3119–3126.

[9] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3d scene geometry to human workspace," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1961–1968.

[10] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1297–1304.

[11] C. M. Bishop, *Pattern Recognit. Mach. Learn.* Springer, 2006.

[12] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu, "Modeling 4d human-object interactions for event and object recognition," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3272–3279.

[13] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3d human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 914–927, 2014.

[14] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The Int. J. of Robot. Research*, vol. 32, no. 8, pp. 951–970, 2013.

[15] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu, "Concurrent action detection with structural prediction," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3136–3143.

[16] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2649–2656.

[17] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1234–1241.

[18] A. Bargi, R. Y. D. Xu, and M. Piccardi, "An online hdp-hmm for joint action segmentation and classification in motion capture data," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2012, pp. 1–7.

[19] M. H. Nguyen, Z.-Z. Lan, and F. D. la Torre, "Joint segmentation and classification of human actions in video," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3265–3272.

[20] A. Ali and J. K. Aggarwal, "Segmentation and recognition of continuous human activity," in *Proc. Workshop on Detection and Recognition of Events in Video*, 2001, pp. 28–35.

[21] Q. Shi, L. Cheng, L. Wang, and A. J. Smola, "Human action segmentation and recognition using discriminative semi-markov models," *Int. J. Comput. Vis.*, vol. 93, pp. 22–32, 2011.

[22] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, 2011.

[23] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, 1989.

[24] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1250–1257.
[25] B. Yao, B. X. Nie, Z. Liu, and S.-C. Zhu, "Animated pose templates for modelling and detecting human actions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 436–452, 2014.
[26] M. Pei, Z. Si, B. Yao, and S.-C. Zhu, "Learning and parsing video events with goal and intent prediction," *Comput. Vis. and Image Understanding*, vol. 117, pp. 1369–1383, 2013.
[27] N. N. Vo and A. F. Bobick, "From stochastic grammar to bayes network: probabilistic parsing of complex activity," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2641–2648.
[28] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *Proc. Int. Conf. Robot. Autom.*, 2012, pp. 842–849.
[29] L. Yang, N. Zheng, M. Chen, Y. Yang, and J. Yang, "Categorization of multiple objects in a scene using a biased sampling strategy," *Int. J. Comput. Vis.*, vol. 105, pp. 1–18, 2013.
[30] K. P. Murphy, A. Torralba, D. Eaton, and W. T. Freeman, "Object detection and localization using local and global features," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Comput. Sci., 2006, vol. 4170, pp. 382–400.
[31] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *Int. J. Comput. Vis.*, vol. 95, pp. 1–12, 2011.
[32] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[33] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: using spatial and functional compatibility for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 1775–1789, 2009.
[34] J. Gall, A. Fossati, and L. van Gool, "Functional categorization of objects using real-time markerless motion capture," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1969–1976.
[35] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit*, 2009, pp. 2929 – 2936.
[36] B. Packer, K. Saenko, and D. Koller, "A combined pose, object, and feature model for action understanding," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1378–1385.
[37] A. Prest, V. Ferrari, and C. Schmid, "Explicit modeling of human-object interactions in realistic videos," INRIA, Tech. Rep., 2011.
[38] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 601–614, 2012.
[39] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[40] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 1691–1703, 2012.
[41] D. J. Moore, I. A. Essa, and M. H. H. III, "Exploiting human actions and object context for recognition tasks," in *Proc. Int. Conf. Comput. Vis.*
[42] V. G. Kim, S. Chaudhuri, L. Guibas, and T. Funkhouser, "Shape2pose: human-centric shape analysis," *ACM Trans. on Graph.*, vol. 33, pp. 120:1–120:12, 2014.
[43] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: human actions as a cue for single view geometry," *Int. J. Comput. Vis.*, vol. 110, pp. 259–274, 2014.
[44] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros, "Scene semantics from long-term observation of people," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 284–298.
[45] H. Kjellstrm, J. Romero, and D. Kragi, "Visual object-action recognition: inferring object affordances from human demonstration," *Comput. Vis. and Image Understanding*, vol. 115, pp. 81–90, 2011.
[46] E. E. Aksoy, A. Abramov, J. Drr, K. Ning, B. Dellen, and F. Wrgtter, "Learning the semantics of objectaction relations by observation," *The Int. J. of Robot. Research*, vol. 30, pp. 1229–1249, 2011.
[47] F. Wrgtter, E. E. Aksoy, N. Krger, J. Piater, A. Ude, and M. Tamosiunaite, "A simple ontology of manipulation actions based on hand-object relations," *IEEE Trans. Auton. Mental Develop.*, vol. 5, pp. 117–134, 2013.
[48] Y. Yang, C. Fermuller, and Y. Aloimonos, "Detection of manipulation action consequences (mac)," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2563–2570.
[49] A. Quattoni, S. B. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, 2007.
[50] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. and Trends in Comput. Graph. and Vis.*, vol. 2, 2006.
[51] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, 1999.
[52] C. Tillmann and H. Ney, "Word reordering and a dynamic programming beam search algorithm for statistical machine translation," *Computational Linguistics*, vol. 29, pp. 97–133, 2003.
[53] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu, "Beyond point clouds: scene understanding by reasoning geometry and physics," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3127–3134.
[54] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1422–1429.
[55] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2010, pp. 9–14.
[56] J. Wang, J. Yuan, Z. Chen, and Y. Wu, "Spatial locality-aware sparse coding and dictionary learning," in *Proc. Asian Conf. Mach. Learn.*, 2012, pp. 491–505.
[57] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints(smij): A new representation for human skeletal action recognition," *J. of Visual Commun. and Image Representation*, vol. 25, pp. 24–38, 2014.

Ping Wei received his BE degree and PhD degree from Xi'an Jiaotong University, China, in 2007 and 2014, respectively. From Nov. 2011 to Apr. 2013, he was a visiting Ph.D student at the VCLA Center of UCLA. He is currently an assistant professor with the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include computer vision, machine learning, and cognition modeling. He is a member of IEEE.

Yibiao Zhao received a PhD degree from University of California, Los Angeles. His research interests include computer vision, cognitive modeling, cognitive robotics, statistical learning and inference. He has been working on scene parsing integrating functionality, geometry and appearance. He is the co-chair of the series of Int'l Workshops on Vision Meets Cognition: Functionality, Physics, Intents and Causality at CVPR 2014 and CVPR 2015. He is a member of IEEE.

Nanning Zheng received a PhD degree from Keio University, Japan, in 1985. He is currently a professor and the director of the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, image processing, and hardware implementation of intelligent systems. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy of Engineering in 1999. He is a Fellow of IEEE.

Song-Chun Zhu received a PhD degree from Harvard University, and is a professor with the Department of Statistics and the Department of Computer Science at UCLA. His research interests include computer vision, statistical modeling and learning, cognition and AI, and visual arts. He received a number of honors, including the Marr Prize in 2003 with Z. Tu et. al. on image parsing,the Aggarwal prize from the Int'l Association of Pattern Recognition in 2008, twice Marr Prize honorary nominations in 1999 for texture modeling and 2007 for object modeling with Y.N. Wu et al., a Sloan Fellowship in 2001, the US NSF Career Award in 2001, and the US ONR Young Investigator Award in 2001. He is a Fellow of IEEE.