This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

Watch-n-Patch: Unsupervised Learning of Actions and Relations

Chenxia Wu, Jiemi Zhang, Ozan Sener, Bart Selman, Silvio Savarese, and Ashutosh Saxena

Abstract—There is a large variation in the activities that humans perform in their everyday lives. We consider modeling these composite human activities which comprises multiple basic level actions in a completely unsupervised setting. Our model learns high-level co-occurrence and temporal relations between the actions. We consider the video as a sequence of short-term action clips, which contains human-words and object-words. An activity is about a set of action-topics and object-topics indicating which actions are present and which objects are interacting with. We then propose a new probabilistic model relating the words and the topics. It allows us to model long-range action relations that commonly exist in the composite activities, which is challenging in previous works.
We apply our model to the unsupervised action segmentation and clustering, and to a novel application that detects forgotten actions, which we call action patching. For evaluation, we contribute a new challenging RGB-D activity video dataset recorded by the new Kinect v2, which contains several human daily activities as compositions of multiple actions interacting with different objects. Moreover, we develop a robotic system that watches and reminds people using our action patching algorithm. Our robotic setup can be easily deployed on any assistive robots.

Index Terms—Unsupervised Learning, Activity Discovery, Robot Application.

1 Introduction

The average adult forgets three key facts, chores or events every day [2]. Hence it is important for a vision system to be able to detect not only what a human is currently doing but also what he forgot to do. For example in Fig. 1, someone fetches milk from the fridge, pours the milk to the cup, takes the cup and leaves without putting back the milk, then the milk would go bad. In this paper, we focus on modeling these *composite human activities* then detecting the *forgotten actions* for a robot, which learns from a completely *unlabeled* set of RGB-D videos.

A human activity is composite, *i.e.*, it is composed of several basic level actions. For example, a composite activity *warming milk* contains a sequence of actions: *fetch-milk-from-fridge*, *microwave-milk*, *put-milk-back-to-fridge*, *fetch-milk-from-microwave*, and *leave*. Modeling this poses several challenges. First, some actions often co-occur in a composite activity but some may not. Second, co-occurring actions have variations in temporal orderings, *e.g.*, people can first *put-milk-back-to-fridge* then *microwave-milk* instead of the inverse order in the above example, as its ordering is more relevant to the action *fetch-milk-from-fridge*. Moreover, these ordering relations could exist in both short-range and long-range, *e.g.*, *pour* is followed by *drink* while sometimes *fetch-book* is related to *put-back-book* with a long *read* between them. Third, the objects the human interacting with are also important to modeling the actions and their relations, as same actions often have common objects in interaction.

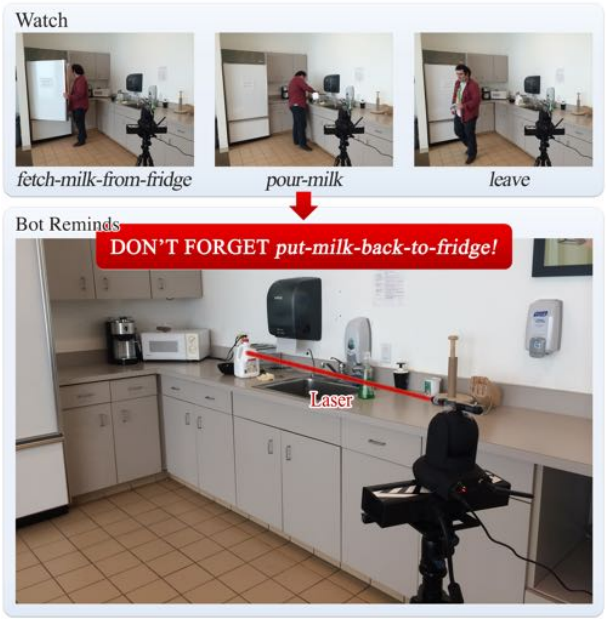


Fig. 1: Our Watch-Bot understands what human is currently doing by automatically segmenting the composite activity into basic level actions. We propose a completely unsupervised approach to modeling the human skeleton and object features to the actions, as well as the pairwise action co-occurrence and temporal relations. Using the learned model, our robot detects humans' forgotten actions and reminds them by pointing out the related object using the laser spot.

The challenge that we undertake in this paper is: Can an algorithm learn about the aforementioned relations in the composite activities when just given a completely *unlabeled* set of RGB-D

- *Wu, Sener and Selman are with the Department of Computer Science, Cornell University, Ithaca, NY 14853. E-mail: chenxiawu,ozan,selman@cs.cornell.edu*
- *Zhang is with Didi Chuxing, China. Email: jmzhang10@gmail.com*
- *Savarese is with the Department of Computer Science, Stanford University, CA 94305. Email: ssilvio@cs.stanford.edu*
- *Saxena is with Brain of Things Inc., Redwood City, CA 94062. Email: asaxena@cs.stanford.edu*

Parts of this work have been published in [62], [63] as the conference version.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

videos?

Most previous works focus on action detection in a supervised learning setting. In the training, they are given fully labeled actions in videos [33], [45], [46], or weakly supervised action labels [9], [13], or locations of human/their interacting objects [30], [39], [53]. Among them, the temporal structure of actions is often discovered by Markov models such as Hidden Markov Model (HMM) [52] and semi-Markov [17], [48], or by linear dynamical systems [5], or by hierarchical grammars [4], [29], [42], [56], [58], or by other spatio-temporal representations [23], [25], [28], [40]. Object-in-use contextual information has also been commonly used for recognizing actions [27], [28], [39], [58]. Besides relying on the manually labeling, most of these works are based on RGB features and only model the short-range relations between actions (see Section 2 for details).

Unlike these approaches, we consider a completely unsupervised setting. The novelty of our approach is the ability to model the long-range action relations in the temporal sequence, by considering pairwise action co-occurrence and temporal relations, *e.g.*, *put-milk-back-to-fridge* often co-occurs with and temporally after (but not necessarily follows) *fetch-milk-from-fridge*. We also use the more informative human skeleton features and RGB-D object features, which have shown higher performance over RGB only features for action recognition [27], [32], [64].

In order to capture the rich structure in the composite activity, we draw strong parallels with the work done on document modeling from natural language (*e.g.*, [8]) and propose a *causal topic model*. We consider an activity video as a document, which consists of a sequence of short-term action clips containing human-skeleton-trajectories as *human-words* and interacting-object-trajectories as *object-words*. An activity is about a set of *action-topics* indicating which actions are present in the video, such as *fetch-milk-from-fridge* in the *warming milk* activity, and a set of *object-topics* indicating which object types are interacting. We draw human-words from the action-topics, and object-words from both action-topics and object-topics[1]. Then we model the following (see Fig. 2):

- *Action co-occurrence.* Some actions often co-occur in the same activity and may have the same objects. We model the co-occurrence by adding correlated topic priors to the occurrence of action-topics and object-topics, *e.g.*, action-topics *fetch-book* and *put-back-book* has strong correlations and are also strongly correlated to object-topic *book*.
- *Action temporal relations.* Some actions often causally follow each other, and actions change over time during the activity execution. We model the relative time distributions between every action-topic pair to capture the temporal relations.

We first show that our model is able to learn meaningful representations from the unlabeled composite activity videos. We use the model to temporally segment videos to action segments by assigning action-topics. We show that these action-topics are promising to be semantically meaningful by mapping them to ground-truth action classes and evaluating the labeling performance.

We then show that our model can be used to detect forgotten actions in the composite activity, a new application that we call *action patching*. We enable a robot, which we call *Watch-Bot*, to detect humans' forgotten actions as well as to localize the related object in the scene. The setup of the robot can be easily deployed on any assistive robots and applied to different areas such as industry, medical work and home use. We evaluate the action patching accuracy to show that the learned co-occurrence and temporal relations are very helpful to inferring the forgotten actions. We also show that our Watch-Bot is able to remind humans of forgotten actions in the real-world robotic experiments.

We also provide a new challenging RGB-D activity video dataset [2] recorded by the new Kinect v2 (see examples in Fig. 12), in which the human skeletons are also recorded. It contains 458 videos of human daily activities as compositions of multiple actions interacting with different objects, in which people forget actions in 222 videos. They are performed by different subjects in different environments with complex backgrounds.

In summary, the main contributions of this work are:

- Our model is completely unsupervised thus being more useful and scalable.
- Our model considers both the short-range and the long-range action relations, showing the effectiveness in the action segmentation and clustering.
- We show a new application by enabling a robot to remind humans of forgotten actions in the real scenes.
- We provide a new challenging RGB-D activity dataset recorded by the new Kinect v2, which contains videos of multiple actions interacting with different objects.

The paper is organized as follows. Section 2 introduces the related works. Section 3 outlines our approach to modeling the composite activity. We present the visual features of the activity video clip in Section 4. Section 5 gives the detailed description of our learning model as well as its learning and inference. Section 6 discusses the novelties of our approach compared with the close works. Section 7 introduces our watch-bot system to reminding of forgotten actions using our learned model. We give an extensive evaluation and discussion in the experiments in Section 8. Section 9 concludes the paper.

2 Related Work

Temporal Structure Modeling in Action Recognition. Our work is related to the works on action recognition in computer vision. There is a large number of early works focusing on classifying pre-segmented action segments into an action class, which can be referred in surveys [3]. In order to model the complex human activities, most recent works introduce how to model the temporal structures in action videos.

Most previous works on action recognition are supervised [9], [13], [30], [33], [36], [40], [45], [53]. Among them, the most popular are linear-chain models such as hidden markov model (HMM) [52], semi-Markov [17], [48] and the linear dynamic system [5]. They focus on modeling the local transitions (between frames, temporal segment, or sub-actions) in the activities. More complex hierarchical relations [29], [42], [56], [58] or graph relations [4], [49] are considered in modeling actions in the complex activity. In detail, Lillo *et al*. [31] describe human activities in a hierarchical discriminative model that operates at three semantic levels, which is able to capture the spatio-temporal compositions for complex activity recognition using RBG-D data. In the model, action and pose transitions between neighboring

1. Here we consider the same object type like book can be variant in appearance in different actions such as close book in the fetch-book action and open book in the reading action.

2. The dataset and tools are released at http://watchnpatch.cs.cornell.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

pairs are considered to model the temporal relations. Wei *et al*. [60] propose a 4D human-object interaction model and evaluate on a multiview RGBD event dataset. They represent the structure of events, sub-events and objects in a hierarchical graph, where the atomic event transition and object coherence in temporal is modeled.

There are also some works focusing on detecting local action patches, primitives, trajectories or spatio-temporal features [20], [35], [37], [67] without considering the high-level action relations. There also exist some unsupervised approaches on action recognition. Yang *et al*. [67] develop a meaningful representation by discovering local motion primitives in an unsupervised way, then a HMM is learned over these primitives. Jones *et al*. [21] propose an unsupervised dual assignment clustering on the dataset recorded from two views. In [41], they present a framework for parsing video events with stochastic temporal and-or graph using unsupervised learning. They represent the temporal relations between multiple subevents by the horizontal links between the nodes. In [14], [54], they introduce a novel probabilistic activity modeling approach that mines recurrent sequential patterns called motifs from documents. They represent the documents as a mixture of sequential activity patterns where the mixing weights are defined by the motif starting time occurrences.

Although these approaches have performed well in different areas, most of them rely on local relations between adjacent clips or actions that ignore the *long-term* action relations. We model the pairwise action co-occurrence and temporal relations in the whole video, thus relations are considered globally and completely with the uncertainty. We also use the learned relations to infer the forgotten actions without any manual annotations.

RGB-D and Human Skeleton Features. We also use human skeletons and RGB-D features to better represent video clips rather than the RGB action features [22], [57]. Action recognition using human skeletons and RGB-D camera have shown the advantages over RGB videos in many works. Skeleton-based approach focus on proposing good skeletal representations [32], [46], [51], [55], [64]. Furthermore, we detect the human interactive objects in an unsupervised way to provide more discriminate features. Object-in-use contextual information has been commonly used for recognizing actions [27], [28], [39], [58]. Moreover, Hu*et al*. [18] propose a joint learning model to simultaneously learn heterogenous features from RGB-D activity videos. Most of them focus on designing or learning good action features. They lost the high-level action relations which are captured in our model.

Bayesian Models. Our work is also related to the Bayesian models. LDA [8] was the first hierarchical Bayesian topic model and widely used in different applications. The correlated topic models [6], [24] add the priors over topics to capture topic correlations. A topic model over absolute timestamps of words is proposed in [59] and has been applied to action recognition [15]. However, the independence assumption of different topics would lead to non smooth temporal segmentations. Recently, a multi-feature max-margin hierarchical Bayesian model [65] is proposed to jointly learn a high-level representation by combining a hierarchical generative model and discriminative maxmargin classifiers in a unified Bayesian framework. Differently, our model considers both correlations and the relative time distributions between topics rather than the absolute time, which captures richer information of action structures in the complex human activity.

Perception of Human Activities for Robotics. Our work is also related to the works on recognizing human actions for robotics [10], [26], [34]. Yang *et al*. [66] present a system that learns manipulation action plans for robot from unconstrained youtube videos. Hu *et al*. [19] propose an activity recognition system trained from soft labeled data for the assistant robot. Chrungoo *et al*. [11] introduce a human-like stylized gestures for better human-robot interaction. Piyathilaka *et al*. [43] use 3D skeleton features and trained dynamic bayesian networks for domestic service robots. Our robot's output laser spot on object is also related to the work 'a clickable world' [38], which selects the appropriate behavior to execute for an assistive object-fetching robot using the 3D location of the click by the laser pointer. However, it is challenging to directly use these approaches to detecting the forgotten actions and remind people.

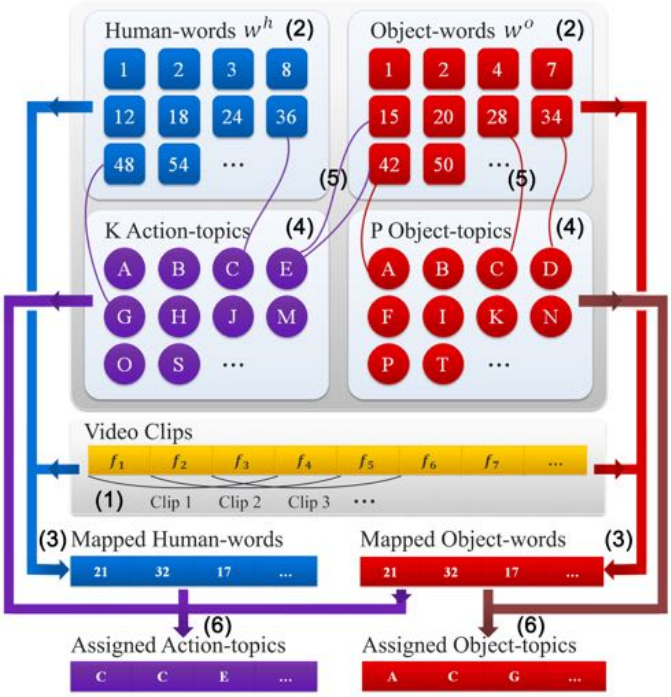


Fig. 2: Video representation. (1) A video frames (f_i) is first decomposed into a sequence of overlapping fixed-length temporal clips. (2) The human-skeleton-trajectories/interactive-object-trajectories are extracted from each clip, and we cluster them to form the human-dictionary/object-dictionary. (3) Then the video is represented as a sequence of human-word and object-word indices by mapping its human-skeleton-trajectories/interactive-object-trajectories to the nearest human-words/object-words in the dictionary. (4) An activity video is about a set of action-topics/object-topics indicating which actions are present and which types of objects are interacting with. (5) We learn the mapping of action-words/object-words to the action-topics/object-topics, as well as the co-occurrence and the temporal relations between the topics. (6) We assign the topics to clips using the learned model.

3 Overview

We present our approach pipeline in this section. Our system takes a RGB-D video with the 3D joints of the tracked human skeletons from Kinect v2 as inputs. First, a video is decomposed into a sequence of overlapping fixed-length (20 frames in our experiments) temporal clips (step (1)). Next the human-skeleton-trajectory features and the interacting-object-trajectory features are extracted from the clips (see details in Section. 4).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

We propose a compact representation of the action video (see Fig. 2) by drawing parallels to document modeling in the natural language [8]. A video is represented as a sequence of action/object words. We use k-means to cluster the human-skeleton-trajectories/interacting-object-trajectories from all the clips in the training set to form a *human-dictionary* and an *object-dictionary*, where we use the cluster centers as *human-words* and *object-words* ((2) in Fig. 2). Then, the video is represented as a sequence of human-word and object-word indices by mapping its human-skeleton-trajectories/interacting-object-trajectories to the nearest human-words/object-words in the dictionary ((3) in Fig. 2). An activity video is also about a set of *action-topics* indicating which actions are present in the video, and a set of *object-topics* indicating which object types are interacting in the actions ((4) in Fig. 2).

We propose an unsupervised probabilistic learning model that models the mapping of action-words/object-words to the action-topics/object-topics, as well as the co-occurrence and the temporal relations between the topics ((5) in Fig. 2). Using the learned model, we can assign the action-topic/object-topic to each clip. So the continuous clips with the same assigned action-topic form an action segment ((6) in Fig. 2).

Note that the unsupervised action assignments of the clips are challenging because there is no annotation during the training stage. Besides extracting rich visual features, we further consider the relations among actions and objects. Unlike previous works, our model captures long-range relations between actions *e.g.*, *put-milk-back-to-fridge* is strongly related to *fetch-milk-from-fridge* even with *pour* and *drink* between them. We model all pairwise co-occurrence and temporal casual relations between topics in a video, using a new probabilistic model (introduced in Section 5).

4 Visual Features

We describe how we extract the visual features of a clip in this section. We extract both human-skeleton-trajectory features and the interacting-object-trajectory features from the output by the Kinect v2 [1], which has an improved body tracker and the higher resolution of RGB-D frame than the Kinect v1. The tracked human skeleton has 25 joints in total. Let $X_u = \{x_u^{(1)}, x_u^{(2)}, \cdots, x_u^{(25)}\}$ be the 3D coordinates of 25 joints of a skeleton in the current frame u. We first compute the cosine of the angles between the connected body parts in each frame: $\alpha^{(pq)} = (p^{(p)} \cdot p^{(q)})/(|p^{(p)}| \cdot |p^{(q)}|)$, where the vector $p^{(p)} = x^{(i)} - x^{(j)}$ represents the body part. The transition between the joint coordinates and angles in different frames can well capture the human body movements. So we extract the motion features and off-set features [64] by computing their Euclidean distances $\mathbb{D}(,)$ to previous frame $f^m_{u,u-1}, f^\alpha_{u,u-1}$ and the first frame $f^m_{u,1}, f^\alpha_{u,1}$ in the clip:

$f^m_{u,u-1} = \{\mathbb{D}(x_u^{(i)}, x_{u-1}^{(i)})\}_{i=1}^{25}$, $f^\alpha_{u,u-1} = \{\mathbb{D}(\alpha_u^{(pq)}, \alpha_{u-1}^{(pq)})\}_{pq}$;
$f^m_{u,1} = \{\mathbb{D}(x_u^{(i)}, x_1^{(i)})\}_{i=1}^{25}$, $f^\alpha_{u,1} = \{\mathbb{D}(\alpha_u^{(pq)}, \alpha_1^{(pq)})\}_{pq}$.
Then we concatenate all $f^m_{u,u-1}, f^\alpha_{u,u-1}, f^m_{u,1}, f^\alpha_{u,1}$ as the human features of the clip.

We also extract the human interacting-object-trajectory based on the human hands, image segmentation, motion detection and tracking. To detect the interacting objects, first we segment each frame into super-pixels using a fast edge detection approach [12] on both RGB and depth images. The RGB-D edge detection provides richer candidate super-pixels rather than pixels to further extract objects. We then apply the moving foreground mask [50] to remove the unnecessary steady backgrounds and select those super-pixels within a distance to the human hands in both 3D points and 2D pixels. Finally, we collect the bounding boxes enclosing these super-pixels as the potential interested objects (see examples in Fig. 3).

Fig. 3: Examples of the human skeletons (red line) and the extracted interacting objects (green mask, left: fridge, right: book).

We then track the bounding box in the segmented clip using SIFT matching and RANSAC to get the trajectories. We use the closest trajectory to the human hands for the clip. Finally, we extract six kernel descriptors [44] from the bounding box of each frame in the trajectory: gradient, color, local binary pattern, depth gradient, spin, surface normals, and KPCA/self-similarity, which have been proven to be useful features for RGB-D data [61]. We concatenate the object features of each frame as the interacting-object-trajectory feature of the clip.

5 Learning Model

In order to incorporate the aforementioned properties of activities, we present a new generative model (see the graphic model in Fig. 4 and the notations in Table 1). We use a joint distribution to model the correlations between action/object topics, that estimates which actions and objects are most likely to co-occur in a video. We model a relative time distribution to capture the temporal causal relations between actions, which estimates the possible temporal ordering of the occurring actions in the video. The novelty of our model is the ability to capture both short-range and long-range relations between actions in the compose activity videos in an unsupervised way. Using these relations, we can simultaneously segment the video and assign the action-topics as well as infer forgotten actions.

Consider a collection of D videos (documents in the topic model). Each video as a document d consists of N_d continuous clips $\{c_{nd}\}_{n=1}^{N_d}$, each of which consists of a human-word w^h_{nd} mapped to the human-dictionary and an object-word w^o_{nd} mapped to the object-dictionary. We assign action-topic to each clip c_{nd} from K latent action-topics, indicating which action-topic they belong to. We assign object-topic to each object-word w^o_{nd} from P latent object-topics, indicating which object-topic is interacting within the clip. The assignments are denoted as $z^{(1)}_{nd}$ and $z^{(2)}_{nd}$. We use superscripts $(1), (2)$ to denote action-topics and object-topics respectively. After assignments, continuous clips with the same action-topic compose an action segment in a video. All the segments assigned with the same action-topic from the training set compose an action cluster.

The topic model such as LDA [8] has been very common for document modeling from language. We use a it to generate a video document using a mixture of topics. In order to model human actions in the video, our model introduces co-occurrence and temporal structure of topics instead of the topic independence assumption in LDA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

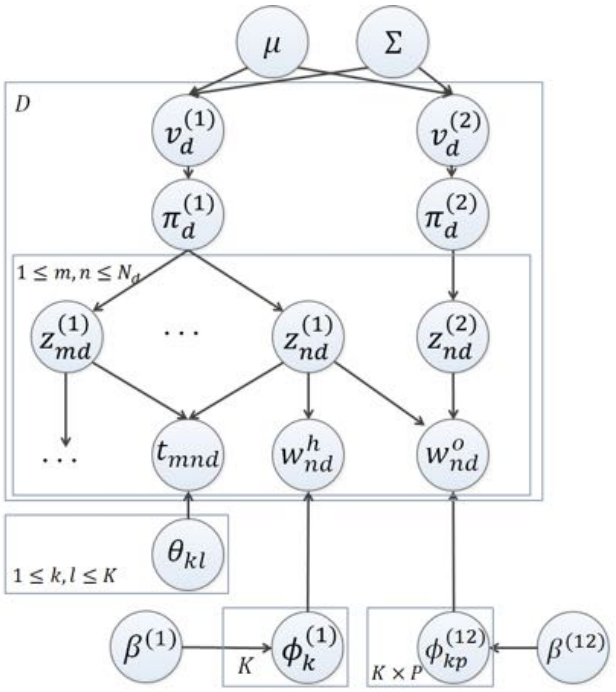


Fig. 4: The graphic model of our causal topic model.

Basic generative process. In a document d, we choose $z_{dn}^{(1)} \sim Mult(\pi_{:d}^{(1)}), z_{dn}^{(2)} \sim Mult(\pi_{:d}^{(2)})$, where $Mult(\pi)$ is a multinomial distribution with parameter π. The human-word w_{nd}^h is drawn from an action-topic specific multinomial distribution $\phi_{z_{nd}^{(1)}}^{(1)}$, $w_{dn}^h \sim Mult(\phi_{z_{dn}^{(1)}}^{(1)})$, where $\phi_k^{(1)} \sim Dir(\beta^{(1)})$ is the human-word distribution of action-topic k, sampled from a Dirichlet prior with the hyperparameter $\beta^{(1)}$. While the object-word w_{nd}^o is drawn from an action-topic and object-topic specific multinomial distribution $\phi_{z_{nd}^{(1)} z_{nd}^{(2)}}^{(12)}$, $w_{dn}^o \sim Mult(\phi_{z_{nd}^{(1)} z_{nd}^{(2)}}^{(12)})$, where $\phi_{kp}^{(12)} \sim Dir(\beta^{(12)})$ is the object-word distribution of action-topic k and object-topic p. Here we consider the same object type like *book* can be variant in appearance in different actions such as a close book in *fetch-book* and a open book in *read* action. So we consider the object-word distribution for different combinations of the action topic and the object topic.

Topic correlations. The co-occurrence such as action *pour* and action *drink*, object *book* and action *read*, is useful to recognizing the co-occurring actions/objects and also gives a strong evidence for detecting forgotten actions. We model the co-occurrence by drawing their priors from a mixture distribution. Let $\pi_{kd}^{(1)}, \pi_{pd}^{(2)}$ be the probability of action-topic k and object-topic p occurring in document d, where $\sum_{k=1}^{K} \pi_{kd}^{(1)} = 1, \sum_{p=1}^{P} \pi_{pd}^{(2)} = 1$. Instead of sampling it from a fix Dirichlet prior with parameter in LDA that models them independently, we construct the probabilities by a stick-breaking process as follows. The stick-breaking notion has been widely used for constructing random weights [24], [47].

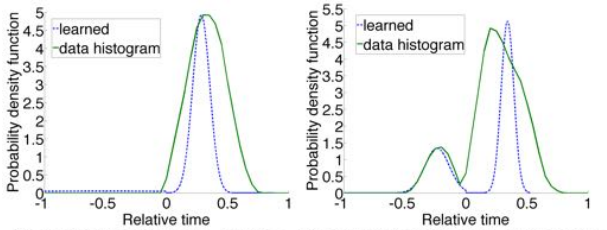
$$\pi_{kd}^{(1)} = \Psi(v_{kd}^{(1)}) \prod_{l=1}^{k-1} \Psi(-v_{ld}^{(1)}),\ \Psi(v_{kd}^{(1)}) = \frac{1}{1+\exp(-v_{kd}^{(1)})},$$

$$\pi_{pd}^{(2)} = \Psi(v_{pd}^{(2)}) \prod_{l=1}^{p-1} \Psi(-v_{ld}^{(2)}),\ \Psi(v_{pd}^{(2)}) = \frac{1}{1+\exp(-v_{pd}^{(2)})},$$

where $0 < \Psi(v_{kd}^{(1)}), \Psi(v_{pd}^{(2)}) < 1$ is a classic logistic function, which satisfies $\Psi(-v_{kd}^{(1)}) = 1 - \Psi(v_{kd}^{(1)}), \Psi(-v_{pd}^{(2)}) = 1 - \Psi(v_{pd}^{(2)})$, and $v_{kd}^{(1)}, v_{pd}^{(2)}$ serves as the prior of $\pi_{kd}^{(1)}, \pi_{pd}^{(2)}$.

TABLE 1: Notations in our model.

Symbols	Meaning
D	number of videos in the training database;
K	number of action-topics;
P	number of object-topics;
N_d	number of human-words/object-words in a video;
c_{nd}	n-th clip in d-th video;
w_{nd}^h	n-th human-word in d-th video;
w_{nd}^o	n-th object-word in d-th video;
$z_{nd}^{(1)}$	action-topic assignment of c_{nd};
$z_{nd}^{(2)}$	object-topic assignment of w_{nd}^o;
t_{nd}	normalized timestamp of of c_{nd};
t_{mnd}	$= t_{md} - t_{nd}$ the relative time between c_{md} and c_{nd};
$\pi_{:d}^{(1)}, \pi_{:d}^{(2)}$	the probabilities of action/object-topics in d-th document;
$v_{:d}^{(1)}, v_{:d}^{(2)}$	the priors of $\pi_{:d}^{(1)}, \pi_{:d}^{(2)}$ in d-th document;
$\phi_k^{(1)}$	multinomial human-word distribution from action-topic k;
$\phi_{kp}^{(12)}$	multinomial object-word distribution from action-topic k and object-topic p;
μ, Σ	multivariate normal distribution of $v_{:d} = [v_{:d}^{(1)}, v_{:d}^{(2)}]$;
θ_{kl}	relative time distribution of t_{mnd}, between action-topic k, l;



(a) *put-back-to-fridge* w.r.t. *pouring* (b) *put-back-to-fridge* w.r.t. *microwaving*

Fig. 5: The relative time distributions learned by our model on training set (the blue dashed line) and the ground-truth histogram of the relative time over the whole dataset (the green solid line).

In order to capture the correlations between action-topics and object-topics, we draw the packed vector $v_{:d} = [v_{:d}^{(1)}, v_{:d}^{(2)}]$ in the stick-breaking notion from a mutivariate normal distribution $N(\mu, \Sigma)$. In practice, we use a truncated vector $v_{:d}^{(1)} = [v_{1d}^{(1)}, \cdots, v_{K-1,d}^{(1)}]$ for (K-1) topics, and set $\pi_{Kd}^{(1)} = 1 - \sum_{k=1}^{K-1} \pi_{kd}^{(1)} = \prod_{k=1}^{K-1} \Psi(-v_{kd}^{(1)})$ as the probability of the final topic for a valid distribution. The same for $v_{:d}^{(2)}$.

Relative time distributions. The temporal relations between actions are also useful to discriminating the actions using temporal ordering and inferring the forgotten actions using the temporal context. We model the relative time of occurring actions by taking their time stamps into account. We consider that the relative time between two words are drawn from a certain distribution according to their topic assignments. In detail, let $t_{nd}, t_{md} \in (0,1)$ be the absolute time stamp of n-th word and m-th word, which is normalized by the video length. $t_{mnd} = t_{md} - t_{nd}$ is the relative time of m-th clip relative to n-th clip. Then t_{mnd} is drawn from a certain distribution, $t_{mnd} \sim \Omega(\theta_{z_{md}^{(1)}, z_{nd}^{(1)}})$, where $\theta_{z_{md}^{(1)}, z_{nd}^{(1)}}$ are the parameters. $\Omega(\theta_{k,l})$ are K^2 pairwise action-topic specific relative time distributions defined as follows:

$$\Omega(t|\theta_{k,l}) = \begin{cases} b_{k,l} \cdot N(t|\mu_{k,l}^+, \Sigma_{k,l}^+) & \text{if } t \ge 0, \\ 1 - b_{k,l} \cdot N(t|\mu_{k,l}^-, \Sigma_{k,l}^-) & \text{if } t < 0, \end{cases} \quad (1)$$

An illustration of the learned relative time distributions are shown in Fig. 5. We can see that the distributions we learned

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

correctly reflect the order of the actions, *e.g.*, *put-back-to-fridge* is after *pour* and can be before/after *microwave*, and the shape is almost similar to the real distributions. Here the Bernoulli distribution $b_{k,l}/1-b_{k,l}$ gives the probability of action k after/before the action l. And two independent normal distributions $N(t|\theta^+_{k,l})/N(t|\theta^-_{k,l})$ estimate how long the action k is after/before the action l[3]. Then the order and the length of the actions will be captured by all these pairwise relative time distributions.

5.1 Learning and Inference

Gibbs sampling is commonly used as a means of statistical inference to approximate the distributions of variables when direct sampling is difficult [7], [24]. Given a video, the word w^h_{nd}, w^o_{nd} and the relative time t_{mnd} are observed. We can integrate out $\Phi^{(1)}_k, \Phi^{(12)}_{kp}$ since $Dir(\beta^{(1)}), Dir(\beta^{(12)})$ are conjugate priors for the multinomial distributions $\Phi^{(1)}_k, \Phi^{(12)}_{kp}$. We also estimate the standard distributions including the mutivariate normal distribution $N(\mu,\Sigma)$ and the time distribution $\Omega(\theta_{kl})$ using the method of moments, once per iteration of Gibbs sampling. As in many applications using the topic model, we use the fixed symmetric Dirichlet distributions by setting $\beta^{(1)}, \beta^{(12)}$ as 0.01.

Then we introduce how we sample the topic assignment $z^{(1)}_{nd}, z^{(2)}_{nd}$. We do a collapsed sampling as in LDA by calculating the posterior distribution of $z^{(1)}_{nd}, z^{(2)}_{nd}$:

$$p(z^{(1)}_{nd}=k|\pi^{(1)}_{:d}, z^{(1)}_{-nd}, z^{(2)}_{nd}, t_{nd})$$
$$\propto \overbrace{\pi^{(1)}_{kd}}^{\text{topic prior}} \overbrace{\omega(k,w^h_{nd})\omega(k,z^{(2)}_{nd},w^o_{nd})}^{\text{topic-word distribution}} \overbrace{p(t_{nd}|z^{(1)}_{:d},\theta)}^{\text{relative time distribution}},$$
$$p(z^{(2)}_{nd}=p|\pi^{(2)}_{:d}, z^{(2)}_{-nd}, z^{(1)}_{nd})$$
$$\propto \overbrace{\pi^{(2)}_{pd}}^{\text{topic prior}} \overbrace{\omega(z^{(1)}_{nd},p,w^o_{nd})}^{\text{topic-word distribution}},$$
$$\omega(k,w^h_{nd}) = \frac{N^{-nd}_{kw^h}+\beta^{(1)}}{N^{-nd}_{k}+N_{w^h}\beta^{(1)}},$$
$$\omega(k,p,w^o_{nd}) = \frac{N^{-nd}_{kpw^o}+\beta^{(12)}}{N^{-nd}_{kp}+N_{w^o}\beta^{(12)}},$$
$$p(t_{nd}|z^{(1)}_{:d},\theta) = \prod_m^{N_d} \Omega(t_{mnd}|\theta_{z^{(1)}_{md},k})\Omega(t_{nmd}|\theta_{k,z^{(1)}_{md}}), \quad (2)$$

where N_{w^h}, N_{w^o} is the number of unique word types in dictionary, $N^{-nd}_{kw^h}/N^{-nd}_{kpw^o}$ denotes the number of instances of word w^h_{nd}/w^o_{nd} assigned with action-topic k/action-topic k and object-topic p, excluding n-th word in d-th document, and N^{-nd}_k/N^{-nd}_{kp} denotes the number of total words assigned with action-topic k/action-topic k and object-topic p. $z^{(1)}_{-nd}/z^{(2)}_{-nd}$ denotes the topic assignments for all words except $z^{(1)}_{nd}/z^{(2)}_{nd}$. The detailed derivation of Eq. (2) is in the Appendix A.

Intuitions. In Eq. (2), note that the topic assignments are decided by topic priors, word-topic distributions and relative time distributions. The topic priors are sampled by the topic co-occurrence distributions, that reflect which actions/objects are more likely to co-occur in a video. The word-topic distributions reflect the visual appearance of the video clips in different actions. The relative time distributions reflect the ordering and the time gaps between actions.

Due to the logistic stick-breaking transformation, the posterior distribution of the topic priors $v_{:d} = [v^{(1)}_{:d}, v^{(2)}_{:d}]$ does not have a closed form. So we instead use a Metropolis-Hastings independence sampler [16]. Let the proposals $q(v^*_{:d}|v_{:d},\mu,\Sigma) = N(v^*_{:d}|\mu,\Sigma)$ be drawn from the prior. The proposal is accepted with probability $\min(\mathbb{A}(v^*_{:d}, v_{:d}), 1)$, where

$$\begin{aligned}
&\mathbb{A}(v^*_{:d}, v_{:d}) \\
&= \frac{p(v^*_{:d}|\mu,\Sigma)\prod_{n=1}^{N_d} p(z^{(1)}_{nd}|v^{(1)*}_{:d})p(z^{(2)}_{nd}|v^{(2)*}_{:d})q(v_{:d}|v^*_{:d},\mu,\Sigma)}{p(v_{:d}|\mu,\Sigma)\prod_{n=1}^{N_d} p(z^{(1)}_{nd}|v^{(1)}_{:d})p(z^{(2)}_{nd}|v^{(2)}_{:d})q(v^*_{:d}|v_{:d},\mu,\Sigma)} \\
&= \prod_{n=1}^{N_d} \frac{p(z^{(1)}_{nd}|v^{(1)*}_{:d})p(z^{(2)}_{nd}|v^{(2)*}_{:d})}{p(z^{(1)}_{nd}|v^{(1)}_{:d})p(z^{(2)}_{nd}|v^{(2)}_{:d})} \\
&= \prod_{k=1}^{K} (\frac{\pi^{(1)*}_{kd}}{\pi^{(1)}_{kd}})^{\sum_{n=1}^{N_d}\delta(z^{(1)}_{nd},k)} \prod_{p=1}^{P} (\frac{\pi^{(2)*}_{pd}}{\pi^{(2)}_{pd}})^{\sum_{n=1}^{N_d}\delta(z^{(2)}_{nd},p)},
\end{aligned}$$

which can be easily calculated by counting the number of words assigned with each topic by $z^{(1)}_{nd}, z^{(2)}_{nd}$. Here the function $\delta(x,y) = 1$ if only if $x=y$, otherwise equal to 0.

For inference of a test video, we sample the unknown topic assignments $z^{(1)}_{nd}, z^{(2)}_{nd}$ and the topic priors $v^{(1)}_{:d}, v^{(2)}_{:d}$ using the learned parameters in the training stage.

5.2 Fast sampler

Using Eq.2, the time complexity of the sampling per iteration would be $O(N^2_d DK)$, which is because $p(t_{nd}|z^{(1)}_{:d},\theta)$ needs to multiply N_d terms for each word and each topic in each document. In this sub-section, we present a fast computation of $p(t_{nd}|z^{(1)}_{:d},\theta)$ using the recursive formula, which makes its computation cost constant time, so that the total sampling time complexity reduces to $O(N_d DK)$.

First, we simplify the notation of $p(t_{nd}|z^{(1)}_{:d},\theta)$ by merging the exponential terms:

$$\begin{aligned}
p(t_n) = p(t_{nd}|z^{(1)}_{:d},\theta) &= \prod_{m\neq n} \gamma_m \cdot e^{\sum_{m\neq n}\lambda_m(t_{mn}-\mu_m)^2} \\
&= \Gamma(n)e^{\Theta(n)},
\end{aligned}$$

$where:$

$$\Gamma(n) = \prod_{m\neq n}\gamma_m, \quad \Theta(n) = \sum_{m\neq n}\lambda_m(t_{mn}-\mu_m)^2.$$

where λ_m, μ_m denotes the merged parameters of the time distributions in Eq.(1) according to action-word m, n's action-topics. If we can derive a recursive formula of $\Gamma(n), \Theta(n)$, we can fast compute $p(t_n)$ using $p(t_{n-1})$ in a constant time. The recursive formula of $\Gamma(n)$ is straightforward:

$$\Gamma(n) = \frac{\gamma_{n-1}}{\gamma_n}\Gamma(n-1).$$

For $\Theta(n)$, we have:

$$\begin{aligned}
&\Theta(n) - \Theta(n-1) \\
=&\lambda_{n-1}(t_{n-1,n}-\mu_{n-1})^2 - \lambda_n(t_{n,n-1}-\mu_n)^2 \\
&+ \sum_{m\neq n,n-1}\lambda_m((t_{mn}-\mu_m)^2 - (t_{m,n-1}-\mu_m)^2),
\end{aligned}$$

3. Specially, when $k=l$, If two words are in the same segments, we draw t from a normal distribution which is centered on zero, and the variance models the length of the action. If not, it also follows Eq. (1) indicating the relative time between two same actions. We also use functions $\tan(-\pi/2+\pi t)(0<t<1), \tan(\pi/2+\pi t)(-1<t<0)$ to feed t to the normal distribution so that the probability is valid, that sums to one through the domain of t.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 6: Examples of composite activity videos. Our model is able to consider complex action relations using pairwise action co-occurrence and temporal modeling.

where:

$$\begin{aligned}&\sum_{m\neq n,n-1}\lambda_m((t_{mn}-\mu_m)^2-(t_{m,n-1}-\mu_m)^2)\\=&-2\Delta t\sum_{m\neq n,n-1}\lambda_m t_{m,n}-\Delta t^2\sum_{m\neq n,n-1}\lambda_m\\&+2\Delta t\sum_{m\neq n,n-1}\lambda_m\mu_m.\end{aligned}$$

Since $t_{m,n-1}-t_{m,n}=\Delta t$ is the time gap between neighbouring action-words, which is fixed in our model. It is not hard to derive the recursive formulas of the three terms in the above equation:

$$\begin{aligned}\lambda(n)&=\sum_{m\neq n,n-1}\lambda_m=\sum_{m\neq n-1,n-2}\lambda_m+\lambda_{n-2}-\lambda_n\\&=\lambda(n-1)+\lambda_{n-2}-\lambda_n,\\\lambda\mu(n)&=\sum_{m\neq n,n-1}\lambda_m\mu_m\\&=\sum_{m\neq n-1,n-2}\lambda_m\mu_m+\lambda_{n-2}\mu_{n-2}-\lambda_n\mu_n\\&=\lambda\mu(n-1)+\lambda_{n-2}\mu_{n-2}-\lambda_n\mu_n,\\\lambda t(n)&=\sum_{m\neq n,n-1}\lambda_m t_{m,n}=\sum_{m\neq n,n-1}\lambda_m(t_{m,n-1}-\Delta t)\\&=\sum_{m\neq n,n-1}\lambda_m t_{m,n-1}-\sum_{m\neq n,n-1}\lambda_m\Delta t\\&=\sum_{m\neq n-1,n-2}\lambda_m t_{m,n-1}+\lambda_{n-2}t_{n-2,n-1}-\lambda_n t_{n,n-1}\\&\quad-\lambda(n)\\&=\lambda t(n-1)+\lambda_{n-2}t_{n-2,n-1}-\lambda_n t_{n,n-1}-\lambda(n).\end{aligned}$$

According to above equations, we can recursively calculate $\Theta(n)$ in constant time by recursively updating $\lambda(n),\lambda\mu(n),\lambda t(n)$.

6 Discussions

Note that the novelty of our approach is the ability to model the long-range action relations in the temporal sequence, by considering pairwise action co-occurrence and temporal relations, for example in Fig. 6, put-milk-back-to-fridge often co-occurs with and temporally after (but not necessarily follows) fetch-milk-from-fridge. In our topic model, these two action topics would have strong correlations in topic hyper prior Σ and a peak in negative axle of the relative time distribution θ.

In previous works on action modeling, temporal relations are often considered between neighbouring frames [5], [14], [17], [48], [52], [54] or in hierarchical structures [29], [31], [41], [42], [56], [58], [60]. However, some obvious long-term relations are missing in the linear neighbouring modeling and actions do not necessarily follow a hierarchy in a video *e.g.*, there is no hierarchy in the examples in Fig. 6, while our pairwise topic co-occurrence distributions and relative time distributions model is more general to capture these relations globally and completely. Moreover, we developed a faster sampler using recursive formulas, which keeps the computation linear to number of words. In our experiments, it only took 30 sec. to sample one round on iMac 2.9GHz Core I5, where the dataset has 11 action topics and 129 videos with 148 words in average in each video.

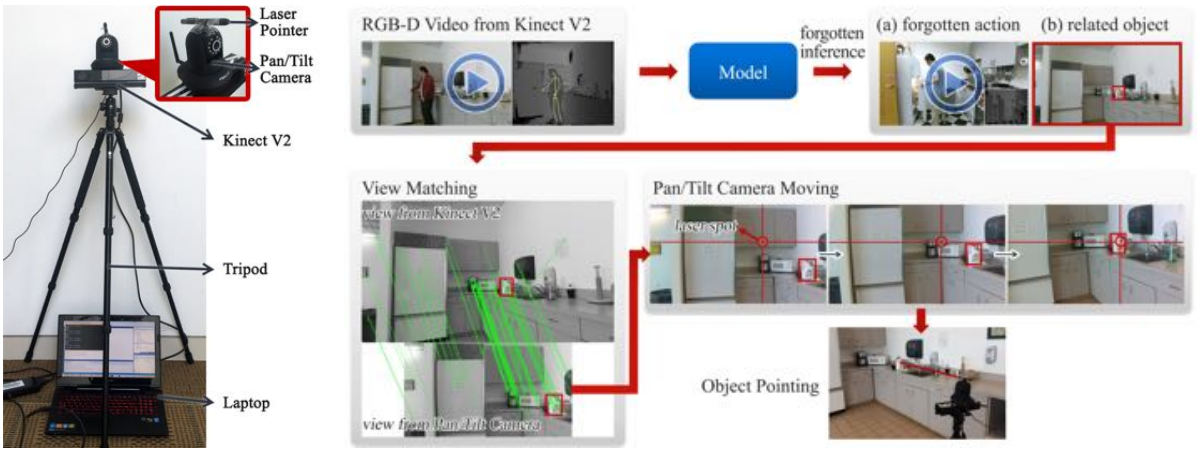
7 Watch-Bot to Reminding of Forgotten Actions

The average adult forgets three key facts, chores or events every day [2]. So it is important for a personal robot to be able to detect not only what a human is currently doing but also what he forgot to do. In this section, we describe a new robot system (see Fig. 7) to detect the forgotten actions and remind people, which we called *action patching*, using our learning model.

Note that detecting forgotten action is more challenging than conventional action recognition, since what to infer is not shown in the query video. Also, our model does not necessarily know the semantic class of the actions. Instead it learns action clusters and relations from the unlabeled action videos and use them to detect forgotten actions and remind people. Therefore, modeling rich relations from videos is important to providing evidence for detecting forgotten actions. Our model models pairwise co-occurrence and long-range temporal relations of actions/topics. As a result, rather than only modeling the single action or the local temporal transitions in the previous works, those actions occurred with a relatively large time interval, occurred after the forgotten actions, as well as the interacting objects can also be used to detect forgotten actions in our model. For example, a *put-back-book* might be forgotten as previously seen a *fetch-book* action before a long *read* action, and seen a *book* and a *leave* action indicates he really forgot it.

We enable a robot, that we call Watch-Bot, to detect humans' forgotten actions as well as localize the related object in the current scene. The robot consists of a Kinect v2 sensor, a pan/tilt camera (which we call camera for brevity in this paper) mounted with a laser pointer, and a laptop (see Fig. 7). This setup can be easily deployed on any assistive robot. Taking the example in Fig. 1, if our robot sees a person fetch a milk from the fridge, pour the milk, and leave without putting the milk back to the fridge. Our robot would first detect the forgotten action and the related object (the milk), given the input RGB-D frames and human skeletons from the Kinect; then map the object from the Kinect's view to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Robot System. (b) System Pipeline.

Fig. 7: (a). Our Watch-Bot system. It consists of a Kinect v2 sensor that inputs RGB-D frames of human actions, a laptop that infers the forgotten action and the related object, a pan/tilt camera that localizes the object, mounted with a fixed laser pointer that points out the object. (b). The system pipeline. The robot first uses the learned model to infer the forgotten action and the related object based on the Kinect's input. Then it maps the view from the Kinect to the pan/tilt camera so that the bounding box of the object is mapped in the camera's view. Finally, the camera moves until the laser spot lies in the bounding box of the target object.

the camera's view; finally pan/tilt the camera till its mounted laser pointer pointing to the milk.

Our goal is to detect the forgotten action and then point out the related object in the forgotten action using our learned model (see Alg. 1). We first use our model to segment the query video into action segments (step 1,2 in Alg. 1), and then infer the most possible forgotten action-topic and the related object-topic (step 4 in Alg. 1). Next we retrieve a top forgotten action segment from the training database, containing the inferred forgotten action-topic and the object-topic (step 5,6 in Alg. 1). Using the extracted object in the retrieved segment, we detect the bounding box of the related forgotten object in the Kinect's view of the query video (step 8,9,10 in Alg. 1). After that, we map the bounding box of the object from the Kinect's view to the camera's view. Finally, the pan/tilt camera moves until its mounted laser pointer points out the related object in the current scene.

Patched Action and Object Inference. Our model infers the forgotten action using the probability inference based on the dependencies between actions and objects. After assigning the action-topics and object-topics to a query video q, we consider adding one additional clip $\hat{c}$ consisting of $\hat{w^h}, \hat{w^o}$ into q in each action segmentation point t_s (see Fig 8). Then the probabilities of the missing action-topics k_m with object-topics p_m in each segmentation point t_s can be compared following the posterior distribution in Eq. (2):

$$\begin{aligned} &p(z^{(1)}_{\hat{c}} = k_m, z^{(2)}_{\hat{c}} = p_m, t_{\hat{c}} = t_s | other) \\ &\propto \pi^{(1)}_{k_m d} \pi^{(2)}_{p_m d} p(t_s | z^{(1)}_{:d}, \theta) \sum_{w^h, w^o} \omega(k_m, w^h) \omega(k_m, p_m, w^o), \\ &s.t. \quad t_s \in T_s, \; k_m \in [1:K] - K_e, \end{aligned} \tag{3}$$

where T_s is the set of segmentation points (t_1, t_2 in Fig. 8) and K_e is the set of existing action-topics in the video (*fetch-book*, *etc*. in Fig. 8). Thus $[1:K] - K_e$ are the missing topics in the video (*put-down-items*, *etc*. in Fig. 8). $p(t_s | z^{(1)}_{:d}, \theta), \omega(k_m, w^h), \omega(k_m, p_m, w^o)$ can be computed as in Eq. (2). Here we marginized $\hat{w^h}, \hat{w^o}$ to avoid the effect of a specific human-word or object-word. Note that, $\pi^{(1)}_{kd}, \pi^{(2)}_{pd}$ gives the probability of a missing action-topic with an object-topic in the video decided by the correlation we learned in the joint distribution prior, *i.e*., the close topics have higher probabilities to occur in this query video. And $p(t_s | z^{(1)}_{:d}, \theta)$ measures the temporal consistency of adding a new action-topic. And the marginized word-topic distribution $\sum_{w^h, w^o} \omega(k_m, w^h) \omega(k_m, p_m, w^o)$ give the likelihood of the topic learned from training data.

Algorithm 1 Forgotten Action and Object Detection.

Input: RGB-D video q with tracked human skeletons.
Output: Claim no action forgotten, or output an action segment with the forgotten action and a bounding box of the related object in the current scene.
1. Assign the action-topics to clips and the object-topics to object-words in q as introduced in Section 5.1.
2. Get the action segments by merging the continuous clips with the same assigned action-topic.
3. If the assigned action-topics K_e in q contains all modeled action-topics $[1:K]$, claim no action forgotten and return;
4. For each action segmentation point t_s, not assigned action-topic $k_m \in [1:K] - K_e$, and object-topic $p_m \in [1:P]$:
   Compute the probability defined in Eq. 3;
5. Select the top tree possible tuples (k_m, p_m, t_s), and get the forgotten action segment candidate set Q which contains segments with topics (k_m, p_m);
6. Select the top forgotten action segment p from Q with the maximum $patch\_score(p)$;
7. If $patch\_score(p)$ is smaller than a threshold, claim no action forgotten and return;
8. Segment the current frame to super-pixels using edge detection [12] as in Section 3;
9. Select the nearest super-pixels to both extracted object bounding box in q and p.
10. Merge the adjacent super-pixels and bound the largest one with a rectangle as the output bounding box.
11. Return the top forgotten action segment and the object bounding box.

Patched Action and Object Detection. Then we select the top three tuples (k_m, p_m, t_s) using the above probability. The action segments of action-topic k_m containing object-topic p_m in the training set consist a patched action candidate segment set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

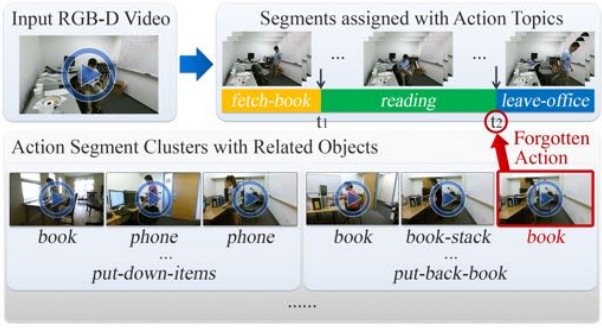


Fig. 8: **Illustration of patched action and object inference using our model.** Given a test video, we infer the forgotten action-topic and object-topic in each segmentation point (t_1, t_2 as above). Then we select the top segment from the inferred action-topic's segment cluster with the inferred object-topic with the maximum $patch\_score$.

Q. We then select the patched action segment from Q with the maximum $patch\_score(p) = ave(\mathbb{D}(f_{pm}, f_{qf}), \mathbb{D}(f_{pm}, f_{qt})) - max(\mathbb{D}(f_{pf}, f_{qt}), \mathbb{D}(f_{pt}, f_{qf}))$, where $\mathbb{D}(,)$ is the average pair-wise distances between frames, $ave(,), max(,)$ are the average and max value. In detail, we consider that the front and the tail of the patched action segment f_{pf}, f_{pt} should be similar to the tail of the adjacent segment in q before t_s and the front of the adjacent segment in q after t_s: f_{qt}, f_{qf}. At the same time, the middle of the patched action segment f_{pm} should be different to f_{qt}, f_{qf}, as it is a different action forgotten in the video.[4] If the maximum score is below a threshold or there is no missing topics ($i.e., K_e = [1 : K]$) in the query video, we claim there is no forgotten actions. Then we detect the bounding box of the patched object. We first segment the current frame into super-pixels as in Section 3, second search the nearest segments using the extracted object in the test video and the patched action, finally merge the adjacent segments into one segment and bound the largest segment with a bounding box.

Real Object Pointing. We now describe how we pan/tilt the camera to point out the real object in the current scene. We first compute the transformation homography matrix between the frame of the Kinect and the frame of the pan/tilt camera using keypoints matching and RANSAC, which can be done very fast within 0.1 second. Then we can transform the detected bounding box from the Kinect's view to the pan/tilt camera's view. Since the position of the laser spot in the pan/tilt camera view is fixed, next we only need to pan/tilt the camera till the laser spot lies within the bounding box of the target object. To avoid the coordinating error caused by distortion and inconsistency of the camera movement, we use an iterative search plus small step movement instead of one step movement to localize the object (illustrated in Fig. 7). In each iteration, the camera pan/tilt a small step towards to the target object according to the relative position between the laser spot and the bounding box. Then the homography matrix is recomputed in the new camera view, so that the bounding box is mapped in the new view. Until the laser spot is close enough to the center of the bounding box, the camera stops moving.

4. Here the middle, front, tail frames are 20%-length of segment centering on the middle frame, starting from the first frame, and ending at the last frame in the segment respectively.

8 Experiments

8.1 Watch-n-Patch Dataset

We collect a new challenging RGB-D activity dataset recorded by the new Kinect v2 camera. Each video in the dataset contains 2-7 actions interacting with different objects (see examples in Fig. 12). The new Kinect v2 has high resolution of RGB-D frames (RGB: 1920×1080, depth: 512×424) and improved body tracking of human skeletons (25 body joints). We record 458 videos with a total length of about 230 minutes and ask 7 subjects to perform human daily activities in 8 offices and 5 kitchens with complex backgrounds. In each environment, the activities are recorded in different views. It composed of fully annotated 21 types of actions (10 in the office, 11 in the kitchen) interacting with 23 types of objects.

In order to get a variation in activities, we ask participants to finish task with different combinations of actions and ordering. Some actions occur together often such as *fetch-from-fridge* and *put-back-to-fridge* while some are not always in the same video such as *take-item* and *read*. Some actions are in fix ordering while some occur in random order. To evaluate the action patching performance, 222 videos in the dataset has action forgotten and the forgotten actions are annotated. We give the examples of action classes in Fig. 12.

8.2 Experimental Setting and Compared Baselines

We evaluate in two environments 'office' and 'kitchen'. In each environment, we split the data into a train set with most full videos (office: 87, kitchen 119) and a few forgotten videos (office: 10, kitchen 10), and a test set with a few full videos (office: 10, kitchen 20) and most forgotten videos (office: 89, kitchen 113). In our experiments, we compare seven unsupervised approaches using two types of visual features and different correlations and topics modeling. We denote the approach settings as (method name)-(topic)-(temporal modeling)-(visual features). We compare seven approaches with action-topic only (A) and using our human skeleton and RGB-D features (SRGBD) introduced in Section 4: Hidden Markov Model (HMM-A-LT-SRGBD), topic model LDA (TM-A-NT-SRGBD), correlated topic model (CTM-A-NT-SRGBD), topic model over absolute time (TM-A-AT-SRGBD), correlated topic model over absolute time (CTM-A-AT-SRGBD), topic model over relative time (TM-A-RT-SRGBD) and our causal topic model with only action-topics (CTM-A-RT-SRGBD) [62], where LT,NT,AT,RT means linear temporal modeling, no temporal modeling, absolute time modeling and relative time modeling. We compare three methods with both action-topics and object-topics (AO): HMM-AO-LT-SRGBD, LDA-AO-NT-SRGBD and our CTM-AO-NT-SRGBD [63]. We also evaluate HMM and our model using the popular features for action recognition, dense trajectories feature (DTF) [57], extracted in RGB videos[5], named as HMM-A-LT-DTF and CTM-A-RT-DTF, CTM-AO-RT-DTF.

In the experiments, we set the number of topics and states of HMM equal to or more than ground-truth classes. For correlated topic models, we use the same topic prior in our model. For models over absolute time, we consider the absolute time of each word is drawn from a topic-specific normal distribution. For models over relative time, we use the same relative time distribution as in our model (Eq. (1)). The clip length of the action-words is

5. We train a codebook with the size of 2000 and encode the extracted DTF features in each clip as the bag of features using the codebook.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2: Results using the same number of topics as the ground-truth action classes. (top one is bold)

'office' (%)	Seg-Acc train	Seg-Acc test	Seg-AP train	Seg-AP test	Frame-Acc train	Frame-Acc test	PA-Acc	PO-Acc
HMM-A-LT-DTF	15.2	9.4	21.4	20.7	20.2	15.9	23.6	-
HMM-A-LT-SRGB	18.0	14.0	25.9	24.8	24.7	21.3	33.3	-
HMM-AO-LT-SRGB	18.2	19.4	26.2	23.1	25.3	27.3	32.2	20.4
TM-A-NT-SRGB	9.3	9.2	20.9	19.6	20.3	13.0	13.3	-
TM-AO-NT-SRGB	9.8	12.2	22.3	19.6	24.6	18.4	15.7	10.5
CTM-A-NT-SRGB	10.0	5.9	18.1	15.8	20.2	16.4	13.3	-
TM-A-AT-SRGB	8.9	3.7	25.4	19.0	18.6	13.8	12.0	-
CTM-A-AT-SRGB	9.6	6.8	25.3	19.8	19.6	15.5	10.8	-
TM-A-RT-SRGB	30.8	30.9	29.0	30.2	38.1	36.4	39.5	-
CTM-A-RT-DTF	28.2	27.0	28.3	27.4	37.4	34.0	33.7	-
CTM-AO-RT-DTF	28.5	29.1	30.6	29.5	37.9	35.0	36.2	30.5
CTM-A-RT-SRGB	30.6	32.9	**33.1**	34.6	39.9	38.5	41.5	-
CTM-AO-RT-SRGB	**33.2**	**35.2**	33.0	**36.0**	**40.1**	**41.2**	**46.2**	**36.4**
'kitchen' (%)	Seg-Acc train	Seg-Acc test	Seg-AP train	Seg-AP test	Frame-Acc train	Frame-Acc test	PA-Acc	PO-Acc
HMM-A-LT-DTF	4.9	3.6	18.8	5.6	12.3	9.8	2.3	-
HMM-A-LT-SRGB	20.3	15.2	20.7	13.8	21.0	18.3	7.4	-
HMM-AO-LT-SRGB	23.9	17.2	21.1	18.8	23.5	20.3	12.4	5.3
TM-A-NT-SRGB	7.9	4.7	21.5	14.7	20.9	11.5	9.6	-
TM-AO-NT-SRGB	7.9	6.7	22.6	17.1	24.9	14.4	10.8	5.3
CTM-A-NT-SRGB	10.5	9.2	20.5	14.9	18.9	15.7	6.4	-
TM-A-AT-SRGB	8.0	4.8	21.5	21.6	20.9	14.0	7.4	-
CTM-A-AT-SRGB	9.7	10.0	19.1	22.6	20.1	16.7	10.7	-
TM-A-RT-SRGB	32.3	26.9	23.4	23.0	35.0	31.2	18.3	-
CTM-A-RT-DTF	26.9	23.6	18.4	17.4	33.3	29.9	16.5	-
CTM-AO-RT-DTF	27.2	25.3	19.1	18.6	32.9	30.2	17.6	13.2
CTM-A-RT-SRGB	**33.2**	29.0	26.4	25.5	37.5	34.0	20.5	-
CTM-AO-RT-SRGB	32.1	**30.7**	**28.5**	**28.5**	**39.2**	**36.9**	**24.4**	**20.6**

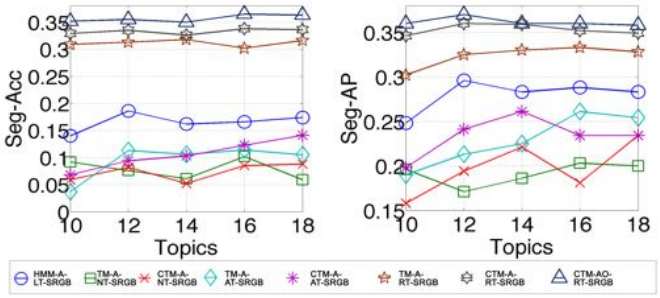


Fig. 9: Segmentation Acc/AP varied with the number of topics in the 'office' test dataset.

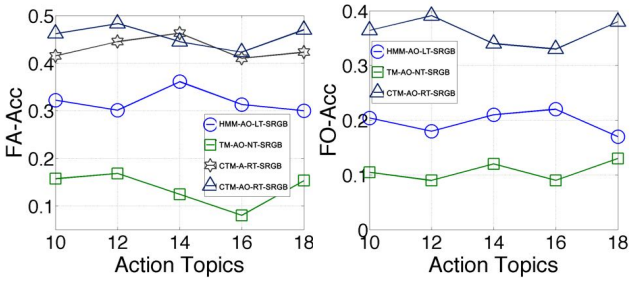


Fig. 10: Forgotten action/object detection accuracy varied with the number of action-topics in the 'office' dataset.

set to 20 frames, densely sampled by step one and the size of action dictionary is set to 500. For patching, the candidate set for different approaches consist of the segments with the inferred missing topics by transition probabilities for HMM, the topic priors for TM and CTM, and both the topic priors and the time distributions for TM-AT, TM-RT, CTM-AT and our CTM-RT.

8.3 Evaluation Metrics

Action Segmentation and Cluster Assignment. First we need to evaluate whether the unsupervised learned action-topics and states of HMM are semantically meaningful. As there are no semantics output from the unsupervised learning, typically the assigned topics are mapped to the ground-truth labels for final evaluation. We first count the mapped frames between topics and ground-truth classes and do the mapping as follows. Let k_i, c_i be the assigned topic and ground-truth class of frame i. The count of a mapping is: $m_{kc} = \frac{\sum_i \delta(k_i,k)\delta(c_i,c)}{\sum_i \delta(c_i,c)}$, where $\sum_i \delta(k_i,k)\delta(c_i,c)$ is the number of frames assigned with topic k as the ground-truth class c and normalized by the number of frames as the ground-truth class c: $\sum_i \delta(c_i,c)$. Then we solve the following binary linear programming to get the best mapping:

$$\max_x \sum_{k,c} x_{kc} m_{kc},$$

$$s.t. \quad \forall k, \sum_c x_{kc} = 1, \quad \forall c, \sum_k x_{kc} \geq 1, \quad x_{kc} \in \{0,1\},$$

where $x_{kc} = 1$ indicates mapping topic k to class c, otherwise $x_{kc} = 0$. And $\sum_c x_{kc} = 1$ constrain that each topic must be mapped to exact one class, $\sum_k x_{kc} \geq 1$ constrain that each class must be mapped by at least one topic.

Two settings are considered: Per frame: *frame-wise accuracy (Frame-Acc)*, the ratio of correctly labeled frames. Segmentation: the *segmentation accuracy (Seg-Acc)*, the ratio of the ground-truth segments that are correctly detected[6], and the *segmentation average precision (Seg-AP)* by sorting all output action segments using the average probability of their words' topic assignments. These three metrics are evaluated by taking the average of each action class.

Forgotten Action and Object Detection. We evaluate the *patching accuracy (PA-Acc)* by the portion of correct patched video, including correctly output the forgotten action segments or claiming no forgotten actions. We consider the output action segments by the algorithm containing over 50% ground-truth forgotten actions as correctly output the forgotten action segments. We also measure the *patching object detection accuracy (PO-Acc)* by the typical object detection metric, that considers a true positive if the overlap rate (union/intersection) between the detected and the ground-truth object bounding box is greater than 40%.

8.4 Results

Table 2 and Fig. 9 show the main results of our experiments. We first perform evaluation in the training set to see if actions can be well segmented and clustered in the train set. We then perform testing in the test set to see if the new video from the test set can be correctly segmented and the segments can be correctly assigned to the action cluster. We discuss our results in the light of the following questions.

Did modeling the long-range relations help? Modeling the correlations and the temporal relations between actions is the key concept in this paper. From the results, we can see that the approaches considering the temporal relations, HMM-LT, TM-RT, and our CTM-RT, achieve better performance than other approaches which assume actions are temporal independent. This shows that understanding temporal structure is critical to

6. a true positive if the overlap (union/intersection) between the detected and the ground-truth segments is more than a default threshold 40% as in [42].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 11: An example of the robotic experiment. The robot detects the human left the food in the microwave, then points to the microwave.

TABLE 3: Robotic experiment results. The higher the better.

	Succ-Rate(%)	Subj-AccScore(1-5)	Subj-HelpScore(1-5)
HMM-AO-LT-SRGB	37.5	2.1	2.3
TM-AO-NT-SRGB	29.2	1.8	2.0
CTM-AO-RT-SRGB	**62.5**	**3.5**	**3.9**

recognizing and patching actions. In detail, the approaches considering the topic correlations CTM, CTM-AT, and our CTM-RT outperforms the corresponding non-correlated topic models TM-NT, TM-AT, and TM-RT. The approaches modeling both the short-range and the long-range relations, TM-RT and CTM-RT, outperforms HMM-LT modeling only local relations. Our CTM-RT, which considers both the action correlation priors and the temporal relations, gives the best performance.

How successful was our unsupervised approach in learning meaningful action-topics? From Table 2, we can see that the unsupervised learned action-topics are promising to be semantically meaningful though ground-truth semantic labels are not provided in the training. In addition to the one-to-one correspondence between topics and semantic action classes, We also plot the performance curves varied with the topic number in Fig. 9. It shows that if we set the topics a bit more than ground-truth classes, the performance increases since a certain action might be divided into multiple action-topics. But as topics increase, more variations are also introduced so that performance saturates.

RGB videos vs. RGB-D videos. We also evaluate our model CTM-RT and HMM-LT using the popular RGB features for action recognition (CTM-A-RT-DTF, CTM-AO-RT-DTF and HMM-A-LT-DTF in Table 2) to see if our RGB-D object and human skeleton features help. Clearly, they outperform the DTF features as more accurate human motion and object are extracted.

How well did our new application of action patching performs? In Table 2, the approaches modeling more action relations mostly give better patching performance. This is due to the learned co-occurrence and temporal structure strongly help indicate which actions are forgotten. Our model capturing both the short-range and long-range action relations shows the best results.

How important is it to consider relations between actions and objects? It is clear to see that the model which did well in forgotten action detection also performed well in detecting forgotten object. Because our model CTM-AO-RT considers richer relations between the action and the object, it performs better in both forgotten action and forgotten object detection than those which models action and object independently as well as CTM-A-RT which only models the actions.

8.5 Robotic Experiments

In this section, we show how our Watch-Bot reminds people of the forgotten actions in the real-world scenarios. We test each two forgotten scenarios in 'office' and 'kitchen' respectively (*put-back-book*, *turn-off-monitor*, *put-milk-back-to-fridge* and *fetch-food-from-microwave*). We use a subset of the dataset to train the model in each activity type separately. In each scenario, we ask 3 subjects to perform the activity twice, in which the subject choose to forget the above four actions itself or not to forget any. Therefore, we test 24 trials in total. We evaluate three aspects. One is objective, the success rate (Succ-Rate): the laser spot lying within the object as correct. The other two are subjective, the average Subjective Accuracy Score (Subj-AccScore): we ask the participant if he thinks the pointed object is correct; and the average Subjective Helpfulness Score (Subj-HelpScore): we ask the participant if the output of the robot is helpful. Both of them are in $1-5$ scale, the higher the better.

Table 3 gives the results of our robotic experiments. We can see that our robot can achieve over 60% success rate and gives the best performance. In most cases people think our robot is able to help them understand what is forgotten. Fig. 11 gives an example of our experiment, in which our robot observed what a human is currently doing, realized he forgot to fetch food from microwave and then correctly pointed out the microwave in the scene.

9 Conclusion and Future Work

In this paper, we presented an algorithm that models the human activities in a completely unsupervised setting. We showed that it is important to modeling the long-range relations between the actions. To achieve this, we considered the video as a sequence of human-words/object-words, and an activity as a set of action-topics/object-topics. Then we modeled the word-topic distributions, the topic correlations and the topic relative time distributions. We then showed the effectiveness of our model in the unsupervised action segmentation and clustering, as well as the action patching. Moreover, we showed that our proposed robot system using the action patching algorithm was able to effectively remind people of forgotten actions in the real-world robotic experiments. For evaluation, we also contributed a new challenging RGB-D activity video dataset.

Though we showed the promising results and the interesting applications of the purely unsupervised models in the paper, we can see that the performance is not more than 50 percent on the large-scale variant data, as we have no knowledge of the semantic information. In the future, we plan to extend the model to the semi-supervised approaches that can effectively use a small portion of the annotated data for better learning, and improve on the performance in the real-world applications.

Appendix

We give the detailed derivation of the posterior distribution of z_{nd} (Eq. (2)) in this section. We begin with the joint distribution $p(\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)} | \pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta)$, where $\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \pi^{(1)}, \pi^{(2)}$ are all variables of the word

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

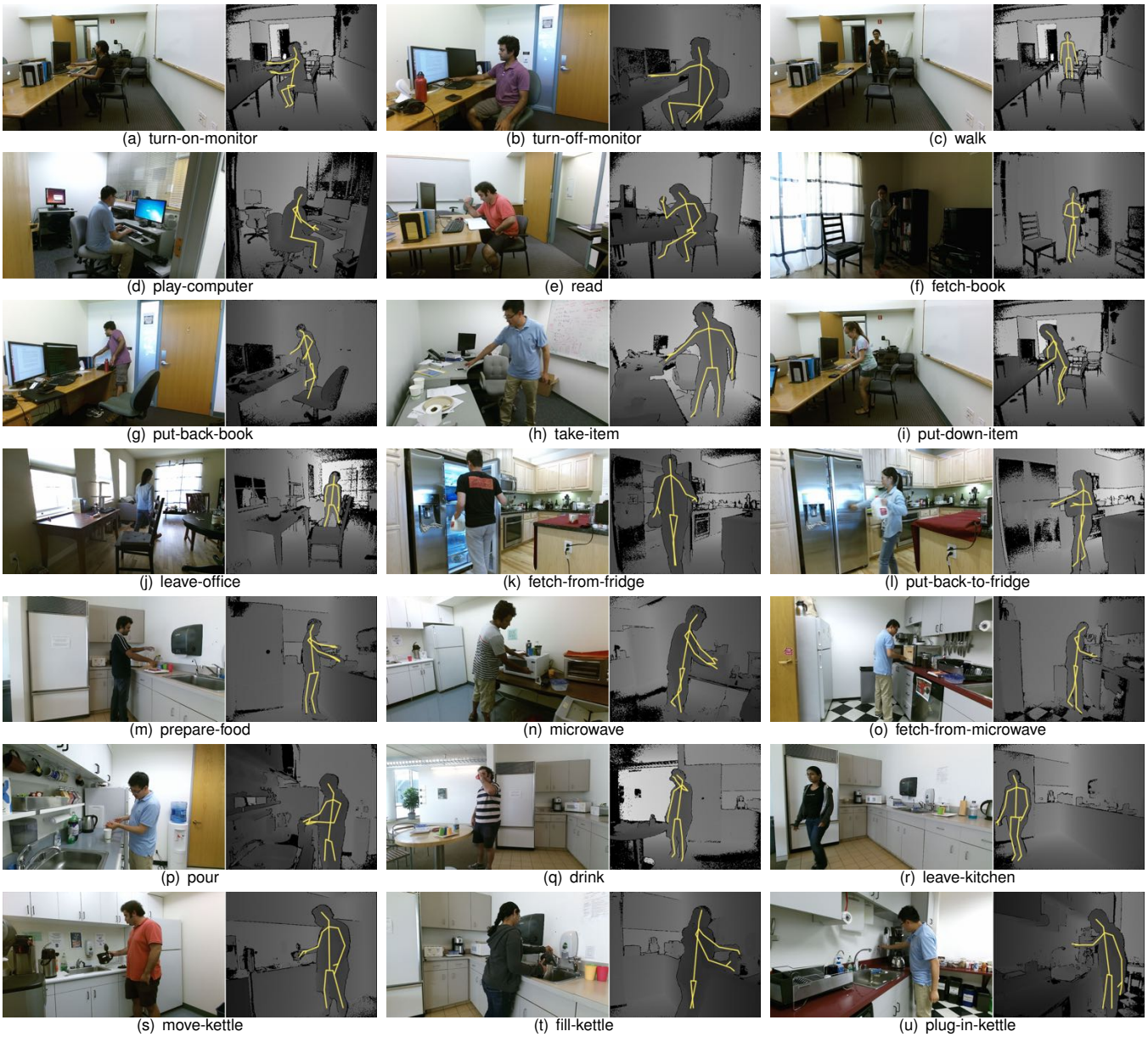


Fig. 12: Examples of every action class in our dataset. The left is RGB frame and the right is depth frame with human skeleton (yellow).

w^h_{nd},w^o_{nd}, the time stamp of a word t_{nd}, the topic-assignment of a word $z^{(1)}_{nd}$,$z^{(2)}_{nd}$ and the topic probability $\pi^{(1)}_{kd}$,$\pi^{(2)}_{kd}$ in D documents of K action topics and P object topics.

$$
\begin{aligned}
&p(\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)} | \pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta) \\
&= p(\mathbf{w}^h | \mathbf{z}^{(1)}, \beta^{(1)}) p(\mathbf{w}^o | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \beta^{(12)}) \cdot \\
&\quad p(\mathbf{t} | \mathbf{z}^{(1)}, \theta) p(\mathbf{z}^{(1)} | \pi^{(1)}) p(\mathbf{z}^{(2)} | \pi^{(2)}) \\
&= \int p(\mathbf{w}^h | \mathbf{z}^{(1)}, \phi^{(1)}) p(\phi^{(1)}, \beta^{(1)}) d\phi^{(1)} \cdot \\
&\quad \int p(\mathbf{w}^o | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \phi^{(12)}) p(\phi^{(12)}, \beta^{(12)}) d\phi^{(12)} \cdot \\
&\quad p(\mathbf{t} | \mathbf{z}^{(1)}, \theta) \cdot p(\mathbf{z}^{(1)} | \pi^{(1)}) p(\mathbf{z}^{(2)} | \pi^{(2)}).
\end{aligned}
$$

where the joint distribution is decided by the following five terms.

topic-word distributions:

$$
\begin{aligned}
&\int p(\mathbf{w}^h | \mathbf{z}^{(1)}, \phi^{(1)}) p(\phi^{(1)}, \beta^{(1)}) d\phi^{(1)} \\
&= \int \prod_{d=1}^{D} \prod_{n=1}^{N_d} \phi^{(1)}_{z^{(1)}_{nd}, w^h_{nd}} \prod_{k=1}^{K} \frac{1}{B(\beta^{(1)})} \prod_{w} \phi^{(1)\beta^{(1)}_w - 1}_{kw} d\phi^{(1)}_k \\
&= \prod_{k=1}^{K} \frac{B(N_k + \beta^{(1)})}{B(\beta^{(1)})} \\
&\quad \int p(\mathbf{w}^o | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \phi^{(12)}) p(\phi^{(12)}, \beta^{(12)}) d\phi^{(12)} \\
&= \prod_{k=1}^{K} \prod_{p=1}^{P} \frac{B(N_{kp} + \beta^{(12)})}{B(\beta^{(12)})},
\end{aligned}
$$

where we denote the Beta function as $B(\beta) = \frac{\prod_{k=1}^{K} \Gamma(\beta_k)}{\Gamma(\sum_{k=1}^{K} \beta_k)}$.

topic-pair relative time distribution:

$$
p(\mathbf{t} | \mathbf{z}^{(1)}, \theta) = \prod_{d=1}^{D} \prod_{m=1}^{N_d} \prod_{n=1}^{N_d} p(t_{mnd} | \theta_{z^{(1)}_{md}, z^{(1)}_{nd}}).
$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2679054, IEEE Transactions on Pattern Analysis and Machine Intelligence

topic priors:

$$p(\mathbf{z}^{(1)}|\pi^{(1)}) = \prod_{d=1}^{D}\prod_{n=1}^{N_d} \pi^{(1)}_{z^{(1)}_{nd},d}\; p(\mathbf{z}^{(2)}|\pi^{(2)}) = \prod_{d=1}^{D}\prod_{n=1}^{N_d} \pi^{(2)}_{z^{(2)}_{nd},d}.$$

Then for a certain assignment $z^{(1)}_{nd}$, we give the posterior using the above joint distribution:

$$\begin{aligned}
&p(z^{(1)}_{nd}|\pi^{(1)}_{:d}, z^{(1)}_{-nd}, z^{(2)}_{nd}, t_{nd}) \\
=&\frac{p(\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta)}{p(\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, z^{(1)}_{-nd}, \mathbf{z}^{(2)}|\pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta)} \\
\propto&\frac{p(\mathbf{w}^h, \mathbf{w}^o, \mathbf{t}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta)}{p(w^h_{-nd}, w^o_{-nd}, t_{-nd}, z^{(1)}_{-nd}, \mathbf{z}^{(2)}|\pi^{(1)}, \pi^{(2)}, \beta^{(1)}, \beta^{(12)}, \theta)} \\
=&\pi^{(1)}_{z^{(1)}_{nd},d}\omega(z^{(1)}_{nd}, w^h_{nd})\omega(z^{(1)}_{nd}, z^{(2)}_{nd}, w^o_{nd})p(t_{nd}|z^{(1)}_{:d}, \theta),
\end{aligned}$$

where:

$$\omega(z^{(1)}_{nd}, w^h_{nd}) = \prod_{k=1}^{K}\frac{B(N_k + \beta^{(1)})}{B(N^{-nd}_k + \beta^{(1)})} = \frac{N^{-nd}_{z^{(1)}_{nd},w^h} + \beta^{(1)}}{N^{-nd}_{z^{(1)}_{nd}} + N_{w^h}\beta^{(1)}}$$

$$\begin{aligned}
\omega(z^{(1)}_{nd}, z^{(2)}_{nd}, w^o_{nd}) &= \prod_{k=1}^{K}\prod_{p=1}^{P}\frac{B(N_{kp} + \beta^{(12)})}{B(N^{-nd}_{kp} + \beta^{(12)})} \\
&= \frac{N^{-nd}_{z^{(1)}_{nd},z^{(2)}_{nd},w^o} + \beta^{(12)}}{N^{-nd}_{z^{(1)}_{nd},z^{(2)}_{nd}} + N_{w^o}\beta^{(12)}}
\end{aligned}$$

$$p(t_{nd}|z^{(1)}_{:d}, \theta) = \prod_m \Omega(t_{mnd}|\theta_{z^{(1)}_{md},z^{(1)}_{nd}})\Omega(t_{nmd}|\theta_{z^{(1)}_{nd},z^{(1)}_{md}}).$$

Then assign $z^{(1)}_{nd}$ with a specific topic k, we have the sampling posterior $p(z^{(1)}_{nd} = k|\pi^{(1)}_{:d}, z^{(1)}_{-nd}, z^{(2)}_{nd}, t_{nd}))$ in Eq. (2). Similarly we can have $p(z^{(2)}_{nd} = p|\pi^{(2)}_{:d}, z^{(2)}_{-nd}, z^{(1)}_{nd})$.

REFERENCES

[1] Kinect v2 sensor. http://www.microsoft.com/en-us/kinectforwindows/develop/.

[2] Adults forget three things a day, research finds. http://www.telegraph.co.uk/news/uknews/5891701/Adults-forget-three-things-a-day-research-finds.html, 2009. The Daily Telegraph.

[3] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16:1–16:43, 2011.

[4] S. M. Assari, A. R. Zamir, and M. Shah. Video classification using semantic concept co-occurrences. In *CVPR*, 2014.

[5] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *CVPR*, June 2014.

[6] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[7] D. M. Blei and J. D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[9] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014.

[10] G. Chen, M. Giuliani, D. S. Clarke, A. K. Gaschler, and A. Knoll. Action recognition using ensemble weighted multi-instance learning. In *ICRA*, 2014.

[11] A. Chrungoo, S. Manimaran, and B. Ravindran. Activity recognition for natural human robot interaction. In *Social Robotics*, volume 8755, pages 84–94. 2014.

[12] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[13] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ECCV*, 2009.

[14] R. Emonet, J. Varadarajan, and J.-M. Odobez. Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *CVPR*, 2011.

[15] T. A. Faruquie, P. K. Kalra, and S. Banerjee. Time based activity inference using latent dirichlet allocation. In *BMVC*, 2009.

[16] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC press, 2013.

[17] M. Hoai, Z. zhong Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011.

[18] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, 2015.

[19] N. Hu, Z. Lou, G. Englebienne, and B. Krse. Learning to recognize human activities from soft labeled data. In *RSS*, 2014.

[20] A. Jain, A. Gupta, M. Rodriguez, and L. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013.

[21] S. Jones and L. Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *CVPR*, 2014.

[22] V. Kantorov and I. Laptev. Efficient feature extraction, encoding and classification for action recognition. In *CVPR*, 2014.

[23] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ECCV*, 2007.

[24] D. I. Kim and E. B. Sudderth. The doubly correlated nonparametric topic model. In *NIPS*, 2011.

[25] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *International Workshop on Sign, Gesture, and Activity in Conjunction with ECCV*, 2010.

[26] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *IJRR*, 32(8):951–970, 2013.

[27] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *RSS*, 2013.

[28] H. S. Koppula and A. Saxena. Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation. In *ICML*, 2013.

[29] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.

[30] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007.

[31] I. Lillo, A. Soto, and J. Carlos Niebles. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *CVPR*, 2014.

[32] Y.-Y. Lin, J.-H. Hua, N. C. Tang, M.-H. Chen, and H.-Y. Mark Liao. Depth and skeleton associated action recognition without online accessible rgb-d cameras. In *CVPR*, 2014.

[33] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011.

[34] M. Losch, S. Schmidt-Rohr, S. Knoop, S. Vacek, and R. Dillmann. Feature set selection and optimal classifier for human activity recognition. In *Robot and Human interactive Communication*, 2007.

[35] S. Ma, L. Sigal, and S. Sclaroff. Space-time tree ensemble for action recognition. In *CVPR*, 2015.

[36] S. Mathe and C. Sminchisescu. Actions in the Eye: Dynamic Gaze Datasets and Learnt Saliency Models for Visual Recognition. *TPAMI*, 2014.

[37] S. Narayan and K. R. Ramakrishnan. A cause and effect analysis of motion trajectories for modeling actions. In *CVPR*, 2014.

[38] H. Nguyen, A. Jain, C. D. Anderson, and C. C. Kemp. A clickable world: Behavior selection through pointing and context for mobile manipulation. In *International Conference on Intelligent Robots and Systems*, 2008.

[39] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, 2014.

[40] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[41] M. Pei, Z. Si, B. Z. Yao, and S.-C. Zhu. Learning and parsing video events with goal and intent prediction. *CVIU*, 117(10):1369–1383, Oct. 2013.

[42] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014.

[43] L. Piyathilaka and S. Kodagoda. Human activity recognition for domestic robots. In *Field and Service Robotics*, volume 105, pages 395–408, 2015.

[44] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012.

[45] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[46] B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[47] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[48] Q. Shi, L. Cheng, L. Wang, and A. Smola. Human action segmentation and recognition using discriminative semi-markov models. *IJCV*, 93(1):22–32, 2011.

[49] F. Souza, S. Sarkar, A. Srivastava, and J. Su. Temporally coherent interpretations for long videos using pattern theory. In *CVPR*, 2015.

[50] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[51] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human

activity detection from rgbd images. In *ICRA*, 2012.
[52] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.
[53] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013.
[54] J. Varadarajan, R. Emonet, and J.-M. Odobez. A sequential topic model for mining recurrent activities from long term video logs. *IJCV*, 103(1):100–126, 2013.
[55] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014.
[56] N. N. Vo and A. F. Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *CVPR*, 2014.
[57] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, 2011.
[58] X. Wang and Q. Ji. A hierarchical context model for event recognition in surveillance video. In *CVPR*, 2014.
[59] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.
[60] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for event and object recognition. In *ICCV*, pages 3272–3279, 2013.
[61] C. Wu, I. Lenz, and A. Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. In *RSS*, 2014.
[62] C. Wu, J. Zhang, S. Savarese, and A. Saxena. Watch-n-patch: Unsupervised understanding of actions and relations. In *CVPR*, 2015.
[63] C. Wu, J. Zhang, B. Selman, S. Savarese, and A. Saxena. Watch-bot: Unsupervised learning for reminding humans of forgotten actions. In *ICRA*, 2016.
[64] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *CVPR*, 2014.
[65] S. Yang, C. Yuan, B. Wu, W. Hu, and F. Wang. Multi-feature max-margin hierarchical bayesian model for action recognition. In *CVPR*, 2015.
[66] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos. Robot learning manipulation action plans by watching unconstrained videos from the world wide web. In *AAAI*, 2015.
[67] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *TPAMI*, 35(7):1635–1648, 2013.

Chenxia Wu is a Ph.D. candidate at the School of Computer Science of Cornell University. He received the M.Sc. degree from School of Computer Science of Cornell University in 2015 and the M.Sc. degree in Computer Science of Zhejiang University, China in 2013. He received his BS degree in Computer Science from Southeast University, China in 2010. His research interests are computer vision, machine learning and robotics.

Jiemi Zhang is an algorithm engineer in Didi Chuxing, China. She received the M.Sc. degree from School of Computer Science in Zhejiang University. She received her BS degree in Mathematics from Southeast University, China. Her research interests are machine learning and computer vision.

Ozan Sener received the M.Sc. degree from School of Electrical and Computer Engineering of Cornell University in 2015, the B.Sc. degree from the Electrical Engineering Department of Middle East Technical University, Ankara, Turkey in 2010, and is currently pursuing the Ph.D. degree at School of Electrical and Computer Engineering of Cornell University. His research interests are machine learning, robotics and computer vision.

Bart Selman is a Professor of Computer Science at Cornell University. He previously was at AT&T Bell Laboratories. His research interests include computational sustainability, efficient reasoning procedures, planning, knowledge representation, and connections between computer science and statistical physics. He has (co-)authored over 100 publications, including six best paper awards. His papers have appeared in venues spanning Nature, Science, Proc. Natl. Acad. of Sci., and a variety of conferences and journals in AI and Computer Science. He has received the Cornell Stephen Miles Excellence in Teaching Award, the Cornell Outstanding Educator Award, an NSF Career Award, and an Alfred P. Sloan Research Fellowship. He is a Fellow of the American Association for Artificial Intelligence and a Fellow of the American Association for the Advancement of Science.

Silvio Savarese is an Assistant Professor of Computer Science at Stanford University. He earned his Ph.D. in Electrical Engineering from the California Institute of Technology in 2005 and was a Beckman Institute Fellow at the University of Illinois at Urbana-Champaign from 20052008. He joined Stanford in 2013 after being Assistant and then Associate Professor (with tenure) of Electrical and Computer Engineering at the University of Michigan, Ann Arbor, from 2008 to 2013. His research interests include computer vision, object recognition and scene understanding, shape representation and reconstruction, human activity recognition and visual psychophysics. He is recipient of several awards including the James R. Croes Medal in 2013, a TRW Automotive Endowed Research Award in 2012, an NSF Career Award in 2011 and Google Research Award in 2010. In 2002 he was awarded the Walker von Brimer Award for outstanding research initiative.

Ashutosh Saxena is the CEO of Brain of Things Inc and the Director of Robot Learning Lab at Computer Science department in Cornell University. His research interests include machine learning, robotics and computer vision. He received his Ph.D. in 2009 from Stanford University, and his B.Tech. in 2004 from IIT Kanpur, India. He has won best paper awards in 3DRR, RSS and IEEE ACE. He has also received Sloan Fellowship in 2012, NSF Career award in 2013, RSS Early Career Award in 2014, and was is a Microsoft Faculty Fellow. He has developed robots that perform household chores such as unload items from a dishwasher, arrange a disorganized house, checkout groceries, etc. Previously, he has developed Make3D, an algorithm that converts a single photograph into a 3D model.