UNSUPERVISED STRUCTURED LEARNING OF HUMAN ACTIVITIES FOR ROBOT PERCEPTION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Chenxia Wu

August 2016

UNSUPERVISED STRUCTURED LEARNING OF
HUMAN ACTIVITIES FOR ROBOT PERCEPTION

Chenxia Wu, Ph.D.
Cornell University 2016

Learning human activities and environments is important for robot perception. Human activities and environments comprise many aspects, including a wide variety of human actions and various objects that interact with humans, which make their modeling very challenging. We observe that these aspects are related to each other spatially, temporally and semantically. They form sequential, hierarchical or graph structures. Understanding these structures is key to the learning algorithms and systems of robot perception.

Therefore, this thesis focuses on *structured modeling* of these complex human activities and environments using *unsupervised learning*. Our unsupervised learning approaches can detect hidden structures from the data itself, without the need for human annotations. In this way, we enable more useful applications, such as forgotten action detection and object co-segmentation.

While structured models in supervised settings have been well-studied and widely used in various domains, discovering latent structures is still a challenging problem in unsupervised learning. In this work, we propose unsupervised structured learning models, including causal topic models and fully connected Conditional Random Field (CRF) auto-encoders, which have the ability to model more complex relations with less independence. We also design efficient learning and inference optimizations that maintain the tractability of computations. As a result, we produce more flexible and accurate robot perceptions

in more interesting applications.

We first note that modeling the hierarchical semantic relations of objects and objects' interactions with humans is very important for developing flexible and reliable robotic perception. We therefore propose a *hierarchical semantic labeling* algorithm to produce scene labels at different levels of abstraction for specific robot tasks. We also propose unsupervised learning algorithms to leverage the interactions between humans and objects, so that the machine can automatically discover the useful common object regions from a set of images.

Second, we note that it is important for a robot to be able to detect not only what a human is currently doing, but also more complex relations, such as action temporal and human-object relations. Thus, the robot is able to achieve better perception performance and more flexible tasks. Thus, we propose a causal topic model to incorporate both short-term and long-term temporal relations between human actions, as well as human-object relations, and we develop a new robotic system that watches not only what a human is currently doing, but also what he has forgotten to do, and reminds the person of the latter where necessary.

In the domain of human activities and environments, we show how to build models that can learn the semantic, spatial and temporal structures in the unsupervised setting. We show that these approaches are useful in multiple domains, including robotics, object recognition, human activity modeling, image/video data mining and visual summarization. Since our techniques are unsupervised and structured modeled, they are easily extended and scaled to other areas, such as natural language processing, robotic planning/manipulation, multimedia analysis, *etc*.

BIOGRAPHICAL SKETCH

Chenxia Wu was born and grew up in Zhenjiang, a beautiful eastern city in China, which is sitting on the southern bank of the Yangtze River. Before joining the computer science PhD program at Cornell University, he obtained a Bachelor's degree in computer science from Southeast University, China and a master degree in computer science from Zhejiang University, China. He enjoys solving challenging science problems such as machine learning, computer vision, robotics and game theory. He also likes to innovate solutions for breakthrough applications such as assistive robots, home automations, and self-driving cars. He loves cooking and photography in his free time.

To my parents – Hui Wu and Xuemei Li, and my wife – Jiemi Zhang.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Ashutosh Saxena, for his great guidance and support through the years. Besides the knowledge of computer science, especially he taught me how to always pursue breakthroughs in my research and how to solve the challenges step by step. He motivates me to work hard, and more importantly to work smart as well as to think in depth. I was also fortunate to collaborate with Silvio Savarese at Stanford AI Lab. I learned many valuable insights into computer vision research from him. I am very grateful to Bart Selman, who was very supportive to my research and PhD studies.

I am very grateful to the other members of my thesis committee, Charles Van Loan and Thorsten Joachims for their insightful and constructive suggestions on my work.

I would also like to thank my colleagues from Robot learning lab at Cornell: Yun Jiang, Hema Koppula, Ian Lenz, Jaeyong Sung, Ozan Sener, Ashesh Jain, Dipendra K Misra. I am grateful to the helps from members at Stanford CVGL group: Amir R. Zamir, David Held, Yu Xiang, Kevin Chen, Kuan Fang. I also thank to my friends and colleagues from Cornell: Xilun Chen and Pu Zhang. I would like to thank friends and colleagues at Brain of Things Inc: David Cheriton, Deng Deng, Lukas Kroc, Brendan Berman, Shane Soh, Jingjing Zhou, Pankaj Rajan, and Jeremy Mary. I am grateful to my colleagues at Google X: Ury Zhilinsky, Congcong Li, Zhaoyin Jia, Jiajun Zhu, and Junhua Mao.

Finally, I thank my parents Hui Wu and Xuemei Li for the love and support. I especially thank my wife Jiemi Zhang, not only my best partner, but also my soulmate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Learning human activities and environments is important for robot perception. This problem is challenging, as human activities and environments comprise many aspects, including a wide variety of human actions and various objects that interact with humans. We observe that these aspects are related to each other spatially, temporally and semantically. They form sequential, hierarchical or graph structures. Therefore, correctly understanding and discovering these structures would be key to many applications, such as assistive robots, self-driving cars, healthcare monitoring systems, *etc*.

Most approaches to the modeling of human activities and environments are supervised learning, which involve huge cost in terms of manually labeling the data. However, there is much unlabeled image and video data available on the Internet and collected by researchers that could be used for the machine learning of human activities and environments. Moreover, learned knowledge and applications have been limited to annotations, *e.g.*, solely concerned with the identification of individual objects or human actions. Thus, many other interesting structures, such as temporal orderings between human actions, human-object interactions and object semantic hierarchies, are often missing in the existing research, especially in unsupervised settings.

This thesis focuses on *structured modeling* of human activities and environments using *unsupervised learning*. We develop algorithms and systems for learning structures based on composite human activities and their environments, rather than individual human actions or objects. Perception performance is improved as a result, since the recognition of one object/action is used to rec-

ognize others. Furthermore, our unsupervised learning approaches can detect and model hidden structures in the data itself, without the need for human annotations. In this way, we can uncover richer information in the data, which enables more useful applications, such as forgotten action detection and object co-segmentation.

Structured modeling in supervised settings, such as a Conditional Random Field (CRF), have been well-studied and widely used in various domains, including computer vision, robotics, natural language processing and computational biology. These approaches incorporate rich features to capture varieties of relations for specific tasks. However, discovering latent structures by incorporating such rich observation features has been addressed to a much lesser extent in unsupervised learning. Most existing works require strong independence assumptions that limits the modeling ability and their applications, *e.g.*, the first-order Markov assumptions in the hidden Markov model (HMM) [16] and topic independence assumptions in Latent Dirichlet allocation (LDA) [20]. In this work, we propose unsupervised structured learning models, which have the ability to model complex relations with less independence, including hierarchical semantic and spatial relations between different objects, as well as short-range and long-range temporal relations between human actions. In this way, we achieve more flexible and accurate robotic perceptions, more useful applications and richer discovered and mined information. Despite the complexities introduced in our models, we design efficient learning and inference

optimizations that render the computations tractable.

1.1 Human Environments Learning

In many applications, such as robotic assistance and healthcare monitoring, it is important to model the semantic relations between objects, as well as the interaction between objects and humans. These semantic and spatial structures help robots to provide more reliable and convenient services.

First, we observe that, in different robotic tasks, the robot requires semantic labels at various levels of abstraction. For example, say we want a robot to fetch a Coke from the fridge. In order to navigate to the fridge door as a first step, it will be easier for the robot to recognize the high-level object of a fridge or fridge-door class, rather than the detailed fridge-handle class. Yet once the robot gets close enough for manipulation, producing a more detailed fridge-handle class is necessary to open the fridge door. In the final step, if the robot does not detect a Coke in the fridge, it is better that it fetches another soda instead of coming back empty-handed. It must therefore understand that Coke is a type of soda.

Therefore, we build a semantic hierarchy graph to represent these 'is part of' and 'is type of' relationships and propose a *hierarchical semantic labeling* algorithm to let the robot hedge its bets in different tasks. We propose a novel approach which uses mixed integer programming to optimize a model isomorphic to a CRF. Our model encodes relations both between color/depth features and labels, and between neighboring segments, as well as constraints that arise due to the hierarchical nature of labels. We demonstrate that our algorithm improves on the scene labeling performance in the offline experiments and that

the PR-2 robot using our hierarchical semantic labeling achieves a high success rate in real-world robotic scenarios.

Second, as a person interacts with an object in the scene, he/she provides an implicit cue that allows for the identification of the object's spatial extent (*i.e.*, its segmentation mask), as well as the functional or affordances properties of the object (*i.e.*, the object regions that a human touches in order to use it). We propose unsupervised learning algorithms to leverage the interactions between humans and objects in automatically discovering the common interesting object regions from a set of images, called image co-segmentation. These object regions are useful information for robot navigation, manipulation or web image organization and retrieval.

In order to discover the rich internal structure of these human-object interactions and objects' visual similarities, we leverage the power and flexibility of the fully connected CRF in an unsupervised setting and propose a *fully connected CRF auto-encoder*. This model is able to incorporate different types of features from the whole data set, which are easily generalized to different structured modeling applications, such as image segmentation, human activity temporal segmentation and language parsing. We show that modeling the object and human spatial structure using our model improves the image co-segmentation considerably as compared to previous approaches.

1.2 Human Activities Learning

It is important for a robot to be able to detect not only what a human is currently doing, but also more complex relations, such as actions' temporal relations and

human-object relations. The robot is able to thus execute more interesting tasks, such as action anticipation and forgotten action detection.

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 such activities poses several challenges. First, some actions often co-occur in a composite activity but some may not. Second, co-occurring actions can vary in their temporal orderings, *e.g.*, people can first *put-milk-back-to-fridge* then *microwave-milk* instead of the inverse order in the above example, while the ordering is more relevant to the action *fetch-milk-from-fridge*. Moreover, these ordering relations can exist in both the short- and long-range, *e.g.*, *pouring* is followed by *drink* while sometimes *fetch-book* is related to *put-back-book* with a long *read* between the two. Third, the objects the human interacts with are also important to modeling actions and their relations, as some actions can involve common objects.

We consider the video as a sequence of short-term action clips that contain human-words and object-words. An activity concerns a set of action-topics and object-topics that indicate which actions are present and which objects are interacted with. We then propose a causal topic model that relates the words and the topics. This allows us to model long-range action relations that commonly exist in the composite activities, which has posed a challenge for previous works. We demonstrate flexibilities in terms of different structures using our model, and show that modeling the long-term temporal relations and co-occurrence of actions offers the best results. We also contribute a new challenging RGB-D ac-

tivity video data set, recorded by the Kinect v2, which contains several human daily activities as compositions of multiple actions interacting with different objects.

Furthermore, we develop a robotic system using our unsupervised structured learning algorithms. The robot watches what a human is currently doing, detects what he has forgotten to do while performing an activity and, if necessary, reminds the person of the latter, using a laser pointer to indicate the relevant object. This simple setup can be easily deployed on any assistive robot. We then outline the promising results of these robotic experiments.

In the domain of human activities and environments, we show how to build models that can learn the semantic, spatial and temporal structures of unsupervised settings. We demonstrate that these approaches are useful in multiple domains, including robotics, object recognition, human activity modeling, image/video data mining and visual summarization. Since our techniques are unsupervised and structured modeled, they are easily extended and scaled to other areas, such as natural language processing, robotic planning/manipulation or multimedia analysis.

The remainder of this thesis is organized as follows. Chapter 2 presents a hierarchical semantic scene labeling algorithm and its robotic applications. Chapter 3 introduces a human-centered co-segmentation method that leverages human-object interactions to improve co-segmenting common objects from given images. Chapter 4 presents an algorithm that models composite human

activities in a completely unsupervised setting. Chapter 5 describes a robotic system using our human activity and environment modeling. The thesis concludes in Chapter 6 with a discussion of future works.

1.3 First Published Appearances of Described Contributions

Most contributions or their initial versions described in this thesis have first appeared as various publications:

- Chapter 2: Wu, Lenz, Saxena [134]
- Chapter 3: Wu, Zhang, Saxena, Savarese [136]
- Chapter 4: Wu, Zhang, Savarese, Saxena [135, 133]
- Chapter 5: Wu, Zhang, Sener, Selman, Savarese, Saxena, Ashutosh [137, 133, 115]

CHAPTER 2

HIERARCHICAL SEMANTIC LABELING FOR TASK-RELEVANT RGB-D PERCEPTION

2.1 Introduction

In human environments learning, semantic scene labeling is crucial to many robotic tasks, allowing a robot to precisely localize objects, build maps, perform mobile manipulation tasks, and achieve many other goals. In recent work, many algorithms have been developed to produce such a labeling for RGB-D images (e.g., [104, 72, 9, 47]). However, these approaches produce only a *flat* labeling of a scene, ignoring important relationships between the label classes. In this work, we present an algorithm whose output is a *hierarchical* labeling of the scene.

These hierarchical labels are very important for a wide range of robotic applications. Segmenting object parts, such as handles, knobs, and buttons, separately from the body of the object is critical to properly afford most household objects. Understanding hierarchical object classes can also enable a robot to make rational substitutions between objects. Consider, for example, the task of fetching a Coke from a fridge (Fig. 2.1). To open the fridge, the robot must detect and grasp the fridge handle separately from its door. Then, if a Coke is not present in the fridge, it is much more desirable for the robot to return with another soda, such as a Pepsi, than empty-handed.

Using a semantic label hierarchy as shown in Fig. 2.2 enables these behaviors, which could not be realized using flat labels. When labeling with this hier-

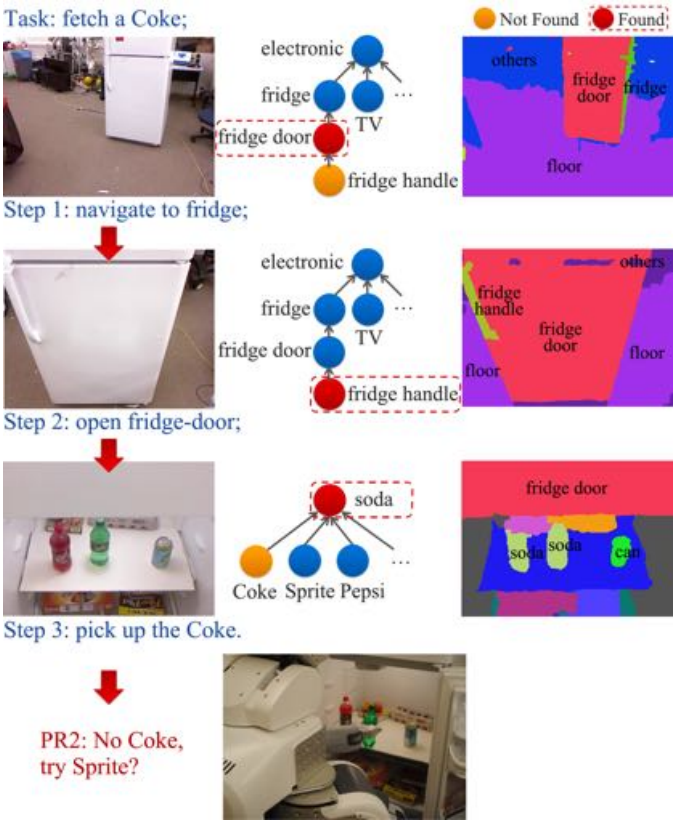


Figure 2.1: **Hierarchical Labels** are produced by our algorithm as required for a robotic task. In the above environment, a robot is asked to fetch a Coke. It needs to perform three sub-tasks: navigate to the fridge, open the fridge door, and pick up the Coke (shown in three rows). For navigation, the robot needs to produce a higher-level *fridge-door* label so that it can approximately navigate close to it. Once it gets closer, producing a more detailed *fridge-handle* label is necessary. In the last step, the robot cannot detect *Coke*, so it fetches another *soda* instead. Such a label hierarchy lets a robot hedge its bets.

archy, each pixel belongs to a series of increasingly-general labels - for example, a pixel of class *fridge-handle* would also be of classes *fridge-door*, *fridge* and *electronics*. This also allows us to represent uncertainty, using a more general class when the algorithm is not sure which low-level class a pixel should belong to.

Conventional flat labeling approaches [104, 47] might simply be applied by flattening all the classes in the semantic hierarchy, but this sacrifices important information. Meanwhile, image classification approaches using semantic hierarchies [33, 99], which predict only one label for each image, cannot be applied to most robotic tasks that require pixel-level labeling of the entire scene. Prop-

erly integrating a semantic hierarchy into the labeling problem is a major challenge, and the main focus of this work.

To this end, we propose a novel approach which uses mixed integer programming to optimize a model isomorphic to a Conditional Random Field (CRF). Our model encodes relations both from color/depth features to labels and between neighboring segments, as well as constraints arising due to the hierarchical nature of labels. It directly integrates hierarchical information, allowing it to represent ambiguities in perception by giving more general labels. In fact, our algorithm allows a desired specificity of the produced labels, allowing for more specific ones for tasks which need them, and more general ones for those that do not. Our approach also combines multiple segmentation trees generated using different metrics to yield more robust labeling results. We demonstrate that all the necessary terms and constraints for our approach can be combined into a model which remains parsimonious and solvable in under 1.5 seconds per image despite incorporating more information than considered in other labeling algorithms.

We validate the performance of our algorithm in an extensive series of experiments, both offline on the NYUD2 dataset [118] and online in a series of robotic experiments using our PR2 robot equipped with a Microsoft Kinect. Our algorithm produces significantly improved results on hierarchical labeling over the state-of-the-art, increasing performance by up to 15%. In robotic experiments, we demonstrate the usefulness of hierarchical as opposed to flat labeling and show that our algorithm can be applied to real-world robotic scenarios, achieving an average success rate of 81% over several challenging tasks. Video of some of these is available at `http://pr.cs.cornell.edu/`

sceneunderstanding/.

In summary, the main contributions of this chapter are:

- We consider a hierarchy of semantic labels when labeling RGB-D scenes, which allows the robot to predict task-relevant labels.
- We design an inference model that incorporates, over a CRF, relations between segment-features and labels, relations between neighboring segments, as well as constraints arising because of the hierarchical nature of the labels. We show that it still remains tractable and is solved by constrained mixed integer programming.
- Our model allows a robot to choose varying levels of specificity in the labels produced.
- We perform extensive evaluation on the NYUD2 dataset as well as on several different robotic tasks.

2.2 Related Work

Scene understanding. Scene understanding from 2D images has been widely explored [111, 113, 45, 27]. Due to the availability of affordable RGB-D sensors, significant effort has been put into RGB-D scene understanding recently [118, 104, 72, 86, 9, 56, 55, 49, 76]. Ren *et al*. [104] developed Kernel Descriptors, highly useful RGB-D feature, and used the segmentation tree to get contextual information. Gupta *et al*. [47] generalized 2D gPb-ucm contour detection to 3D, giving more effective segmentation. Koppula *et al*. [72] and Anand *et al*. [9] used rich contextual information for semantic labeling of 3D

point clouds. Jia *et al*. [55] interpreted objects in a scene by reasoning about blocks, support, and stability. All these works predict flat labels, which are not applicable to many robotic tasks. Instead, our approach outputs a hierarchical labeling, which aids navigation, object finding and rational target substitution in robotic applications.

Visual recognition using semantic hierarchies. Our work is also related to visual recognition using semantic hierarchies [32, 108]. One similar work [33] classified large scale images by optimizing accuracy-specificity trade-offs. Ordonez *et al*. [99] considered predicting labels that people actually use to name an object. Both of these works targeted web image classification, and so predict a single label for each image denoting the most salient object. For many robotic tasks, we must consider pixel level labeling of multiple objects in a complex scene using a semantic hierarchy.

Robotic tasks using vision. There is also a huge body of works using vision algorithms to help perform different robotic tasks [46, 106, 50, 83], such as object grasping [112, 39, 79], navigation [12, 73], trajectory control [116], and activity anticipation [70]. Many works focused on improving SLAM techniques to better depict an environment for planning and navigation [95, 78], such as incremental smoothing and mapping using the Bayes Tree [62], real-time visual SLAM over large-scale environments [132], and object level SLAM [110]. Milford [92], He and Upcroft [48] proposed a place recognition algorithm for mobile robots. Katz and Brock [64] developed interactive segmentation for observing object motion during manipulation. Pangercic *et al*. [100] built semantic object maps for manipulation tasks for an autonomous service robot. Hinkle and Edwin [51] proposed a technique for functionally classifying objects using features obtained

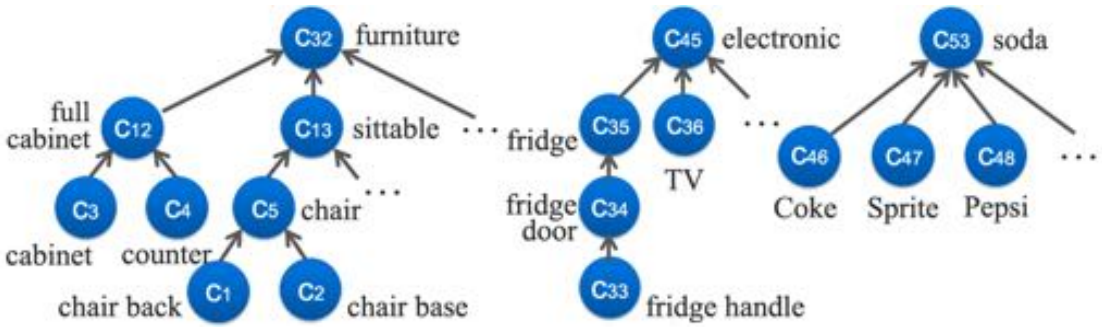


Figure 2.2: **Semantic hierarchy graph.** Each node denotes a class and each directed edge denotes a 'belong to' relation.

through physical simulations.

2.3 Overview

The input to our algorithm is a co-registered RGB and Depth image pair $I \in \mathbb{R}^{m \times n \times 3}, D \in \mathbb{R}^{m \times n}$, where m, n are the image height and width. Our goal is to predict the label of each pixel and output the label matrix $L \in C^{m \times n}$, where C is the set of possible hierarchical semantic labels. We achieve this by mapping a semantic hierarchy graph to the segmentation tree built on the input image. We will first introduce the semantic hierarchy graph and the segmentation tree in this section.

Semantic hierarchy graph. For many robotic actions, we need semantic labels at different levels of abstraction rather than a simple object level. Therefore, we consider two types of relations in a semantic hierarchy:

- *Is-part-of.* For some robotic tasks, we need detailed localization of specific object parts. For example, to open a fridge, it is much better to know exactly

where the *fridge-handle* is from the labeling rather than to simply guess based on a higher-level *fridge-door* label.

- *Is-type-of.* Understanding which objects belong to the same higher-level semantic class allows a robot to make rational substitutions between such objects. For example, if the robot is sent to find a *Coke* but cannot, it could instead return with any *soda* such as a *Pepsi*.

We represent this semantic hierarchy by a directed acyclic graph, called a *semantic hierarchy graph*, where the nodes $C = \{c_k\}$ represent the possible labels and the edges represent one of aforementioned relations. See Fig. 2.2 for an example.

Segmentation tree of the RGB-D image. We begin by segmenting the image into small segments. This gives us a set of candidate segments $\{s_i\}$ to label. If a segment is too small, visual and/or geometric information might be limited; if it is too large, it might straddle a class boundary. We therefore build a segmentation tree and label over this tree. In detail, we first obtain leaf node over-segmentations using a gPb-ucm approach extended for RGB-D images [47]. Second, we merge the most similar pairs of nodes step-by-step based on a similarity measure (the gPb-ucm boundary value)[1], forming a tree as shown in Fig. 2.3.

Note that mapping the semantic hierarchy graph to the segmentation tree is challenging, because both labels and segments are hierarchical rather than flat as in previous works. For example, for a parent segment with two child segments, it is possible to label them with parent-child labels such as labeling

[1] In order to improve the chances of obtaining desirably-sized segments for labeling, we actually build multiple segmentation trees based on different similarity measures [87]: the gPb-ucm boundary value (ucm tree), the similarities between the normals of two neighboring segments (normal tree), and the semantic similarities of any two segments (category tree). These diverse trees provide rich candidate segments for the labeling stage.

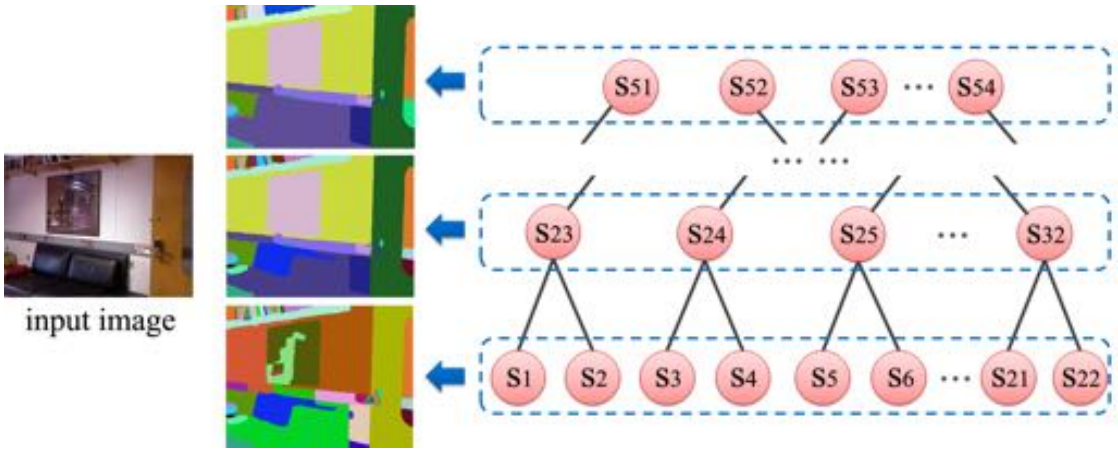


Figure 2.3: **Illustration of segmentation tree.** Pixels are grouped into small segments which are then merged to form a segmentation tree.

the parent as *chair* and the children as *chair-back* and *chair-base*, or to only label the children as two unrelated classes such as *TV* and *cabinet*. Thus, we need to take into account appropriate constraints in designing our CRF-based objective function. For many robotic applications, it is also desirable to be able to select the degree of specificity of the produced labels in the semantic hierarchy. Integrating all these desiderata into a parsimonious model is challenging.

2.4 Preliminaries

Our approach is based on a Conditional Random Field (CRF), modeling the unary terms of, and pair-wise relations between, the segments. We will introduce the unary term and a CRF model to label RGB-D images with flat-labels in this section. We first define the notations in Table 2.1.

Table 2.1: Major notations in this chapter.

c_k	k-th label in the semantic hierarchy graph.
s_i	i-th segment in the segmentation tree.
y_{ik}	$\in \{0, 1\}$. If s_i is labeled with c_k, $y_{ik} = 1$, o.w. $y_{ik} = 0$.
a_i	number of pixels in segment s_i.
a_{ik}	number of pixels of class c_k in segment s_i.
$w_{ik} = a_{ik}/a_i$,	fraction of c_k class pixels in segment s_i.

2.4.1 Unary term of a segment

The unary term relates the features of a segment to its label. Kernel descriptors have been proven to be useful features for RGB-D scene labeling [104], so we extract six such descriptors from each segment: gradient, color, local binary pattern, depth gradient, spin, surface normals, and KPCA/self-similarity. The feature vector of segment s_i is denoted as $\mathbf{z}_i$. We then use the fraction of c_k class pixels in segment : $w^*_{ik} = a_{ik}/a_i$ as a confidence score for s_i belonging to c_k. Since each pixel belongs to several ground-truth classes in the hierarchy such as *chair-back*, *chair*, *sittable*, *furniture*, we treat this as a linear regression problem rather than a classification problem as in previous work [104]. In detail, ridge linear regression is used to train the linear prediction function $\hat{w}_{ik} = \theta_k^\top \mathbf{z}_i$.

2.4.2 Labeling RGB-D Images with Flat Labels

Previous work in [9] started by dividing the RGB-D image into small segments, with the goal of labeling each segment from a flat label set $\{c_k\}$. They then used a CRF to model the unary terms of and pair-wise relations between the segments. Since each segment is allowed to belong to only one class, we have the constraint

$\sum_{c_k} y_{ik} = 1$. The objective function is as follows:

$$\texttt{FlatSeg-FlatLabel}_{\mathbf{y}}(\hat{\mathbf{w}}, \Phi):$$

$$\max_{\mathbf{y}} \overbrace{\sum_{s_i, c_k} y_{ik} \hat{w}_{ik}}^{\text{unary terms}} + \overbrace{\sum_{(s_i, s_j) \in \mathbb{N}, c_k} y_{ik} y_{jk} \Phi(s_i, s_j)}^{\text{edge terms}}, \tag{2.1}$$

$$s.t. \quad \sum_{c_k} y_{ik} = 1 \ \ \forall s_i, \quad y_{ik} \in \{0, 1\}.$$

Here the unary term is $\hat{w}_{ik}$, the edge term is $\Phi(s_i, s_j) = \alpha \exp(-\beta\ gPb(s_i, s_j))$, in which $gPb(s_i, s_j)$ is the gPb-ucm boundary weight between s_i, s_j, and α, β are two weighting parameters. The edge term encourages neighboring segments $(s_i, s_j) \in \mathbb{N}$ with small boundaries to take the same label.

2.5 Hierarchical Semantic Labeling

In this section, we will describe an improved CRF model with constraints which allow labeling over semantic trees using hierarchical labels. We first define the notations in Table 2.2.

Table 2.2: Major notations in hierarchical semantic labeling.

$\pi(v)$	a function that takes a vertex v in a directed graph and returns the set of its ancestors, including itself.
$\dot{\pi}(v)$	the set of ancestors without v itself: $\pi(v) - \{v\}$.
$\check{s}_l$	l-th leaf node segment in the segmentation tree.
H_l	hierarchical relation graph of the ancestor set $\pi(\check{s}_l)$.
$\mathbb{Q}_{lt}$	t-th maximal independent set of graph H_l.

2.5.1 Labeling Segmentation Trees with Flat Labels

Now we describe how we label a segmentation tree, where flat segments are merged to form a tree as in Fig. 2.3. As some segments in the tree overlap, we first need to select which ones to label, and second predict their labels. We achieve this by enforcing that, for each leaf node segment, only one of its ancestors (including itself) is labeled. This is because a pixel can have only one label in a flat labeling scheme while these segments are overlapping. So following constraints are added.

Non-overlapping constraints (NO-CT). We replace the sum-to-one constraint $\sum_{c_k} y_{ik} = 1, \forall s_i$ in Eq. 2.1 with $\sum_{s_i \in \pi(\check{s}_l), c_k} y_{ik} = 1, \forall \check{s}_l$. Since all leaf nodes are considered, every pixel is labeled with exactly one label. We also need to ensure that the area of the child segment vs. the parent segment is accounted for in the objective function. We therefore weight each $\hat{w}_{ik}$ by the total number of pixels a_i of the segment s_i. The objective function then becomes:

$$\begin{aligned} &\texttt{TreeSeg-FlatLabel}_{\mathbf{y}}(\hat{\mathbf{w}}, \mathbf{a}, \Phi) : \\ &\max_{\mathbf{y}} \overbrace{\sum_{s_i, c_k} y_{ik} \hat{w}_{ik} a_i}^{\text{unary terms}} + \overbrace{\sum_{(s_i, s_j) \in \mathbb{N}, c_k} y_{ik} y_{jk} \Phi(s_i, s_j)}^{\text{edge terms}}, \\ &s.t. \overbrace{\sum_{s_i \in \pi(\check{s}_l), c_k} y_{ik} = 1 \;\; \forall \check{s}_l}^{\text{NO-CT}}, \;\; y_{ik} \in \{0, 1\}. \end{aligned} \tag{2.2}$$

2.5.2 Labeling Segmentation Trees with Hierarchical Labels

When hierarchical labels are introduced, the following interesting property emerges: even if a child node is labeled, its ancestors can be labeled with its

ancestor classes. This complicates the specification of constraints in the model, so we add following hierarchical relation constraints. We summarize our RGB-D hierarchical semantic labeling approach in Alg. 1.

Algorithm 1 RGB-D Hierarchical Semantic Labeling.

Input: RGB and Depth image matrix I, D.
Output: Pixel-level label matrix L.
1. Obtain segment set $\{s_i\}$ by building the segmentation tree on I, D (Section 2.3);
2. Extract feature $\mathbf{z}_i$ from each segment s_i (Section 2.4.1);
3. Compute terms $a_i, \hat{w}_{ik}, \tilde{r}_k, \Phi$ in $\texttt{OpTreeSeg-HierLabel}_{\mathbf{y},\xi}(\hat{\mathbf{w}}, \mathbf{a}, \tilde{\mathbf{r}}, \Phi)$ Eq. 2.5: $\hat{w}_{ik} = \theta_k^\top \mathbf{z}_i$, $\tilde{r}_k = r_k^\eta$, $\Phi(s_i, s_j) = \alpha \exp(-\beta\, gPb(s_i, s_j))$ (Section 2.4.1, 2.4.2, 2.5.2);
4. Obtain ancestor-set $\pi(\check{s}_l)$ for each leaf node $\check{s}_l$;
6. Find hierarchical relations for each $\pi(\check{s}_l)$: $\Omega_l = \{(s_i, s_j, c_k, c_z) | c_z \in \dot{\pi}(c_k), s_j \in \dot{\pi}(s_i), \forall s_i, s_j \in \pi(\check{s}_l), \forall c_k, c_z\}$, (Section 2.5.2 (1));
7. Build hierarchical relation graph $H_l = (\mathbb{V}_l, \mathbb{E}_l)$: $\mathbb{V}_l = \{y_{ik}, \forall s_i \in \pi(\check{s}_l), \forall c_k\}$, $\mathbb{E}_l = \{(y_{ik}, y_{jz}), \forall (s_i, s_j, c_k, c_z) \in \Omega_l\}$, (Section 2.5.2 (2));
8. Enumerate maximal independent set $\mathbb{Q}_{lt}$ on each H_l (Section 2.5.2 (3));
9. Solve $\texttt{OpTreeSeg-HierLabel}_{\mathbf{y},\xi}(\hat{\mathbf{w}}, \mathbf{a}, \tilde{\mathbf{r}}, \Phi)$ Eq. 2.5 (Section 2.6);
10. Label each pixel p with the most specific label from the set $\{c_k | p \in s_i \;\&\; y_{ik} = 1\}$: $L_p = \arg\max_{c_k} r_k$ subject to $p \in s_i \;\&\; y_{ik} = 1$.

Hierarchical relation constraints (HR-CT). Now, we allow labeling more than one segment in $\pi(\check{s}_l)$ with hierarchical labels, such as labeling the parent node as *chair* and the child as *chair-back*. To achieve this, we do the following:

1. *Find hierarchical relations.* We first define a tuple (s_i, s_j, c_k, c_z), called *hierarchical relation* if it follows $c_z \in \dot{\pi}(c_k), s_j \in \dot{\pi}(s_i)$. This allows the pair of segments $(s_i, s_j) \in \pi(\check{s}_l)$ to be labeled with (c_k, c_z) respectively, as their order is consistent

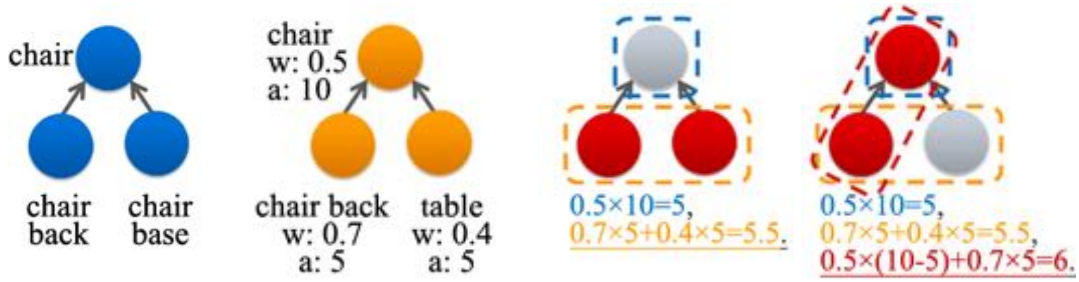
in both the segmentation tree and the semantic hierarchy graph. All such tuples comprise a set Ω_l for each $\pi(\check{s}_l)$.

2. *Build hierarchical relation graph.* In order to find all the constraints in each ancestor-set $\pi(\check{s}_l)$ considering both the non-overlapping and hierarchical labeling properties, we build a undirected graph $H_l = (\mathbb{V}_l, \mathbb{E}_l)$, called *hierarchical relation graph*, of which the vertices are all possible assignments: $\mathbb{V}_l = \{y_{ik}, \forall s_i \in \pi(\check{s}_l), \forall c_k\}$ and edges link vertices if they follow the hierarchical relation: $\mathbb{E}_l = \{(y_{ik}, y_{jz}), \forall (s_i, s_j, c_k, c_z) \in \Omega_l\}$.

3. *Find constraints on the hierarchical relation graph.* Following the hierarchical relation, if two vertices (y_{ik}, y_{jz}) on H_l are linked by an edge, they can be both set to one. Otherwise, at most one can be set to one following the non-overlapping constraint. To give efficient and sufficient constraints, we constrain the sum of all y_{ik} in each maximal independent set (the set of vertices, no pair of which are adjacent) to be not greater than one. The problem then becomes to enumerate all maximal independent sets[2] $\{\mathbb{Q}_{lt}, t = 1, \ldots\}$ of H_l. In practice, we will introduce a parsimonious model (Sec. 2.6), leading to a sparse graph H_l thus more efficient constraint-finding. After finding these sets, we add the constraints $\sum_{y_{ik} \in \mathbb{Q}_{lt}} y_{ik} \leq 1, \forall c_l, t$. To further ensure that all pixels to be labeled, we add the completeness constraints (CM-CT) $\sum_{s_i \in \pi(\check{s}_l), c_k} y_{ik} \geq 1, \forall l$ to ensure at least one segment in each $\pi(\check{s}_l)$ to be labeled.

4. *Overlapping unary correction.* To give even weighting for each pixel, we also modify the unary term for the overlapping pixels when both parent and child segments are labeled. If y_{ik} and y_{jz} are both set to 1, $y_{ik}y_{jz} = 1$, when

[2]To enumerate maximal independent sets of H_l, we first divide H_l into a subgraph $(\tilde{\mathbb{V}}_l, \emptyset)$, where $\tilde{\mathbb{V}}_l$ are all isolated vertices in H_l, $\emptyset$ is the empty edge set, and another subgraph $\tilde{H}_l = (\mathbb{V}_l - \tilde{\mathbb{V}}_l, \mathbb{E}_l)$. Then we enumerate all maximal independent sets $\{\tilde{\mathbb{Q}}_{lt}, t = 1, \cdots\}$ of $\tilde{H}_l$ by enumerating all cliques of its complementary graph, which is a well-studied problem in graph theory [6, 23] and is solved by the Bron-Kerbosch algorithm [23] in our approach. Finally, $\mathbb{Q}_{lt} = \tilde{\mathbb{Q}}_{lt} \cup \tilde{\mathbb{V}}_l$.



(a) Ground truth. (b) Estimated score. (c) NO-CT. (d) NO-CT, HR-CT.

Figure 2.4: **An illustration of the benefit of adding HR-CT.** In the example, (a) shows the ground-truth labels of the segments. (b) gives the highest estimated confidence score $\hat{w}$, its corresponding estimated label and the area a of each node. (c) considers non-overlapping segments selection leading to two possible selections and (d) further considers the hierarchical relation leading to one more possible selection. According to the sum of scores, (c) fails to label the right child node while (d) gives a reasonable labeling, because the (*chair,chair-back*) relation strengthens each other avoiding the possible error incurred by the poor estimated $\hat{w}$.

$(s_i, s_j, c_k, c_z) \in \Omega_l$, we would rather label their overlapping pixels a_i with the more specific label c_k. So, the summation of the unary term would be $\hat{w}_{ik}a_i + \hat{w}_{jz}(a_j - a_i)$. Then, the objective function relating these two terms changes to $y_{ik}\hat{w}_{ik}a_i + y_{jz}\hat{w}_{jz}a_j - y_{ik}y_{jz}\hat{w}_{jz}a_i$.

Note that considering these hierarchical relations and constraints allows the model to avoid possible errors caused by poor local estimates of $\hat{w}$ (see an example in Fig. 2.4).

Choosing the degree of specificity for hierarchical labels. For many robotic applications, it is also desirable to be able to decide the degree of specificity of the produced labels. Here we use the information gain r_k to represent the specificity of each class as in [33]:

$$r_k = \log_2 |C| - \log_2 \sum_{c_z \in C} I(c_k \in \pi(c_z)), \tag{2.3}$$

where the first term is the total number of classes and the second term gives the number of c_k's child nodes. We can see r_k is larger for lower level classes

and smaller for higher levels in the semantic hierarchy. We weight the unary term $\hat{w}_{ik}$ by $\tilde{r}_k = r_k^{\eta}$, where η is the parameter deciding the degree of specificity of prediction.[3]

In summary, the *final objective function* becomes:

$$\begin{aligned}
&\texttt{TreeSeg-HierLabel}_{\mathbf{y}}(\hat{\mathbf{w}}, \mathbf{a}, \tilde{\mathbf{r}}, \Phi) : \\
&\max_{\mathbf{y}} \overbrace{\sum_{s_i, c_k} y_{ik} \hat{w}_{ik} \tilde{r}_k a_i}^{\text{unary terms}} - \overbrace{\sum_{\check{s}_l, (s_i, s_j, c_k, c_z) \in \Omega_l} y_{ik} y_{jz} \hat{w}_{jz} \tilde{r}_z a_i}^{\text{overlapping correction terms}} \\
&\quad + \overbrace{\sum_{(s_i, s_j) \in \mathbb{N}, c_k} y_{ik} y_{jk} \Phi(s_i, s_j)}^{\text{edge terms}}, \\
&s.t. \overbrace{\sum_{y_{ik} \in \mathbb{Q}_{lt}} y_{ik} \leq 1 \;\; \forall \check{s}_l, t}^{\text{NO-CT, HR-CT}}, \overbrace{\sum_{s_i \in \pi(\check{s}_l), c_k} y_{ik} \geq 1 \;\; \forall \check{s}_l}^{\text{CM-CT}}, \\
&\quad y_{ik} \in \{0, 1\}.
\end{aligned} \tag{2.4}$$

After solving this, we label each pixel p with the most specific label from the set: $\{c_k | p \in s_i \;\&\; y_{ik} = 1\}$.

2.6 Efficient Optimization

The quadratic term in the objective function makes optimization difficult. So, we equivalently formulate it by replacing quadratic term $y_{ik} y_{jz}$ with an auxiliary variable ξ_{ij}^{kz} leading to a linear objective which can be solved by a mixed integer

[3] With larger η, the relative weight for more specific class: $(r_i / r_j)^{\eta}, r_i > r_j$ is larger, thus prediction is more specific. The prediction is balanced when $\eta = 0$.

programming (MIP) solver [2]:

$$
\begin{aligned}
&\texttt{OpTreeSeg-HierLabel}_{\mathbf{y},\xi}(\hat{\mathbf{w}}, \mathbf{a}, \tilde{\mathbf{r}}, \Phi): \\
&\max_{\mathbf{y},\xi} \overbrace{\sum_{s_i,c_k} y_{ik}\hat{w}_{ik}\tilde{r}_k a_i}^{\text{unary terms}} - \overbrace{\sum_{\check{s}_l,(s_i,s_j,c_k,c_z)\in\Omega_l} \xi_{ij}^{kz}\hat{w}_{jz}\tilde{r}_z a_i}^{\text{overlapping correction terms}} \\
&\quad + \overbrace{\sum_{(s_i,s_j)\in\mathbb{N},c_k} \xi_{ij}^{kk}\Phi(s_i,s_j)}^{\text{edge terms}}, \\
&s.t. \overbrace{\sum_{y_{ik}\in\mathbb{Q}_{lt}} y_{ik} \le 1 \ \ \forall\check{s}_l, t}^{\text{NO-CT, HR-CT}}, \overbrace{\sum_{s_i\in\pi(\check{s}_l),c_k} y_{ik} \ge 1 \ \ \forall\check{s}_l}^{\text{CM-CT}}, \\
&\quad \xi_{ij}^{kz} \le y_{ik}, \xi_{ij}^{kz} \le y_{jz}, y_{ik} + y_{jz} \le \xi_{ij}^{kz} + 1, \forall s_i, c_k \\
&\quad y_{ik} \in \{0,1\}, \ \ \xi_{ij}^{kz} \in \{0,1\},
\end{aligned}
\tag{2.5}
$$

Parsimonious Model. We observe that there is some redundancy in the above objective, and introduce a parsimonious model to avoid this.

First, we do not need to consider all possible classes for each segment. Classes with low unary terms $\hat{w}_{ik}\tilde{r}_k a_i$ can be omitted for s_i. We consider only the top τ classes, leaving only τ possible y_{ik} for each s_i.

Second, in constraint-finding, some hierarchical relations $(s_i, s_j, c_k, c_z) \in \Omega_l$ are mutually exclusive.[4] So we also consider $\hat{w}_{ik}\tilde{r}_k a_i$ in each hierarchical relation, reducing the number of relations by greedily selecting the top ones with no conflicts. In detail, we first rank all the possible hierarchical relations (s_i, s_j, c_k, c_z) by the sum of unary terms of each pair $w_{ik}\tilde{r}_k a_i + w_{jz}\tilde{r}_z a_j$, all of which consist a candidate relation list. We select the one with the highest score from the list, link the corresponding edge in graph H_l, and remove all its violating relations

[4]For example, consider (s_1, s_3, c_1, c_2) and (s_2, s_4, c_3, c_4), where $s_2 \in \dot{\pi}(s_1), s_3 \in \dot{\pi}(s_2), s_4 \in \dot{\pi}(s_3), c_2 \in \dot{\pi}(c_1), c_3 \in \dot{\pi}(c_2), c_4 \in \dot{\pi}(c_3)$. They are both belong to hierarchical relations according to the definition. However, they are mutually exclusive because when $y_{1,1}$=1, $y_{3,2}$=1, $y_{2,3}$ cannot be 1 as the segment s_2 is within segments s_1, s_3 while class c_3 is higher than classes c_1, c_2.

from the list. We repeat this selection until no relations remain in the list. As a result, the graph H_l becomes sparse with many isolated vertices, since only most confident relations are considered.

The most time consuming step in Alg. 1 is to enumerate the maximal independent sets in step 8. In the worst case it is $O(n_l 3^{h_s \cdot \tau/3})$, where n_l is the number of leaf nodes of and h_s is the height of the segmentation tree, and τ is the number of top considered classes. Though the worst-case running time is non-polynomial, the Bron-Kerbosch algorithm runs much faster in practice [7]. In our experiments on the NYUD2 dataset, it only takes an average of 0.84 and 0.49 seconds per image respectively to find the constraints and optimize the objective using our parsimonious model.

2.7 Scene Labeling Experiments

Data. We evaluate our approach on a hierarchically-labeled subset of the NYUD2 dataset [118], which consists of RGB-D images from a wide variety of environments. We manually labeled a subset of 500 images using a hierarchy. We used 20 most common object classes and one *background* class, and additionally labeled 12 object-part classes and generalized 10 higher level classes. In total, we have 43 classes in the semantic hierarchy. We use the standard split of the NYUD2 dataset, giving 269 training images and 231 test images.

Implementation details. In our experiments, we used six RGB-D kernel descriptors to represent segments for both [104] and our approach. We kept the same setting as in [104] to run their approach: first, we ran gPb-ucm algorithm [10] on both the RGB and depth images separately and linearly com-

bine them to get the gPb-ucm values, then built one segmentation tree by using different values to threshold these values. To make a fair comparison, we also ran the 3D gPm-ucm [47] algorithm to get the gPb-ucm value for both approach [104] and ours. So we denote the approach [104] based on original gPb-ucm as [104] and based on 3D gPb-ucm as [47]+[104].

Evaluation metric. We use three metrics for evaluating scene labeling performance: *cumulative pixel accuracy*, *average information gain* and *average class recall*. We label each scene image at the pixel level and consider it correct to label a pixel with its ground truth label or any of its ancestors, *e.g.*, a pixel of class *chair-back* is also of class *chair*. If $\hat{L}_p$ is a prediction of a pixel label and L_i^* is its ground truth leaf node label, the cumulative pixel accuracy over the whole dataset is defined as: $\sum_p I(\hat{L}_p \in \pi(L_p^*))/n_p$, where $I(.)$ is an indicator function and n_p is the number of pixels in the whole dataset, $\pi(L_p^*)$ is the set of all possible correct predictions including L_p^* and all its ancestors in the semantic hierarchy.

With hierarchical labels, an algorithm can always predict the top-level parent classes and get higher performance, e.g., it is easier to label *furniture* vs *table-leg*. Therefore, following [33], we evaluate the degree of specificity for prediction. Specifically, we compute the information gain (Eq. 2.3) of each predicted class as defined earlier and compute the average.

Recall for class c is defined as: $(\sum_p I(\hat{L}_p \in ch(c) \,\&\, \hat{L}_p \in \pi(L_p^*)))/(\sum_p I(c \in \pi(L_p^*)))$, where $ch(c)$ represent the class set of all c's children plus c itself in the semantic hierarchy. So, the numerator is the number of correctly predicted pixels for class c, and the denominator is the number of pixels with c as ground truth label.

Table 2.3: Average class recall of each class level on NYUD2 dataset.

Recall(%)	class level0	class level1	class level2	class level3
[104]	24.77	30.52	36.02	41.66
[47]+[104]	28.96	34.14	41.69	46.29
Ours(bs+bc)	30.08	36.09	45.96	51.80
Ours(ts+bc)	32.78	41.38	49.26	55.48
Ours(ts+hc)	**33.35**	**44.46**	**51.79**	**61.51**

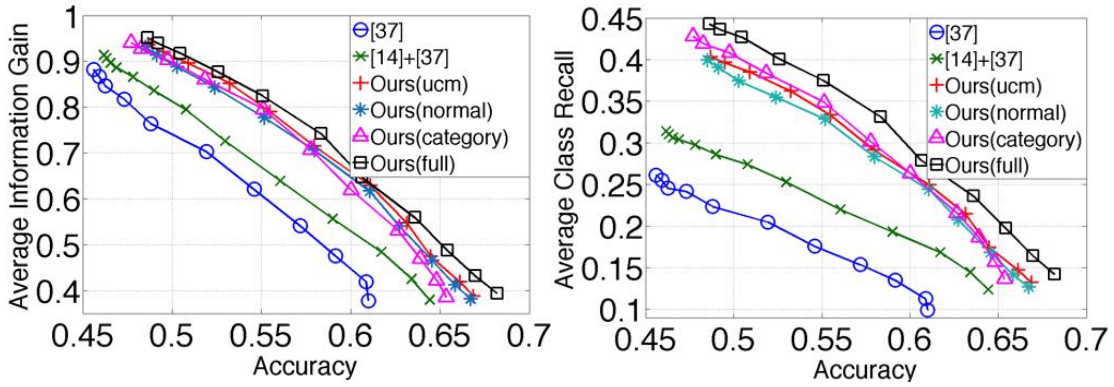
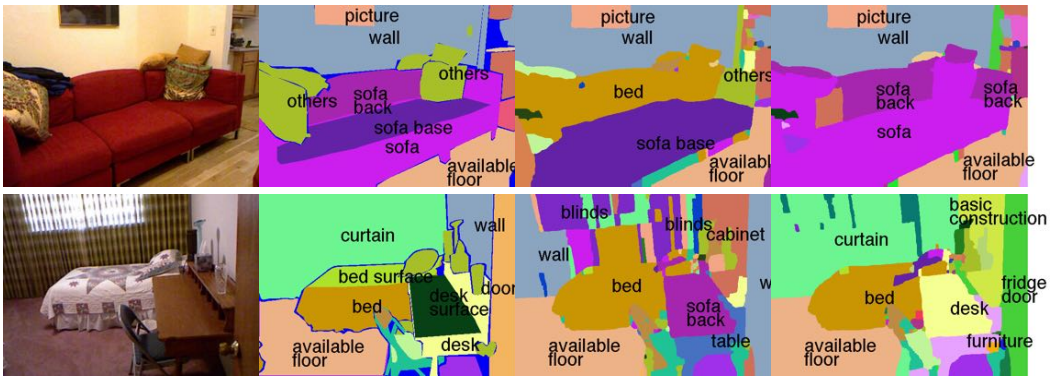


Figure 2.5: **Results on NYUD2 dataset**. For the same degree of specificity for prediction (i.e., same information gain, left) and recall (right), our algorithm performs better.

2.7.1 Results

We first evaluate the average class recall on four levels of the semantic hierarchy. Table. 2.3 summarizes the results. Class level0 contains the base classes, the most specific classes in the tree, *e.g*. object parts and low-level object classes. Higher levels are obtained by merging nodes in each previous level, leading to more general classes. Fig. 2.7 shows all classes for each level.

In this experiment, we train and predict labels on the base classes for flat labeling approaches [104],[47]+[104]. For our approach, we train and predict labels using leaf node segments on the base classes (Ours(ls+bc)), the segmentation tree on the base classes (Ours(ts+bc)) and the segmentation tree on the test class level and all classes below them in the semantic hierarchy (Ours(ts+hc)), with $\eta = 0$ for balanced prediction. These results reveal a number of interesting



(a) Input Image (b) Ground Truth (c) [47]+[104] (d) Ours

Figure 2.6: **Some samples of the results** on NYD2 dataset (small areas are not shown with label names for clarity). In the first row, *sofa back* is labeled correctly since semantic hierarchy (*sofa*,*sofa back*) is considered. In the second row, our algorithm labeled the higher level classes *desk*, *basic construction* instead of *desk surface*, *wall* to avoid possible mistakes with the help of semantic hierarchy.

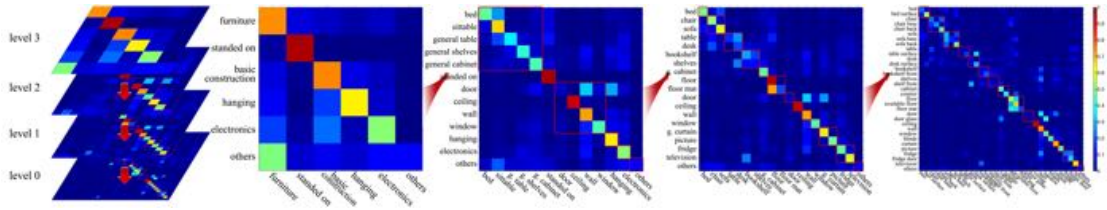


Figure 2.7: **Multi-level confusion matrix** of our final results on NYUD2 dataset. From left to right, the confusion matrix zooms in to see more specific results in the next level below. In each confusion matrix, the red border square gives the classes merged in the next level up.

points as follows:

- The proposed approach Ours(ts+hc) shows the best results at each level, even though predicting more hierarchical labels is harder than the task of the other compared approaches, which only predict the base classes. This is because our approach effectively considers the mapping of the semantic hierarchy to the segmentation tree.
- Labeling on segmentation trees, *e.g*. Ours(ts+bc) and Ours(ts+hc), outperform methods labeling on flat segmentations. In [104], they considered hierarchical segmentation by packing all semantic features together in a tree path. How-

ever, they still label on the flat leaf node segmentations, losing some visual information.

- Prediction becomes easier when classes are more general. Thus, for tasks where specificity is not strictly required, we can predict more general labels to achieve higher accuracy.

To further evaluate the labeling performance using our semantic hierarchy, we plot average information gain vs. accuracy curves (Fig. 2.5-left) and average class recall vs. accuracy curves (Fig. 2.5-right) by varying the degree of specificity for prediction parameter η. We compare approaches [104], [47]+[104] and our approaches using single ucm tree (Ours(ucm)), single normal tree (Ours(normal)), single semantic tree (Ours(category)) and using all three trees (Ours(full)). For the flat labeling approach [104], [47]+[104], we treat each class in the hierarchy as an arbitrary class without considering the hierarchy and train a one-vs-all SVM as in [104]. From these results, we can see that our approaches outperform the flat labeling approaches by a large margin, since the semantic hierarchy is considered. For the same degree of specificity, our algorithms give higher accuracy. Using multiple segmentation trees also improves the performance.

We give two visual examples of labeling results in Fig. 2.6. In the first example, we can see that our algorithm yields a better labeling because semantic hierarchical relations such as (*sofa*,*sofa back*) are considered. The second example shows that the hierarchical labeling can use higher level classes to avoid possible mistakes, such as using *desk* or *basic construction* rather than *desk surface* or *wall*.

To further study the labeling results of our algorithm, we illustrate a multi-

level confusion matrix in Fig. 2.7. We can see that some between-class labeling errors occur within one general class such as *sofa*, *chair*, *stood on*, most of which vanish in the next-higher level. However, some classes are hard to discriminate at any levels, such as *door* and *wall*, *door* and *hanging*. Our algorithm performed poorly for the background class *others* as it contains large variations in visual appearance.

2.8 Robotic Experiments

We evaluated our approach on three robotic tasks: object search, retrieval, and placement. We used a PR2 robot equipped with a Microsoft Kinect as our robotic platform. Table 2.4 shows a summary of the results, listing the perception accuracy ('perc') and end-to-end execution ('exec') separately.

2.8.1 Object Search Experiments

Here the goal for the robot is to locate a particular object in a room by moving around.[5] We compare our approach to [104]. For repeatable experiments, we pre-recorded a search tree at 20 discrete locations, each with a corresponding RGB-D frame (not in the training set).

We ran four separate trials for each algorithm, with the goal of searching for

[5] *Experimental setup details:* The robot moves in discrete steps through the room, effectively moving through a search tree spanning the room. At each node in the tree, it turns to face each potential next location to move to, recording and labeling an RGB-D image for each. The robot will then move to the next location with the highest confidence score for containing the target object. If there are no unvisited neighboring locations, or this score is below some threshold, the robot will instead backtrack.

Figure 2.8: **Fetching a drink with our robot.** A few snapshots of our algorithm running on our PR2 robot for the task of fetching a drink. From left to right: the robot starts some distance from the fridge, navigates to it using our labeling, detects the handle, and grasps it. It then opens the fridge, and finally retrieves a soda from it.

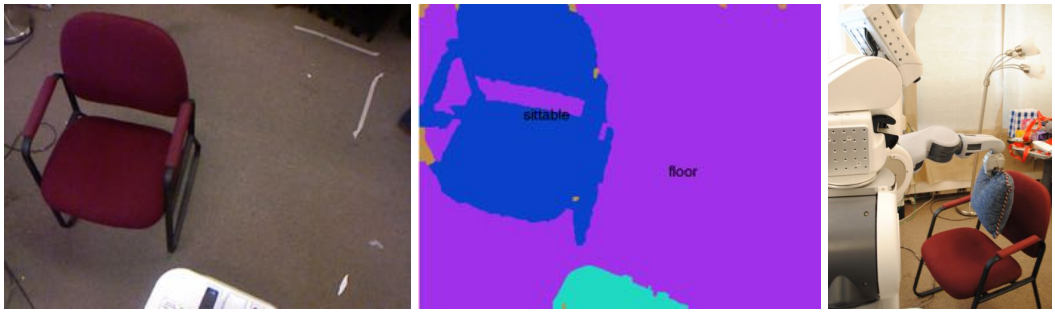


Figure 2.9: **Placing a cushion.** No sofa was present, but the robot used our hierarchy to determine that the chair was another *sittable* object and thus a reasonable place for the cushion.

a *chair back*, *fridge handle*, *mug handle*, and *baseball*. To evaluate performance, the robot to takes a fixed number of steps, and then reports the location at which it had the highest confidence of finding the given object. We score the algorithm's performance based on the overlap ratio of the reported and ground-truth pixels of the target class for that frame, i.e. $|p_d \cap p_g|/|p_d \cup p_g|$, where p_d, p_g are the detected object pixels and ground-truth object pixels.

Fig. 2.10 shows that for any fixed number of steps, our algorithm was able to outperform the approach from [104] for this task. Our algorithm was able to achieve an average overlap ratio of 0.4 after only 6 steps, while [104] took 15, showing that our approach does a better job of informing the search. After 20 steps, both algorithms converged, and ours achieves an average overlap ratio of 0.64 versus the 0.44 ratio from the baseline approach, thus also improving

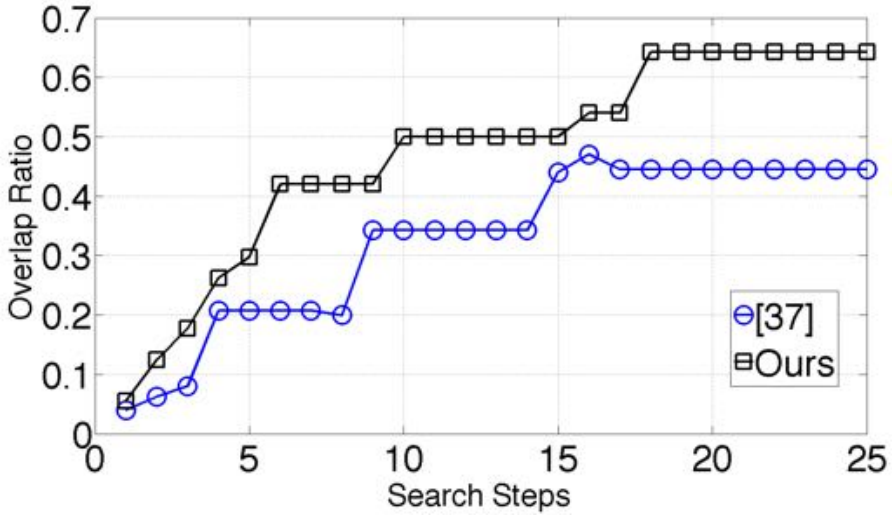


Figure 2.10: **Robot Object Search results.** Figure shows the accuracy vs the number of movement steps taken by the robot.

Table 2.4: Robotic experiment results. Success rates for perception ('perch') and actual robotic execution ('exec') of each task.

	Search		Retrieval				Placement		Average	
	@20 steps		Soda		Bowl		Cushion			
	perc	exec	perc	exec	perc	exec	perc	exec	perc	exec
Flat	44	44	33	33	38	38	50	50	42	42
Hierar. (ours)	64	64	90	80	80	80	100	100	84	81

long-term accuracy.

2.8.2 Object Retrieval Experiments

In this experiment, the robot has to perform a series of perception and motion/manipulation steps for retrieving an object—to fetch a drink from a fridge, and to fetch a bowl from a kitchen counter. The robot first detects and navigates to a semantically appropriate area to find the object in, then locates the target object, grasps it, and brings it back.

In some cases, the desired object may not be available, and the robot is then allowed to retrieve an appropriate substitute. We define this as some other de-

scendant of a class's parent in the semantic hierarchy - for example, *Pepsi* is a substitute for *Coke* because both have the parent class *soda*. A flat labeling scheme is incapable of determining such substitutes, and will report failure if the target class is not found.

From Table 2.4, we can see that our algorithm achieves a very high rate of success for the complex drink-retrieval task shown in Fig. 2.8. Even though this task requires three separate phases of perception, our perception algorithm failed only once in ten trials, failing to find the fridge handle, giving a 90% perception success rate. One more execution failure was due to the fridge door swinging closed before the robot could hold it open, giving an 80% overall success rate. Results for the bowl retrieval experiment were similar. Video of some of these experiments is available at: `http://pr.cs.cornell.edu/sceneunderstanding/`.

At long distances, neither ours nor the baseline labeling algorithms were able to distinguish the handle from the door of the fridge, but our hierarchy informed the robot that the handle was part of the door. The flat labeling approach, meanwhile, lacked this information and simply failed if it could not identify the handle. In fact, the robot was only able to open the fridge 50% of the times using flat labels. Once opened, it could not identify proper substitutes if the desired drink was not present, leading to a mere 33% perception success rate.

2.8.3 Object Placement Experiments

We also performed a series of experiments in which the robot's goal was object placement rather than retrieval. In particular, we considered the task of placing a cushion on a *sofa*, or on some other *sittable* object such as a *chair* if a *sofa* is not present. In every experiment performed, our algorithm was able to successfully locate the sofa, or a substitute if there was no sofa. One example of the robot successfully placing a cushion is shown in Fig. 2.9. By contrast, when using a flat labeling approach, the robot did not understand to place the cushion on another *sittable* surface if the sofa was not present, and thus succeeded only in the 50% of cases.

2.9 Summary

Objects in human environments can be classified into a meaningful hierarchy, both because these objects are composed of parts (*e.g*. fridge-fridge door-fridge handle) and because of different levels of abstraction (*e.g*. drink-soda-Coke). Modeling this is very important in enabling a robot to perform many tasks in these environments. In this chapter, we developed an approach to labeling a segmentation tree with such hierarchical semantic labels. We presented a model based on a CRF which incorporated several constraints to allow labeling using this hierarchy. Our model allows for different levels of specificity in labeling, while still remaining tractable for inference. We showed that our method outperforms state-of-the-art scene labeling approaches on a standard dataset (NYUD2), and demonstrated its use on several robotic tasks.

CHAPTER 3

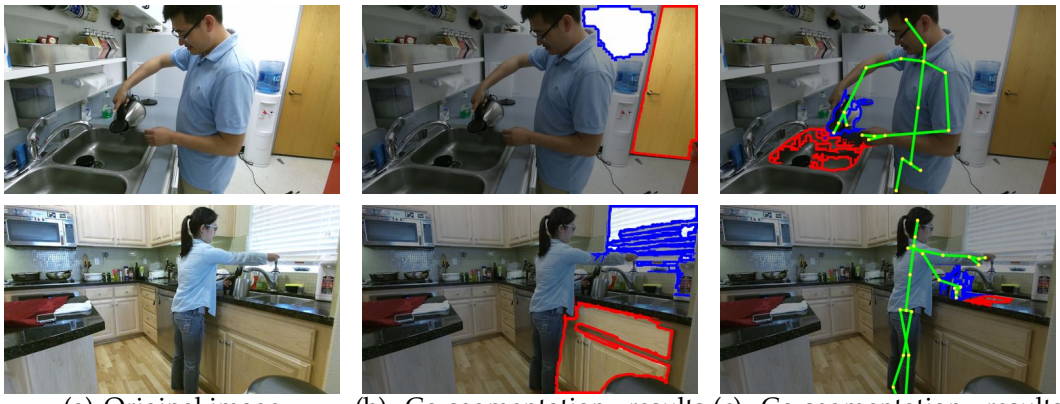
HUMAN CENTERED OBJECT CO-SEGMENTATION

3.1 Introduction

In many applications for humans such as robotic assistance, home automation, it is important to modeling how objects are interacting with humans. Then we can discover more useful and accurate information to enable robot for humans.

In this chapter, we introduce an unsupervised learning algorithms to leverage the interactions between humans and objects in automatically extracting the common semantic regions, called *foregrounds*, from a set of images, which is called image co-segmentation. It provides an unsupervised way to mine and organize the main object segment from the image, which is useful in many applications such as object localization for assistive robotics, image data mining, visual summarization, *etc*.

The challenge of co-segmentation is that semantic labels are not given and we only have the visual information from the images. The only assumption is that the given images share some common semantic regions, *i.e.*, they belong to the same semantic category. So far, the dominating approaches are to co-discover these common regions only relying on their visual similarities [42, 125, 91, 61]. For instance, [53, 126, 14] propose methods for unsupervised pixel-accurate segmentation of "similarly looking objects" in a given set of images. Vicente *et al*. [125] then introduce the concept of "objectness", which follows the principle that the regions of interest should be "objects" such as bird or car, rather than "stuff" such as grass or sky. This helps focus the "attention"



(a) Original image (b) Co-segmentation results using only object appearance (c) Co-segmentation results using object appearance and human

Figure 3.1: We propose a **human centred object co-segmentation** approach by modeling both object visual appearance and human-object interactions. As in the example, human-object interactions help mining of more useful objects (the pots and the sinks) more accurately in the complex backgrounds, with view changes and occlusions. (Output foregrounds are blue, red circled and human skeletons are green colored.)

of the co-segmenter to discover common objects of interest as opposes to irrelevant regions.

However, we argue that in situations where either objects' appearance changes due to intra-class variations are severe (the sinks in Fig. 3.1) or when objects are observed under large view changes or partially occluded by other objects or humans (the pots in Fig. 3.1) or when multiple salient objects are present in the image (the door and the counter are also salient in Fig. 3.1), existing co-segmentation methods will not work well. In this work, we argue that when images do contain humans that use objects in the scene (*e.g.*, a person opens a fridge, or washes dishes in a sink), which is typical in applications such as robotics, navigation or surveillance, *etc.*, we can leverage the interaction between humans and objects to help solve the co-segmentation problem. In essence, as a person interacts with an object in the scene, he/she provides an implicit cue that allows to identify the object's spatial extend (*e.g.*, its segmenta-

tion mask) as well as the functional or affordances properties of the object (*i.e.*, the object regions that an human touches in order to use it).

Therefore, in this work, we propose a *human centred co-segmentation* method whereby the common objects are those often used by the observed humans in the scene and sharing the similar human-object interactions such as the pots and the sinks in Fig. 3.1-(c). We show that leveraging this information improves the results considerably compared to previous co-segmentation approaches.

The main challenge of a human centred object co-segmentation is to modeling the rich relations between objects as well as objects and humans in the image set. To achieve this, we first generate a set of object proposals as foreground candidates from the images. In order to discover the rich internal structure of these proposals reflecting their human-object interactions and visual similarities, we then leverage the power and flexibility of the fully connected conditional random field (CRF) [74] in a unsupervised setting and propose a *fully connected CRF auto-encoder*.

Our model uses the fully connected CRF to encode rich features to detect similar objects from the whole dataset. The similarity depends not only on object visual features but also a novel human-object interaction representation. We propose an efficient learning and inference algorithm to allow the full connectivity of the CRF with the auto-encoder, that establishes pairwise similarities on all pairs of the proposals in the dataset. As a result, the model selects the object proposals which have the most human interactions and are most similar to other objects in the dataset as the foregrounds. Moreover, the auto-encoder allows to learn the parameters from the data itself rather than supervised learning [56, 57] or manually assigned parameters [42, 43, 134] as done in conventional CRF.

In the experiments, we show that our human centred object co-segmentation approach improves on the state-of-the-art co-segmentation algorithms on two human activity key frame Kinect datasets and a musical instrument RGB image dataset. To further show the generalization ability of the model, we also show a very encouraging co-segmentation result on a dataset combining the images without humans from the challenging Microsoft COCO dataset and the images with tracked humans.

In summary, the main contributions of this chapter are:

- We are the first to demonstrate that modeling human is useful to mining common objects more accurately in the unsupervised co-segmentation task.
- We propose an unsupervised fully connected CRF auto-encoder, and an efficient learning and inference approach to modeling rich relations between objects and humans.
- We show the leading performance of our human centred object co-segmentation in the extensive experiments on four datasets.

3.2 Related Work

Co-segmentation. Many efforts have been made on co-segmenting multiple images [24, 60, 88, 13, 85]. The early works used histogram matching [105], scribble guidance [15], or discriminative clustering [53] based on low-level descriptors to extract common foreground pixels. A mid-level representation using "objectness" was considered in [125, 91] to extract similarly looking foreground

objects rather than just common regions. Recently, Fu *et al*. [42] proposed an object-based co-segmentation from RGB-D images using the CRF models with mutex constraints. Our work also generates the object proposals using "objectness" and selects the foregrounds from the candidates. Differently, we are the first one to consider the human interaction to extract the foreground objects.

Early works on co-segmentation only considered two foreground and background classes. Recently, there are many co-segmentation methods which are able to handle multiple foreground objects. Kim *et al*. [67] proposed an anisotropic diffusion method by maximizing the overall temperature of image sites associated with a heat diffusion process. Joulin *et al*. [61] presented an effective energy-based method that combines a spectral-clustering term with a discriminative term, and an efficient expectation-minimization algorithm to optimize the function. Lopamudra *et al*. [94] proposed a method by analyzing the subspace structure of related images. In [28, 43], they presented a video co-segmentation method that extracts multiple foreground objects in a video set. Our work is also able to extract multiple foreground objects by formulating a fully connected CRF auto-encoder, which learns rich information from both objects and humans to extract the foregrounds.

Human-Object Interactions. Modeling human-object interactions or object affordances play an important role in recognizing both objects and human actions in previous works. The mutual context of objects and human poses were modeled in [141] to recognize human-object interactions that improve both human pose estimation and object detection. In [34, 30, 40], human-object interactions in still images or videos were modeled to improve action recognition. Wei *et al*. [131] modeled 4D human-object interactions to improve event and object

recognition. In [56, 57], hallucinated humans were added into the environments and human-object relations are modeled using the latent CRF to improve scene labeling. In this work, we show that human-object interactions provide an important evidence in the unsupervised mining of common objects from images. We also present a novel human-object interaction representation .

Learning Models. Our learning and inference model is extended from the work of CRF auto-encoder [8], which focuses on part-of-speech induction problems using a linear chain sequential latent structure with first-order Markov properties. However, the training and inference become impractical when using fully connected CRF with the auto-encoder. Thus, we introduce an efficient mean field approximation to make the computation still feasible. Our work is also close to the fully connected CRF [74] for the supervised semantic image segmentation, which also uses the mean field approximation to achieve an efficient inference. In contrast with this approach,, we use the mean field approximation to fast compute the gradients of two partition functions, which are exponential growing with object classes without the approximation in the CRF auto-encoder.

3.3 Problem Formulation

Our goal is to segment out common foreground objects from a given set of images. Similar to most co-segmentation approaches, we first generate a set of object proposals $X = \{x_i\}_{i=1,2,\cdots,N}$ as foreground candidates from each image, where N is the total number of object proposals in the given set of images. Here we use selective search [123], which merges superpixels to generate proposals based

on the hierarchical segmentation. This approach has been broadly used as the proposal method of choice by many state-of-the-art object detectors.

Let us assume there are K objects in the images. We then formulate the object co-segmentation as a probabilistic inference problem. We denote the object cluster assignment of x_i as $y_i \in \mathcal{Y} = [1, 2, \cdots, K]$ and $Y = \{y_i\}_{i=1,2,\cdots,N} \in \mathcal{Y}^N$. We want to infer the cluster assignment of a proposal using other proposals with the probability $p(y_i|X)$. We select the object proposals with the highest inference probabilities as the foregrounds, since they have the most human interactions and similar objects as determined by our probabilistic model.

3.4 Model Representation

We propose a fully connected CRF auto-encoder to discover the rich internal structure of the object proposals reflecting their human-object interactions and visual similarities. Unlike previous works relying mainly on the visual similarities between objects, we also encourage the objects with more human interactions and having more similar interactions with other objects to be segmented out. We plot a learned CRF graph from a set of images in Fig. 3.2. In the example, the fridges in (a) and (b) are visually similar, and the fridges in (b) and (c) are similar to each other on human-object interactions even though they look different. These similar objects with more human interactions are more likely to be segmented out in our approach, since they have higher terms in the CRF graph.

The fully connected CRF auto-encoder consists of two parts (The graphic model is shown in Fig. 3.3). The encoding part is modeled as a fully connected

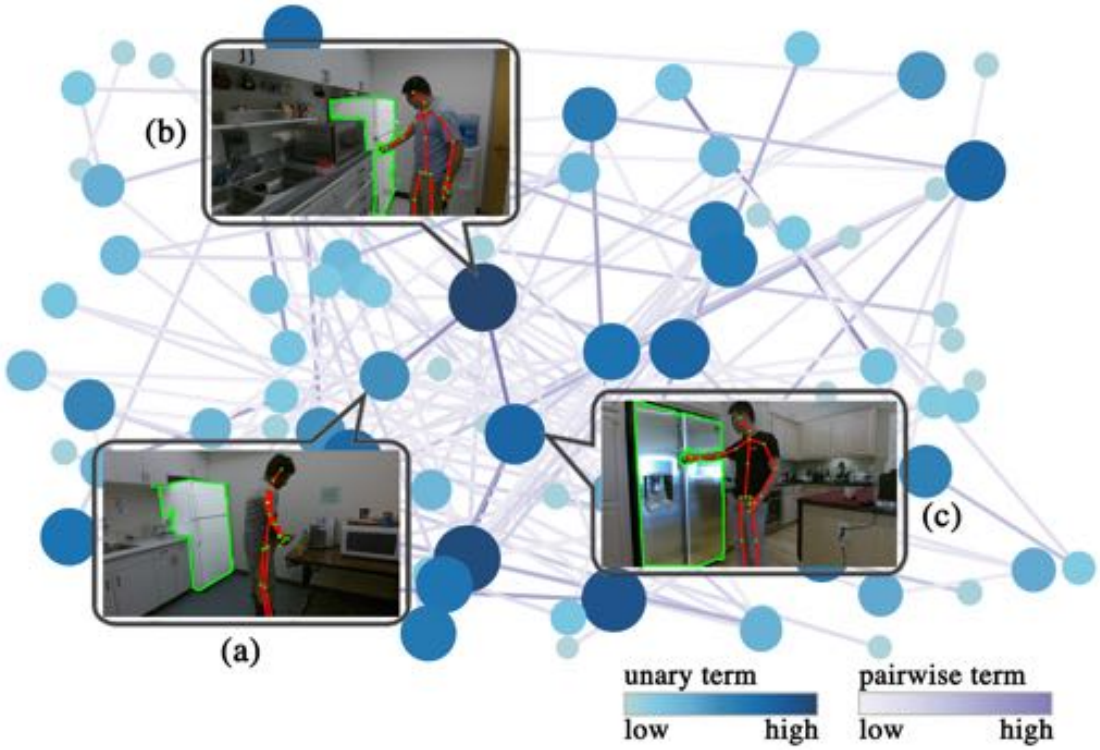


Figure 3.2: **Learned CRF graph from the data.** The nodes are unary terms of object proposals encoding object appearance and human-object interaction features. The edges are pairwise terms encoding similarities on object appearance and human-object interactions. In the example, the fridges in (a) and (b) are visually similar, and the fridges in (b) and (c) have the similar human-object interactions. These similar objects with more human interactions are more likely to be segmented out in our approach, since they have higher unary terms and pairwise terms in the CRF graph. For a good visualization, we only plot the terms with respect to the most likely object cluster for each object proposal and the edges below a threshold are omitted.

CRF, which encodes the observations x_i of the object proposal into object cluster hidden nodes y_i. The reconstruction part reconstructs the hidden nodes by generating a copy of the observation itself $\hat{x}_i$, which considers that a good hidden structure should permit reconstruction of the data with high probability [8],

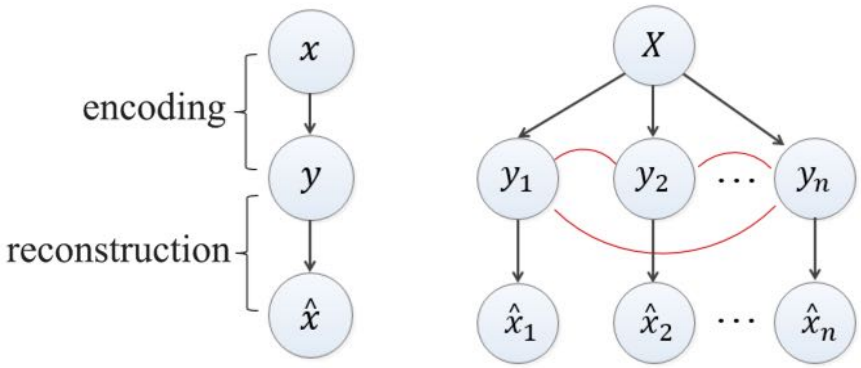


Figure 3.3: Graphic model of our **fully connected CRF auto-encoder**.

3.4.1 Fully Connected CRF Encoding

We first introduce how the fully connected CRF encodes the observations of the object proposals. In CRF, the conditional distribution is given by $P_\lambda(Y|X) = \frac{1}{Z}\exp\{\Phi_\lambda(X,Y)\}$, where $Z = \sum_{Y'\in\mathcal{Y}^N}\exp\{\Phi_\lambda(X,Y')\}$ is a partition function and $\Phi_\lambda(.)$ is called the energy function defined as follows:

$$\Phi_\lambda(X,Y) = \overbrace{\sum_i \lambda^{(u)}(y_i)^\top x_i}^{\text{unary terms}} + \overbrace{\sum_{i<j}\lambda^{(p)}(y_i,y_j)S(x_i,x_j)}^{\text{pairwise terms}}, \tag{3.1}$$

where $\lambda^{(u)}(y_i)^\top x_i$ is the unary term that encodes object visual appearance features and human-object interaction features:

$$\lambda^{(u)}(y_i)^\top x_i = \overbrace{\lambda^{(uo)}(y_i)^\top f_i}^{\text{object appearance}} + \overbrace{\lambda^{(uh)}(y_i)^\top h_i}^{\text{human-object interaction}} . \tag{3.2}$$

Object Appearance Modeling. In Eq. (3.2), $\lambda^{(uo)}(y_i)^\top f_i$ encodes the object visual feature vector f_i by the linear weights $\lambda^{(uo)}(k)$ of each cluster k. It encourages an object to give larger value $\lambda^{(uo)}(k)^\top f_i$ if it belongs to the object cluster k. We use rich kernel descriptors by kernel principal component analysis [21] on the input images: gradient, color, local binary pattern for the RGB image, depth gradient

of depth image and spin, surface normals of point cloud for the Kinect data, which have been proven to be useful features for scene labeling [103, 134].

Human-Object Interaction Modeling. In Eq. (3.2), $\lambda^{(uh)}(y_i)^\top h_i$ encodes the human-object interaction feature vector h_i by the linear weights $\lambda^{(uh)}(k)$ of each cluster k.

To capture the interactions between objects and humans, we propose a novel feature to represent physical human-object interactions such as sitting on the chair, opening the fridge, using their spatial relationships. This feature helps detect those objects used by humans in the scene.

We illustrate the feature representation in Fig. 3.4 for RGB-D data. In detail, we convert the depth image into the real-world 3D point cloud and are also given the 3D coordinate of each joint of a tracked human. Each human body part is represented as a vector starting from a joint to its neighboring joint (Fig. 3.4-(a)(b)). Then we consider a cylinder with the body part as the axis and divide the cylinder into 15 bins by segmenting the body part vertically into 3 parts and the circle surrounding the body part into 5 regions evenly[1] (Fig. 3.4-(b)). Given the point cloud in an object proposal region, we calculate the histogram of the points in these 15 bins and normalize it by the number of the total points in the object proposal as the final feature h_i (Fig. 3.4-(c)). For multiple people in the scene, we compute the max of the histograms of all humans.

For RGB only data, we assume a 2D bounding box of human is given by a person detector. We divide the bounding box evenly into 6×6 bins and compute the normalized histogram of pixels within the bins.

[1]To avoid the affect of points of the human part, we do not consider the innermost circle.

(a) (b) (c)

Figure 3.4: **Our human-object interaction feature for RGB-D data.** In (a), we are given the joints (green dots) of the tracked human, and a object proposal region (green mask). In (b), we divide the cylinder surrounding each body part vector (red line, *spine-base* to *spine-mid* in the example) into 15 bins by segmenting the body part vertically into 3 parts and the circle surrounding the body part into 5 regions. In (c), we compute the histogram of the points in these 15 bins and normalize it by the total number of the total points in the object proposal.

The feature captures the distributions of the object points relative to the human body. It can represent the relative position, size between the humans and objects as well as the human poses and object shapes especially in 3D space. For example in Fig. 3.4-(c), we plot the histogram feature of the body part *spine-base* to *spine-mid* relative to the *chair*. We can see that the distribution reflects the shape of the chair in the following way: the points of the chair-back lie in the distant bins of the upper part (mostly in bin 5 among bin 1-5), and from the chair-back to the chair-base, points become more evenly distributed from the middle part (bin 6-10) to the lower part (bin 11-15).

Since our feature vector h_i has larger values for more human-object interactions, we constrain the weights of the human-object interaction features $\lambda^{(uh)}(k) \geq 0$ to encourage more interactions.

Pairwise Similarity Modeling. In Eq. (3.1), $\lambda^{(p)}(y_i, y_j)S(x_i, x_j)$ is the pairwise term that encodes similarities on object visual appearance and human-object interactions of each object pair:

$$\lambda^{(p)}(y_i, y_j)S(x_i, x_j) = \overbrace{\lambda^{(po)}(y_i, y_j)S(f_i, f_j)}^{\text{object similarities}} + \overbrace{\lambda^{(ph)}(y_i, y_j)S(h_i, h_j)}^{\text{interaction similarities}}, \tag{3.3}$$

where $S(a, b) = \exp\{\frac{-\|a-b\|_2^2}{\delta}\}$ is the similarity function between features[2] . Note that we consider same class to be similar to each other, so we constrain $\lambda^{(p)}(k, k) > 0$ and $\lambda^p(k, l) = 0, k \neq l$.

3.4.2 Reconstruction

To independently generate $\hat{x}$ given y, we define the reconstruction model as a multivariate normal distribution:

$$P_\theta(\hat{x}|y) = N(\hat{x}|\theta^{(\mu)}(y), \theta^{(\Sigma)}(y)), \tag{3.4}$$

where $\theta^{(\mu)}(k), \theta^{(\Sigma)}(k)$ are the mean and covariance of the normal distributions of each class k.

3.5 Model Learning and Inference

In this section, we introduce how we learn the parameters in our fully connected CRF autoencoder from the data and the inference.

Eq. 3.5 gives the parametric model for the observations X:

[2]In practice, we use the augmented features $f_i' = [f_i, -1], h_i' = [h_i, -1], S'(x_i, x_j) = [S(x_i, x_j), -1]$ to also learn the bias of the features. So $\lambda^{(uo)}(k) = [\omega^{(uo)}, b^{(uo)}], \lambda^{(uh)}(k) = [\omega^{(uh)}, b^{(uh)}], \lambda^{(p)}(k, k) = [\omega^{(p)}, b^{(p)}]$ also add another dimension, where ω are the importance weights and $b > 0$ is the bias.

$$
\begin{aligned}
P_{\lambda,\theta}(\hat{X}|X) &= \sum_{Y \in \mathcal{Y}^N} \overbrace{P_\lambda(Y|X)}^{\text{encoding}} \overbrace{P_\theta(\hat{X}|Y)}^{\text{reconstruction}} \\
&= \sum_{Y \in \mathcal{Y}^N} \frac{\exp\{\Phi_\lambda(X,Y)\}}{\sum_{Y' \in \mathcal{Y}^N} \exp\{\Phi_\lambda(X,Y')\}} \prod_i P_\theta(\hat{x}_i|y_i) \\
&= \frac{\sum_{Y \in \mathcal{Y}^N} \exp\{\Phi_\lambda(X,Y) + \log \sum_i P_\theta(\hat{x}_i|y_i)\}}{\sum_{Y' \in \mathcal{Y}^N} \exp\{\Phi_\lambda(X,Y')\}},
\end{aligned}
\tag{3.5}
$$

where λ, θ are the parameters of the encoding parts $P_\lambda(Y|X)$ and the reconstruction parts $P_\theta(\hat{X}|Y)$.

3.5.1 Efficient Learning and Inference

We maximize the regularized conditional log likelihood of $P_{\lambda,\theta}(\hat{X}|X)$ over all the object proposals to learn the parameters λ, θ:

$$
\log L(\lambda, \theta) = R_1(\lambda) + R_2(\theta) + \log \sum_Y P_\lambda(Y|X) P_\theta(\hat{X}|Y), \tag{3.6}
$$

where $R_1(\lambda), R_2(\theta)$ are the regularizers. Note that the space of Y is $[1, 2, \cdots, N]^K$, which is exponential to object cluster number. Maximizing the above function requires computing gradients, while the summation over Y, Y' in the space is intractable.

Mean Field Approximation. In order to make the learning and inference efficient, we use a mean field approximation that has been widely used in the complex graph inference [35, 74]. To use mean field approximation in our model, we first introduce two probabilities of Y:

$$P_{\lambda,\theta}(Y) = \frac{1}{Z}\exp\{\Phi_\lambda(X,Y) + \log P_\theta(\hat{X}|Y)\}, \quad P'_\lambda(Y) = \frac{1}{Z'}\exp\{\Phi_\lambda(X,Y)\}, \tag{3.7}$$

where

$$Z = \sum_{Y\in\mathcal{Y}^n}\exp\{\Phi_\lambda(X,Y) + \log P_\theta(\hat{X}|Y)\}, \quad Z' = \sum_{Y\in\mathcal{Y}^n}\exp\{\Phi_\lambda(X,Y)\}, \tag{3.8}$$

are the partition functions summing over Y to make two probabilities valid. Then Eq.(3.6) can be rewritten as:

$$\log L(\lambda,\theta) = R_1(\lambda) + R_2(\theta) + \log Z - \log Z'. \tag{3.9}$$

Instead of directly computing $P_{\lambda,\theta}(Y), P'_\lambda(Y)$, we use the mean field approximation to compute the distributions $Q(Y), Q'(Y)$ that minimize the KL-divergence $D(Q\|P_{\lambda,\theta}), D(Q'\|P'_\lambda)$. Then all distributions Q, Q' can be expressed as a product of independent marginals, $Q(Y) = \prod_i Q_i(y_i), Q'(Y) = \prod_i Q'_i(y_i)$ [35]. As a result, the gradients of $\log Z$ and $\log Z'$ can be easily computed approximately (see Eq. (3.11)).

By minimizing the KL-divergence, $Q_i(Y), Q'_i(Y)$ has the following iterative update equations:

$$\begin{aligned} Q_i(y_i = k) =& \frac{1}{Z_i}\exp\{\lambda^{(u)}(k)^\top x_i + \lambda^{(p)}(k,k)\sum_{j\neq i} S(x_i,x_j)Q_j(k) \\ &+ \log N(\hat{x}_i|\theta^{(\mu)}(k),\theta^{(\Sigma)}(k))\}, \\ Q'_i(y_i = k) =& \frac{1}{Z'_i}\exp\{\lambda^{(u)}(k)^\top x_i + \lambda^{(p)}(k,k)\sum_{j\neq i} S(x_i,x_j)Q'_j(k)\}. \end{aligned} \tag{3.10}$$

The detailed derivation is in the supplementary material. Following the above equations, the computation of Q, Q' can be done using an iterative update as shown in Algorithm 2.

Algorithm 2 Mean field to compute Q and Q'.

Initialize Q and Q':
  $Q_i(k) = \frac{1}{Z_i} \exp\{\lambda^{(u)}(k)^\top x_i + \log N(\hat{x}_i|\theta^{(\mu)}(k), \theta^{(\Sigma)}(k))\}$,
  $Q'_i(k) = \frac{1}{Z'_i} \exp\{\lambda^{(u)}(k)^\top x_i\}$,
while not converged **do**
  $\hat{Q}_i(k) = \lambda^{(p)}(k,k) \sum_{j \neq i} S(x_i, x_j) Q_j(k)$,
  $Q_i(k) = \exp\{\lambda^{(u)}(k)^\top x_i + \hat{Q}_i(k) + \log N(\hat{x}_i|\theta^{(\mu)}(k), \theta^{(\Sigma)}(k))\}$,
  $\hat{Q'}_i(k) = \lambda^{(p)}(k,k) \sum_{j \neq i} S(x_i, x_j) Q'_j(k)$,
  $Q'_i(k) = \exp\{\lambda^{(u)}(k)^\top x_i + \hat{Q'}_i(k)\}$,
  normalize $Q_i(k), Q'_i(k)$.
end while

In each update, it costs $O(N^2 \times K)$. It also can be used a similar message passing by the high-dimensional filtering as in [74] to speed up the update, which is not the focus of the work. The update was mostly converged within 10 rounds in our experiments (see Fig. 3.7(b)).

Learning. We iteratively learn λ and θ by maximizing the log likelihood with respect to each type of parameters. We update λ using the Adagrad approach [38] which computes the gradients and update θ using EM [31]. Using the mean field approximation described above, we can easily compute the gradients of $\log Z, \log Z'$ in Eq.(3.9), leading to a simple approximation of the gradients:

$$\begin{aligned} \frac{\partial(\log Z - \log Z')}{\partial \lambda^{(u)}(k)} &= \sum_i x_i (Q_i(k) - Q'_i(k)), \\ \frac{\partial(\log Z - \log Z')}{\partial \lambda^{(p)}(k,k)} &= \sum_{i \neq j} S(x_i, x_j)(Q_i(k)Q_j(k) - Q'_i(k)Q'_j(k)). \end{aligned} \tag{3.11}$$

The detailed derivation is in the supplementary material.

Inference. After learning the parameters λ, θ, we can infer the posterior, conditioning on both observations and reconstructions, $\hat{Y} = \arg\max_Y P_{\lambda,\theta}(Y|X, \hat{X})$. This is proportional to $P_{\lambda,\theta}(Y)$ in Eq.(3.7), which can be efficiently computed using the approximation probability $Q(Y) = \prod_i Q_i(y_i)$. So the probability of each object

proposal for each class $p_{\lambda,\theta}(y_i|X,\hat{X}) \propto Q_i(y_i)$, which we use as the confidence score to extract the foreground objects of each image.

3.6 Experiments

3.6.1 Compared Baselines

We compared our human centred object co-segmentation approach with two state-of-the-art algorithms: a multi-class image co-segmentation approach by combining spectral- and discriminative-clustering (Joulin *et al*. 2012 [61]) and a CRF based solution with the manually assigned parameters (Fu *et al*. 2014 [43]). Fu *et al*. 2014 [43] is designed for co-segmenting foregrounds from videos, so we keep their features for the static frame and remove the temporal features. We run their source code on the given RGB images with the default settings. We also evaluate our model using object-only visual features in the experiments.

3.6.2 Evaluations

We use the same evaluation metrics as in multi-class co-segmentation [61]: the intersection-over-union score defined as $\max_k \frac{1}{|\mathbf{I}|}\sum_{i\in\mathbf{I}}\frac{GT_i \cap R_i^k}{GT_i \cup R_i^k}$, where $\mathbf{I}$ is the image dataset, GT_i is the ground-truth region and R_i^k is the region associated with the k-th cluster in image i. The evaluation validates that the foreground object is indeed rather well represented by one of the object clusters in most test cases, which may be sufficient to act as a filter in most applications such as [107]. As discussed in multi-class co-segmentation [61], we set K to be more than the

number of groundtruth foreground object classes plus background class to give better segmentation results. We set $K = 4$ for the data with a single foreground and $K = 6$ for the data with two foregrounds. We also evaluate how performance varies with K in the experiments.

3.6.3 Datasets

Human Activity Key Frame Kinect Dataset We first evaluate on human activity key frames extracted from the activity videos in two datasets: Cornell Activity Dataset-120 (CAD-120)[3] [69] recorded by Microsoft Kinect v1, and Watch-n-Patch[4] [135] recorded by the Microsoft Kinect v2. Each frame in the datasets has a registered RGB and depth image pair as well as tracked human skeletons.

CAD-120 contains activity sequences of ten different high level activities performed by four different subjects. We evaluate the image co-segmentation with the top four existing foreground objects in CAD-120: microwave, bowl, box, cup. For each type, we extract the 60 key frames of the activity videos containing the object. We evaluate the foreground regions using the provided groundtruth bounding box.

Watch-n-Patch activity dataset contains human daily activity videos performed by 7 subjects in 8 offices and 5 kitchens with complex backgrounds. In each environment, the activities are recorded in different views. In each video, one person performed a sequence of actions interacting with different types of objects. We evaluate three types of scenes in the dataset, each of which has two

[3] http://pr.cs.cornell.edu/humanactivities/data.php
[4] http://watchnpatch.cs.cornell.edu/

foreground objects: table and chair, fridge and microwave, pod and sink. For each scene, we extract the 70 key frames of the relevant activity video containing the objects. We label the groundtruth foreground pixels for evaluation.

People Playing Musical Instrument Dataset. We also evaluate on a RGB dataset to see the performance using RGB only features. We use the People Playing Musical Instrument (PPMI) dataset[5], which was used to evaluate recognizing human-object interactions in [141]. It contains RGB images of people 'playing' or 'with' different types of musical instruments. Some of the images have multiple people interacting with the instruments. We evaluate three types of instruments, cello, French horn and violin by randomly selecting 80 images from each class and label the foreground pixels in the image for evaluation.

MS COCO combining with Watch-n-Patch Dataset Since image data with humans are not always available, we finally give the co-segmentation results on the images without humans from the challenging Microsoft COCO (MS COCO) dataset [80] combining with a small portion of images with tracked humans from Watch-n-Patch dataset. The images from MS COCO dataset has more clustered backgrounds from variant sources. We evaluate three classes: chair, fridge and microwave. For each class, we randomly select 50 images from indoor scenes in MS COCO dataset and 20 images from Watch-n-Patch dataset, then combine them as the test set. We use the same RGB features and human features as described above.

Note that one challenge for image co-segmentation is the scalability, as the state-of-the-art algorithms rely on heavy computations on relation graphs. Therefore, most evaluation datasets in previous works [60, 125, 61, 42] have less

[5] http://ai.stanford.edu/~bangpeng/ppmi.html

Table 3.1: Co-Segmentation results on CAD-120 dataset (%).

class	microwave	bowl	box	cup
Joulin *et al*. 2012 [61]	21.6	22.5	19.2	17.7
Fu *et al*. 2014 [43]	47.5	14.9	**40.2**	9.3
Ours (object-only)	45.1	19.7	32.6	22.8
Ours	**54.3**	**24.8**	38.2	**27.9**

Table 3.2: Co-Segmentation results on Watch-n-Patch dataset (%).

class	table	chair	fridge	microwave	pod	sink
Joulin *et al*. 2012 [61]	34.7	17.2	29.9	5.5	5.3	17.9
Fu *et al*. 2014 [43]	21.6	15.7	25.4	**24.5**	21.3	23.6
Ours (object-only)	41.9	26.9	33.1	17.5	20.4	23.2
Ours	**50.0**	**36.4**	**44.7**	20.5	**23.7**	**28.6**

than 50 images per class. In our experiments, the test set has more than 50 but still less than 100 images per class.

3.6.4 Results

We give the results in Table 3.1, 3.2, 3.3, 3.4. We can see that in most cases, our approach performs better than the state-of-the-art image co-segmentation methods. We discuss our results in the light of the following questions.

Did modeling human-object interaction help? In most cases, we can see that our approach to modeling the human-object interaction gives the best result. This is because more human-object interactions give the higher unary term for the interesting objects interacting with the humans, and the similar human-object interactions link the same objects in the CRF graph with larger pairwise terms even though they may not look similar. As a result, we are able to segment out the common object accurately even with view, scale changes, occlusions and

Table 3.3: Co-Segmentation results on PPMI dataset (%).

class	cello	frenchhorn	violin
Joulin *et al*. 2012 [61]	21.9	20.0	18.1
Fu *et al*. 2014 [43]	30.1	40.8	26.4
Ours (object-only)	34.5	41.0	28.3
Ours	**36.2**	**49.2**	**31.5**

Table 3.4: Co-Segmentation results on MS COCO + Watch-n-Patch dataset (%).

class	chair	fridge	microwave
Joulin *et al*. 2012 [61]	4.2	14.1	10.3
Fu *et al*. 2014 [43]	6.9	11.4	9.2
Ours (object-only)	7.5	10.2	10.5
Ours	**12.5**	**17.5**	**15.9**

in complex backgrounds. We show some example results in Fig. 3.5 and Fig. 3.6.

How successful is our fully connected CRF auto-encoder? From the results, we can see that our fully connected CRF auto-encoder model using the object only features also performs better than other algorithms. This is because our model is able to learn the parameters from the data itself rather than manually assigning parameters of the typical CRF model in Fu *et al*. 2014 [43], then the model is more data dependent and does not require much parameter tuning. Though Fu *et al*. 2014 [43] performed well in some cases, the approach is not stable as the parameters are preset. Also, benefit from our efficient learning and inference algorithm, we are able to use fully connected hidden nodes to model the rich relations between all objects and humans in the dataset, so that we have more information to detect the common foreground objects.

Can human information be generalized to the segments without humans? In our first three datasets with humans in the image, there are a few images where humans are not interacting with the foreground objects such as the fridge on the

(a) table-chair (b) fridge-microwave (c) pot-sink

Figure 3.5: Visual examples of our co-segmentation results on Watch-n-Patch dataset.

(a) cello (b) French horn (c) violin

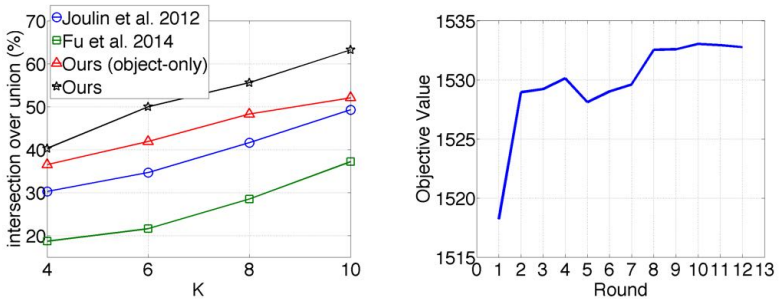
Figure 3.6: Visual examples of our co-segmentation results on PPMI dataset.

right in Fig. 3.5(b). In these cases, our approach was still possible to segment it out correctly, since it was linked to other visually similar objects which are interacting with humans in the fully connected CRF graph.

From the results in Table 3.4 on the MS COCO + Watch-n-Patch dataset, we can see that the task is very challenging since images are from different domains with more clustered backgrounds. Though all compared methods perform not more than 20 percent in accuracy, modeling humans even in a few images still improves the performance.

2D bounding box of human vs. 3D human skeleton. From Table 3.3, we can see that even in 2D images with the bounding box of the detected human, modeling the human-object interactions improves the co-segmentation performance. Us-

ing our proposed 3D human-object representation on the more accurate tracked humans gives more improvements as shown in Table 3.1, 3.2. In the examples in Fig. 3.5(b), we also found that our 3D human-object representation is useful to deal with the occlusions and view changes, which is challenging for object visual appearance only based co-segmentation approaches.



(a) (b)

Figure 3.7: (a). Results of table class on Watch-n-Patch dataset varying with cluster number K. (b). Learning curve of our approach.

How performance varies with cluster number K? We show the performance varying with the cluster number K in Fig. 3.7(a). We can see that the accuracy increase with the class number K as it has higher chance to hit the ground truth regions and more backgrounds are modeled. Our approach has the best performance for each K.

How fast is the learning? We also plot a learning curves of our model in Fig. 3.7(b). The learning of our approach can be converged mostly within 10 iterations.

3.7 Summary

In this chapter, we proposed a novel human centered object co-segmentation approach using a fully connected CRF auto-encoder. We encoded a novel human-object interaction representation and rich object visual features as well as their similarities using the powerful fully connected CRF. Then we used the auto-encoder to learn the parameters from the data itself using an efficient learning even for this complex structure. As a result, we were able to extract those objects which have the most human interactions and are most similar to other objects in the dataset as the foregrounds. In the experiments, we showed that our approach extracted foreground objects more accurately than the state-of-the-art algorithms on two human activity Kinect key frame dataset as well as the musical instruments RGB image dataset. We also showed that our model was able to use the human information in a small portion of images to improve the co-segmentation results.

In the future, we consider extending our human centered object co-segmentation approach into the semi-supervised setting and incorporating temporal information for video data.

CHAPTER 4

UNSUPERVISED LEARNING OF HUMAN ACTIONS AND RELATIONS

4.1 Introduction

In this chapter, we consider modeling the human composite activity which is composed of a sequence of actions (see an example in Fig. 5.1), as perceived by an RGB-D sensor in home and office environments. In the human composite activity such as *warming milk* in the example, there are not only short-range action relations, *e.g.*, *microwaving* is often followed by *fetch-bowl-from-oven*, but there are also long-range action relations, *e.g.*, *fetch-milk-from-fridge* is strongly related to *put-milk-back-to-fridge* even though several other actions occur between them.

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

Most previous works focus on action detection in a supervised learning setting. In the training, they are given fully labeled actions in videos [82, 109, 114], or weakly supervised action labels [37, 22], or locations of human/their interactive objects [77, 122, 97]. Among them, the temporal structure of actions is often discovered by Markov models such as Hidden Markov Model (HMM) [121] and semi-Markov [52, 117], or by linear dynamical systems [17], or by hierarchical grammars [101, 127, 75, 129, 11], or by other spatio-temporal representations [65, 98, 68, 71]. Most of these works are based on RGB features and only model the short-range relations between actions (see Section 5.2 for details).

Different from these approaches, we consider a completely unsupervised

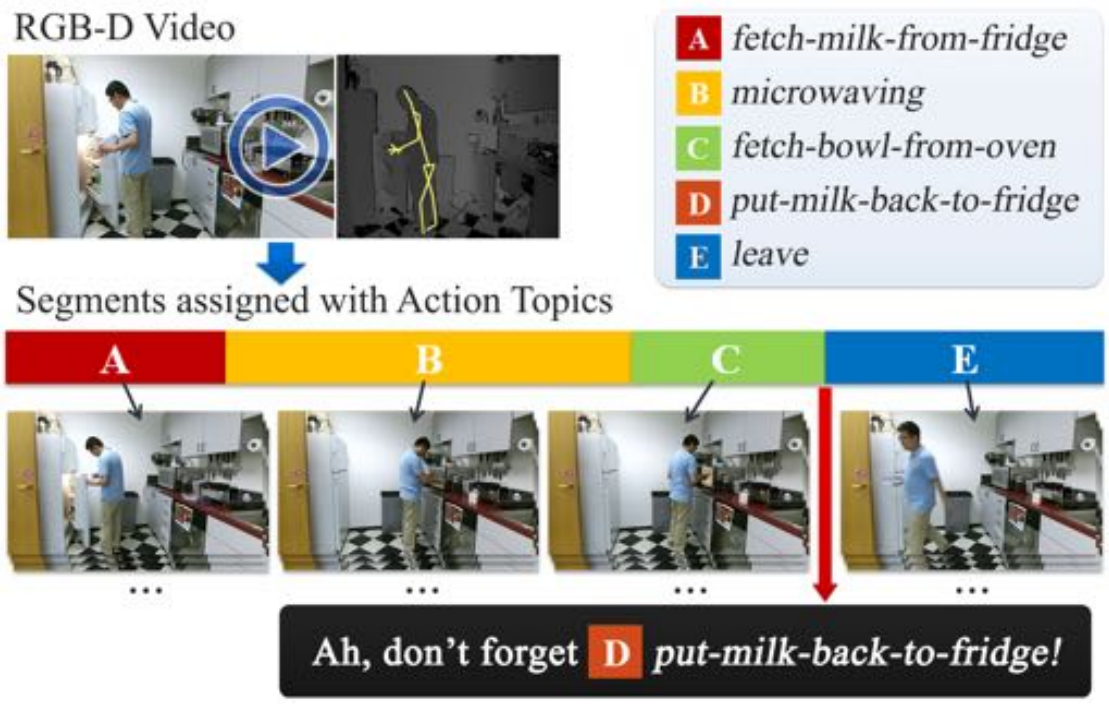


Figure 4.1: Our goal is to automatically segment RGB-D videos and assign action-topics to each segment. We propose a completely unsupervised approach to modeling the human skeleton and RGB-D features to actions, as well as the pairwise action co-occurrence and temporal relations. We then show that our model can be used to detect which action people forgot, a new application which we call *action patching*.

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 *fetch-milk-from-fridge*. We also use the more informative human skeleton and RGB-D features, which show higher performance over RGB only features for action recognition [70, 138, 81].

In order to capture the rich structure in the activity, we draw strong parallels with the work done on document modeling from natural language (e.g., [20]). We consider an activity video as a document, which consists of a sequence of short-term action clips as *action-words*. And 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. Action-words are drawn from these action-topics and has a distribution for each topic. Then we model the following (see Fig. 4.2):

- *Action co-occurrence.* Some actions often co-occur in the same activity. We model the co-occurrence by adding correlated topic priors to the occurrence of action-topics, *e.g.*, *fetch-milk-from-fridge* and *put-milk-back-to-fridge* has strong correlations.
- *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 activity videos. We use the model to temporally segment videos to segments assigned with action-topics. We show that these action-topics are semantically meaningful by mapping them to ground-truth action classes and evaluating the labeling performance.

We then also show that our model can be used to detect forgotten actions in the activity, a new application that we call *action patching*. We show that the learned co-occurrence and temporal relations are very helpful to infer the forgotten actions by evaluating the patching accuracy.

We also provide a new challenging RGB-D activity video dataset recorded by the new Kinect v2 (see examples in Fig. 4.10), in which the human skeletons and the audio are also recorded. It contains 458 videos of human daily activities as compositions of multiple actions interacted with different objects, in which

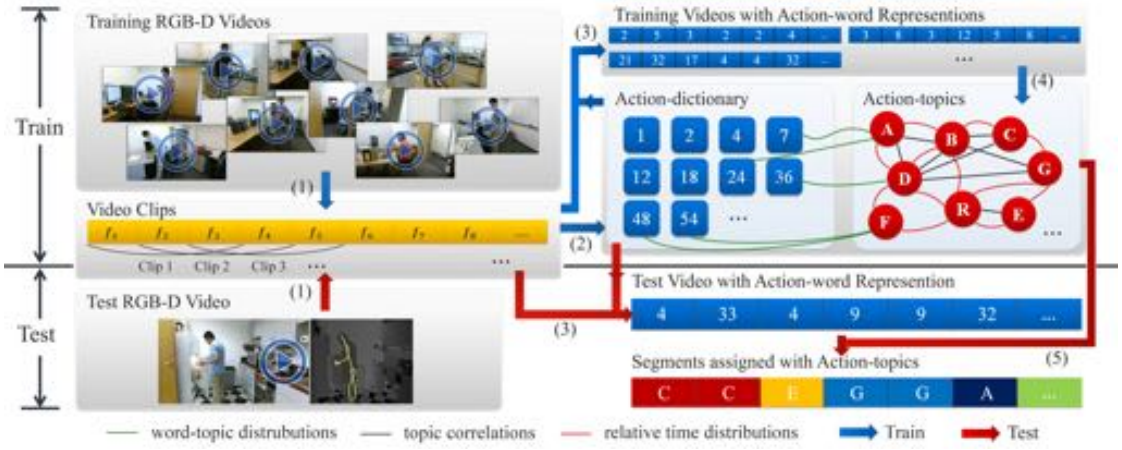


Figure 4.2: **The pipeline of our approach.** Training (blue arrows) follows steps (1), (2), (3), (4). Testing (red arrows) follows steps (1), (3), (5). The steps are: (1) Decompose the video into a sequence of overlapping fixed-length temporal clips. (2) Learn the action-dictionary by clustering the clips, where the cluster centers are action-words. (3) Map the clips to the action-words in the action-dictionary to get the action-word representation of the video. (4) Learn the model from the action-word representations of training videos. (5) Assign action-words in the video with action-topics using the learned model.

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 and non-parametric, 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 recognition, as well as in a new application action patching.
- We provide a new challenging RGB-D activity dataset recorded by the new Kinect v2, which contains videos of multiple actions interacted with different objects.

4.2 Related Work

Most previous works on action recognition are supervised [77, 37, 98, 82, 109, 122, 22, 90]. Among them, the linear models [121, 52, 117, 17] are the most popular, which focus on modeling the action transitions in the activities. More complex hierarchical relations [101, 127, 75, 129] or graph relations [11] are considered in modeling actions in the complex activity. Although they 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.

There also exist some unsupervised approaches on action recognition. Yang *et al*. [139] develop a meaningful representation by discovering local motion primitives in an unsupervised way, then a HMM is learned over these primitives. Jones *et al*. [59] propose an unsupervised dual assignment clustering on the dataset recorded from two views.

Different from these approaches, we use the richer human skeleton and RGB-D features rather than the RGB action features [128, 63]. 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.

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 [114, 120, 124, 138, 81]. Besides of the human skeletons, we also 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 [70, 71, 97, 129]. Most of them depend on correct object tracking or local motion changes. They lost the high-level action relations which can be captured in our model.

Our work is also related to the topic models. LDA [20] was the first hierarchical Bayesian topic model and widely used in different applications. The correlated topic models [18, 66] add the priors over topics to capture topic correlations. A topic model over absolute timestamps of words is proposed in [130] and has been applied to action recognition [41]. However, the independence assumption of different topics would lead to non smooth temporal segmentations. 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.

4.3 Overview

We outline our approach in this section (see approach pipline in Fig. 4.2). The input to our system is RGB-D videos with the 3D joints of human skeletons from Kinect v2. We first decompose a video into a sequence of overlapping fixed-length temporal clips (step (1)). We then extract the human skeleton features and the human interactive object features from the clips (introduced in Section. 4.4), which show higher performance over RGB only features for action recognition [70, 138, 81].

In order to build a compact representation of the action video, we draw parallels to document modeling in the natural language [20] to represent a video as a sequence of words. We use k-means to cluster the clips to form an *action-*

dictionary, where we use the cluster centers as *action-words* (step (2)). Then, the video can be represented as a sequence of action-word indices by mapping its clips to the nearest action-words in the dictionary (step (3)). And an activity video is about a set of *action-topics* indicating which actions are present in the video.

We then build an unsupervised learning model (step (4)) that models the mapping of action-words to the action-topics, as well as the co-occurrence and the temporal relations between the action-topics. Using the learned model, we can assign the action-topic to each clip (step (5)), so that we can get the action segments, the continuous clips with the same assigned topic.

The unsupervised action-topic assignments of action-words are challenging because there is no annotations during the training stage. Besides extracting rich visual features, we well consider the relations between action-topics. Different from previous works, our model can capture long-range relations between actions *e.g.*, *put-milk-back-to-fridge* is strongly related to *fetch-milk-from-fridge* with *pouring* and *drinking* between them. We model all pairwise co-occurrence and temporal casual relations between occurring action-topics in the video, using a new probabilistic model (introduced in Section 5.4). Specifically, we use a joint distribution as the correlated topic priors. They estimate which actions are most likely to co-occur in a video. And we use a relative time distributions of topics to capture the temporal causal relations. They estimate the possible temporal ordering of the occurring actions in the video.

4.4 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 [138] 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.

Figure 4.3: Examples of the human skeletons (red line) and the extracted interactive objects (green mask, left: fridge, right: book).

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 [36] 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 [119] 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. 4.3).

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 [103] 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 [134]. We concatenate the object features of each frame as the interacting-object-trajectory feature of the clip.

4.5 Learning Model

In order to incorporate the aforementioned properties of activities for patching, we present a new generative model (see the graphic model in Fig. 5.4-right and the notations in Fig. 4.5 and Table 5.1). The novelty of our model is the ability to infer the probability of forgotten actions in a complex activity video.

Consider a collection of D videos (documents in the topic model). Each video consists of N_d action-words $\{w_{nd}\}_{n=1}^{N_d}$ mapped to the action-dictionary. Consider K latent action-topics, z_{nd} is the topic assignment of each word, indicating

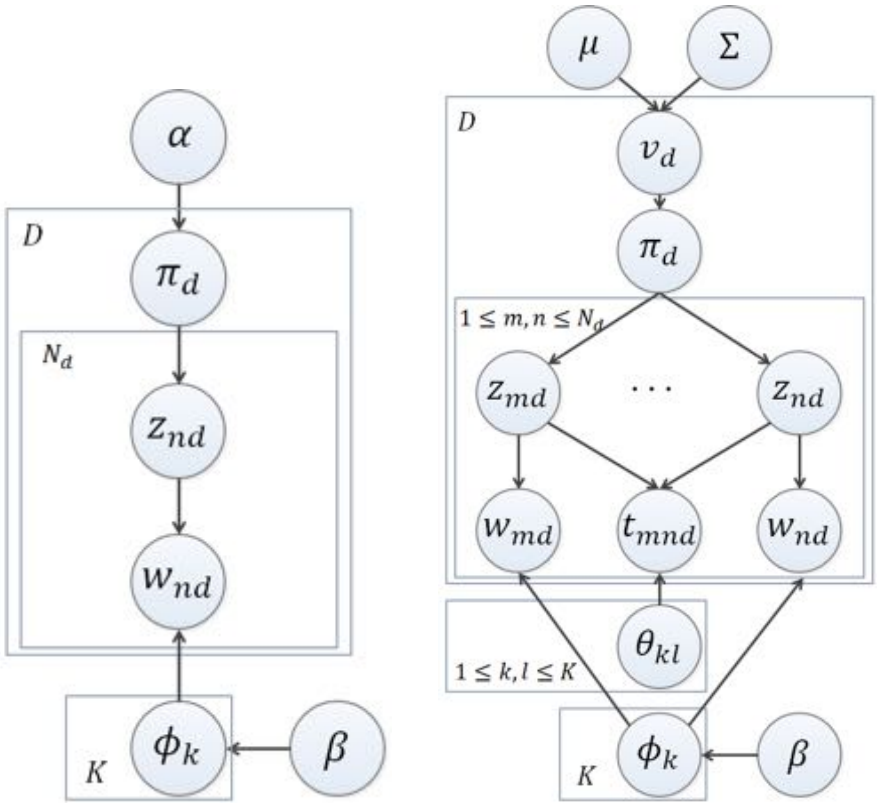


Figure 4.4: The graphic model of LDA (left) and our model (right).

which action-topic the action-word w_{nd} belongs to in the video. Then continuous action-words with the same topic in a video consist an action segment, and the segments assigned with the same topic from different videos consist an action-topic segment cluster.

The topic model such as LDA [20] has been very common for document modeling from language (see graphic model in Fig. 5.4-left), which generates a document using a mixture of topics. To model human actions in the video, our model introduces co-occurrence and temporal structure of topics instead of the topic independency assumption in LDA.

Basic generative process. In a document d, the topic assignment z_{nd} is chosen from a multinomial distribution with parameter $\pi_{:d}$, $z_{dn} \sim Mult(\pi_{:d})$, where $\pi_{:d}$ is sampled from a prior. And the word w_{nd} is generated by a topic-specific

Table 4.1: Notations in our model.

Symbols	Meaning
D	number of videos in the training database;
K	number of action-topics;
N_d	number of words in a video;
w_{nd}	n-th word in d-th document;
z_{nd}	topic-word assignment of w_{nd};
t_{nd}	the normalized timestamp of of w_{nd};
t_{mnd}	$= t_{md} - t_{nd}$ the relative time between w_{md} and w_{nd};
$\pi_{:d}$	the probabilities of topics in d-th document;
$v_{:d}$	the priors of $\pi_{:d}$ in d-th document;
ϕ_k	the multinomial distribution of the word from topic k;
μ, Σ	the mutivariate normal distribution of $v_{:d}$;
θ_{kl}	the relative time distribution of t_{mnd}, between topic k, l;

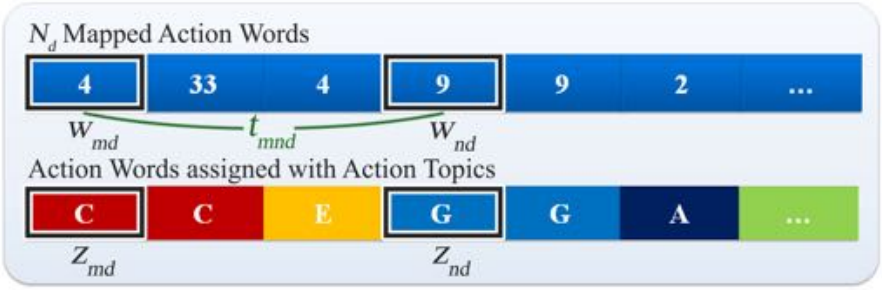


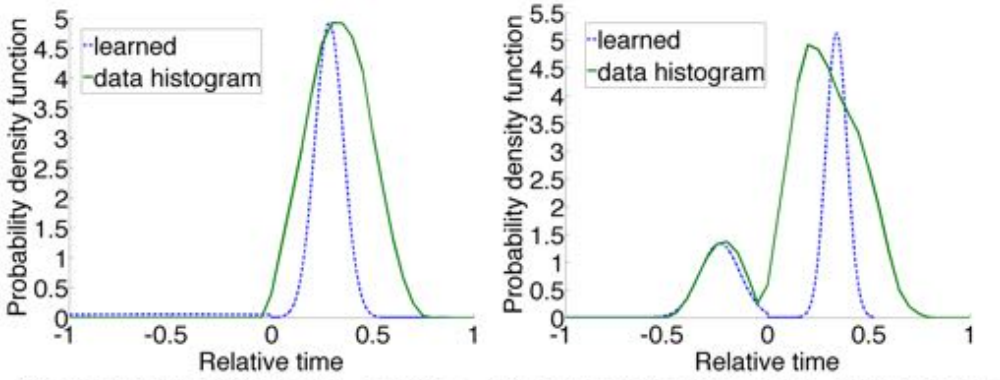
Figure 4.5: Notations in a video.

multinomial distribution $\phi_{z_{nd}}$, $w_{dn} \sim Mult(\phi_{z_{dn}})$, where $\phi_k \sim Dir(\beta)$ is the word distribution of topic k, sampled from a Dirichlet prior with the hyperparameter β.

Topic correlations. First we consider correlations between topics to model the probabilities of co-occurrence of actions. Let π_{kd} be the probability of topic k occurring in document d, where $\sum_{k=1}^{K} \pi_{kd} = 1$. Instead of sampling it from a fix Dirichlet prior with parameter α in LDA, we construct the probabilities by a stick-breaking process:

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

where $0 < \Psi(v_{kd}) < 1$ is a classic logistic function, which satisfies $\Psi(-v_{kd}) = 1 - \Psi(v_{kd})$, and v_{kd} serves as the prior of π_{kd}. The vector $v_{:d}$ in a video are drawn from a mutivariate normal distribution $N(\mu, \Sigma)$, which captures the correlations



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

Figure 4.6: 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).

between topics. In practice, $v_{:d} = [v_{1d}, \cdots, v_{K-1,d}]$ is a truncated vector for $K-1$ topics, then we can set $\pi_{Kd} = 1 - \sum_{k=1}^{K-1} \pi_{kd} = \prod_{k=1}^{K-1} \Psi(-v_{kd})$ as the probability of the final topic for a valid distribution of $\pi_{:d}$.

Relative time distributions. Second 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 word relative to n-th word (the green line in Fig. 4.5). Then t_{mnd} is drawn from a certain distribution, $t_{mnd} \sim \Omega(\theta_{z_{md}, z_{nd}})$, where $\theta_{z_{md}, z_{nd}}$ are the parameters. $\Omega(\theta_{k,l})$ are K^2 pairwise topic-specific relative time distributions defined as follows:

$$\Omega(t|\theta_{k,l}) = \begin{cases} b_{k,l} \cdot N(t|\theta^+_{k,l}) & \text{if } t \geq 0, \\ 1 - b_{k,l} \cdot N(t|\theta^-_{k,l}) & \text{if } t < 0, \end{cases} \tag{4.1}$$

An illustration of the learned relative time distributions are shown in Fig. 4.6. We can see that the distributions we learned can correctly reflect the order of the

actions, *e.g.*, *put-back-to-fridge* is after *pouring* and can be before/after *microwaving*, and the shape is mostly 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[1]. Then the order and the length of the actions will be captured by all these pairwise relative time distributions.

4.6 Gibbs Sampling for 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 [19, 66]. Given a video, the word w_{nd} and the relative time t_{mnd} are observed. In the training stage, given a set of training videos, we use Gibbs sampling to approximately sample other hidden variables from the posterior distribution of our model. Since we adopt conjugate prior $Dir(\beta)$ for the multinomial distributions Φ_k, we can easily integrate out Φ_k and need not to sample them. For simplicity and efficiency, we estimate the standard distributions including the mutivariate normal distribution $N(\mu, \Sigma)$ and the time distribution $\Omega(\theta_{kl})$ by the method of moments, once per iteration of Gibbs sampling. And as in many applications using the topic model, we use fixed symmetric Dirichlet distributions by setting $\beta = 0.01$.

In the Gibbs sampling updates, then we need to sample the topic assignment z_{nd} and the topic prior $v_{:d}$. We can do a collapsed sampling as in LDA by

[1]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. (4.1) 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 summits to one through the domain of t.

calculating the posterior distribution of z_{nd}:

$$
\begin{aligned}
p(z_{nd} = k|\pi_{:d}, z_{-nd}, t_{nd}) &\propto \pi_{kd}\omega(k, w_{nd})p(t_{nd}|z_{:d}, \theta), \\
\omega(k, w_{nd}) &= \frac{N_{kw}^{-nd} + \beta}{N_{k}^{-nd} + N\beta}, \\
p(t_{nd}|z_{:d}, \theta) &= \prod_{m} \Omega(t_{mnd}|\theta_{z_{md},k})\Omega(t_{nmd}|\theta_{k,z_{md}}),
\end{aligned}
\tag{4.2}
$$

where N is the number of unique word types in dictionary, N_{kw}^{-nd} denotes the number of instances of word w_{nd} assigned with topic k, excluding n-th word in d-th document, and N_{k}^{-nd} denotes the number of total words assigned with topic k. z_{-nd} denotes the topic assignments for all words except z_{nd}.

Note that, in Eq. (5.1), π_{kd} is the topic prior generated by a joint distribution giving which actions are more likely to co-occur in the video. $\omega(k, w_{nd})$ is the word distribution for topic k giving which topic the word is more likely from. And $p(t_{nd}|z_{:d}, \theta)$ is the time distribution giving which topic-assignment of the word is more causally consistent to other topic-assignments.

Due to the logistic stick-breaking transformation, the posterior distribution of $v_{:d}$ does not have a closed form. So we instead use a Metropolis-Hastings independence sampler [44]. 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

$$
\begin{aligned}
\mathbb{A}(v_{:d}^*, v_{:d}) &= \frac{p(v_{:d}^*|\mu, \Sigma)\prod_{n=1}^{M_d} p(z_{nd}|v_{:d}^*)q(v_{:d}|v_{:d}^*, \mu, \Sigma)}{p(v_{:d}|\mu, \Sigma)\prod_{n=1}^{M_d} p(z_{nd}|v_{:d})q(v_{:d}^*|v_{:d}, \mu, \Sigma)} \\
&= \prod_{n=1}^{M_d} \frac{p(z_{nd}|v_{:d}^*)}{p(z_{nd}|v_{:d})} = \prod_{k=1}^{K} (\frac{\pi_{kd}^*}{\pi_{kd}})^{\sum_{n=1}^{N_d} \delta(z_{nd},k)},
\end{aligned}
$$

which can be easily calculated by counting the number of words assigned with each topic by z_{nd}. Here the function $\delta(x, y) = 1$ if only if $x = y$, otherwise equal to 0. The time complexity of the sampling per iteration is $O(N_d^2 KD)$.

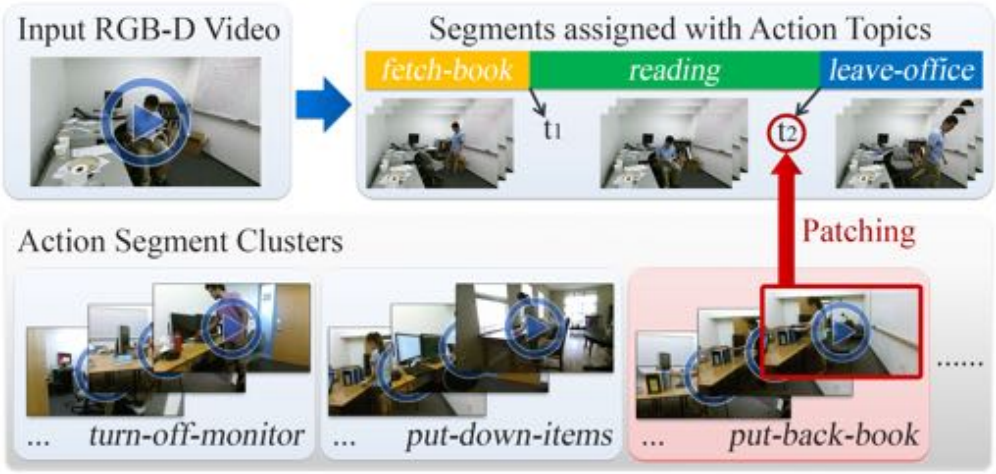


Figure 4.7: **Illustration of action patching using our model.** Given a test video, we infer the forgotten topic from all missing topics in each segmentation point (t_1, t_2 as above) using the learned co-occurrence and temporal relations of the topics. Then we select the top segment from the inferred action-topic's segment cluster by ranking them using a frame-wise similarity score.

Given a test video, we fix all parameters learned in the training stage and only sample the topic assignments z_{nd} and the topic priors $v_{:d}$.

4.7 Applications

4.7.1 Action Segmentation and Recognition

After we learn the topic-assignment of each action-word, we can easily get the action segments by merging the continuous clips with the same assigned topic. Also the assigned topic of the segment indicate which action it is and these segments with the same assigned topic consist an action-topic segment cluster.

4.7.2 Action Patching

We also apply our model in a new significant application, called *action patching*. It reminds people of forgotten actions by output a segment containing the forgotten action from the training set (illustrated in Fig. 5.5). It is more challenging than conventional similarity search, since the retrieved target is not shown in the query video. Therefore, learning the action co-occurrence and the temporal relations is important in this application.

Different from existing models on action relations learning, our model learns all the pairwise relations rather than only the local and the past-to-future transitions. This is very useful to patching, since those actions occurred with a relatively large time interval with or actions occurred after the forgotten actions are also helpful to detect it, *e.g.*, a *put-back-book* might be forgotten as previously seen a *fetch-book* action before a long *reading*, and seen a *leaving* action indicates he really forgot to *put-back-book*.

Our model infers the forgotten action using the probability inference based on the dependencies. After assigning the topics to the action-words of a query video q, we consider adding one additional action-word $\hat{w}$ into the video in each segmentation point t_s. Then the probabilities of the missing topics k_m in each segmentation point can be compared following the posterior distribution in Eq. (5.1):

$$p(z_{\hat{w}} = k_m, t_{\hat{w}} = t_s | other) \propto \pi_{k_m d} p(t_s | z_{:d}, \theta) \sum_{w} \omega(k_m, w),$$

$$s.t. \quad t_s \in T_s, \; k_m \in [1:K] - K_e,$$

where T_s is the set of segmentation points (t_1, t_2 in Fig. 5.5) and K_e is the set of existing topics in the video (*fetch-book*,*etc*. in Fig. 5.5). Thus $[1:K] - K_e$ are the

missing topics in the video (*turn-off-monitor*,*etc*. in Fig. 5.5). $p(t_s|z_{:d}, \theta), \omega(k_m, w)$ can be computed as in Eq. (5.1). Here we marginized $\hat{w}$ to avoid the effect of a specific action-word. Note that, π_{kd} gives the probability of a missing 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 $p(t_s|z_{:d}, \theta)$ measures the casual consistency of adding a new topic.

Then we rank the pair (k_m, t_s) using the above score and select the top ones (three in the experiments). The segments with the selected topics k_m in the training set consist a candidate patching segment set. Finally, we select the top one from the candidates to output by comparing their frame-wise distances. In detail, we consider that the front and the tail of the patching 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 patching segment f_{pm} should be different to f_{qt}, f_{qf}, as it is a different action forgotten in the video.[2] So we choose the patching segment with the maximum score: $ave(\mathbb{D}(f_{pm}, f_{qf}), \mathbb{D}(f_{pm}, f_{qt})) - max(\mathbb{D}(f_{pf}, f_{qt}), \mathbb{D}(f_{pt}, f_{qf}))$, where $\mathbb{D}(,)$ is the average pairwise distances between frames, $ave(,), max(,)$ are the average and max value. 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.

[2]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.

4.8 Experiments

4.8.1 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. 4.10). The new Kinect v2 has higher 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. We ask 7 subjects to perform human daily activities in 8 offices and 5 kitchens with complex backgrounds. And 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. We also record the audio, though it is not used in this work.

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 such as *fetch-book* and *put-back-book* while some occur in random order such as *put-back-to-fridge* and *microwave*. Moreover, to evaluate the action patching performance, 222 videos in the dataset has action forgotten by people and the forgotten actions are annotated. We give the examples of action classes in Fig. 4.10.

4.8.2 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). We compare seven unsupervised approaches in our experiments. They are Hidden Markov Model (HMM), topic model LDA (TM), correlated topic model (CTM), topic model over absolute time (TM-AT), correlated topic model over absolute time (CTM-AT), topic model over relative time (TM-RT) and our causal topic model (CaTM), that is the correlated topic model over relative time. All these methods use the same human skeleton and RGB-D features introduced in Section 4.4. We also evaluate HMM and our model CaTM using the popular features for action recognition, dense trajectories feature (DTF) [128], extracted only in RGB videos[3], named as HMM-DTF and CaTM-DTF.

In the experiments, we set the number of topics (states of HMM) equal to or more than ground-truth action 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. (4.1)). The clip length of the action-words is 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-

[3]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.

AT, TM-RT, CTM-AT and our CaTM. Then we use the same ranking score as in Section 5.5 to select the top one patched segments.

4.8.3 Evaluation Metrics

We want to evaluate if the unsupervised learned action-topics (states for HMM) are semantically meaningful. We first map the assigned topics to the ground-truth labels for evaluation. This could be done by counting the mapped frames between topics and ground-truth classes. 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 can 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.

We then measure the performance in two ways. Per frame: we compute *frame-wise accuracy (Frame-Acc)*, the ratio of correctly labeled frames. Segmentation: we consider 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 [101]. Then we compute *segmentation accuracy (Seg-Acc)*, the ratio of the ground-truth segments that are correctly detected, and *segmentation average pre-*

cision (Seg-AP) by sorting all action segments output by the approach using the average probability of their words' topic assignments. All above three metrics are computed by taking the average of each action class.

We also evaluate the *patching accuracy (P-Acc)* by the portion of correct patched video, including correctly output the forgotten action segments or correctly 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.

4.8.4 Results

Table 5.2 and Fig. 5.6 show the main results of our experiments. We first perform evaluation in the offline setting to see if actions can be well segmented and clustered in the train set. We then perform testing in an online setting 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 can see that our approach performs better than the state-of-the-art in unsupervised action segmentation and recognition, as well as action patching. We discuss our results in the light of the following questions.

Did modeling the long-range relations help? We studied whether modeling the correlations and the temporal relations between topics was useful. The approaches considering the temporal relations, HMM, TM-RT, and our CaTM, outperform other approaches which assume actions are temporal independent. This demonstrates that understanding temporal structure is critical to recognizing and patching actions. The approaches, TM-RT and CaTM, which model

Table 4.2: Results using the same number of topics as the ground-truth action classes. HMM-DTF, CaTM-DTF use DTF RGB features and others use our human skeleton and RGB-D features.

'office'	Seg-Acc		Seg-AP		Frame-Acc		P-Acc
(%)	Offline	Online	Offline	Online	Offline	Online	
HMM-DTF	15.2	9.4	21.4	20.7	20.2	15.9	23.6
HMM	18.0	14.0	25.9	24.8	24.7	21.3	33.3
TM	9.3	9.2	20.9	19.6	20.3	13.0	13.3
CTM	10.0	5.9	18.1	15.8	20.2	16.4	13.3
TM-AT	8.9	3.7	25.4	19.0	18.6	13.8	12.0
CTM-AT	9.6	6.8	25.3	19.8	19.6	15.5	10.8
TM-RT	**30.8**	30.9	29.0	30.2	38.1	36.4	39.5
CaTM-DTF	28.2	27.0	28.3	27.4	37.4	34.0	33.7
CaTM	30.6	**32.9**	**33.1**	**34.6**	**39.9**	**38.5**	**41.5**
'kitchen'	Seg-Acc		Seg-AP		Frame-Acc		P-Acc
(%)	Offline	Online	Offline	Online	Offline	Online	
HMM-DTF	4.9	3.6	18.8	5.6	12.3	9.8	2.3
HMM	20.3	15.2	20.7	13.8	21.0	18.3	7.4
TM	7.9	4.7	21.5	14.7	20.9	11.5	9.6
CTM	10.5	9.2	20.5	14.9	18.9	15.7	6.4
TM-AT	8.0	4.8	21.5	21.6	20.9	14.0	7.4
CTM-AT	9.7	10.0	19.1	22.6	20.1	16.7	10.7
TM-RT	32.3	26.9	23.4	23.0	35.0	31.2	18.3
CaTM-DTF	26.9	23.6	18.4	17.4	33.3	29.9	16.5
CaTM	**33.2**	**29.0**	**26.4**	**25.5**	**37.5**	**34.0**	**20.5**

both the short-range and the long-range relations perform better than HMM only modeling local relations. Also, the approaches considering the topic correlations CTM, CTM-AT, and our CaTM perform better than the corresponding non-correlated topic models TM, TM-AT, and TM-RT. Our CaTM, which considers both the action correlation priors and the temporal relations, shows the best performance.

How successful was our unsupervised approach in learning meaningful action-topics? From Table 5.2, we can see that the unsupervised learned action-topics can be semantically meaningful even though ground-truth semantic labels are not provided in the training. In order to qualitatively estimate the performance, we give a visualization of our learned topics in Fig. 4.9. It shows that the actions with the same semantic meaning are clustered together though they

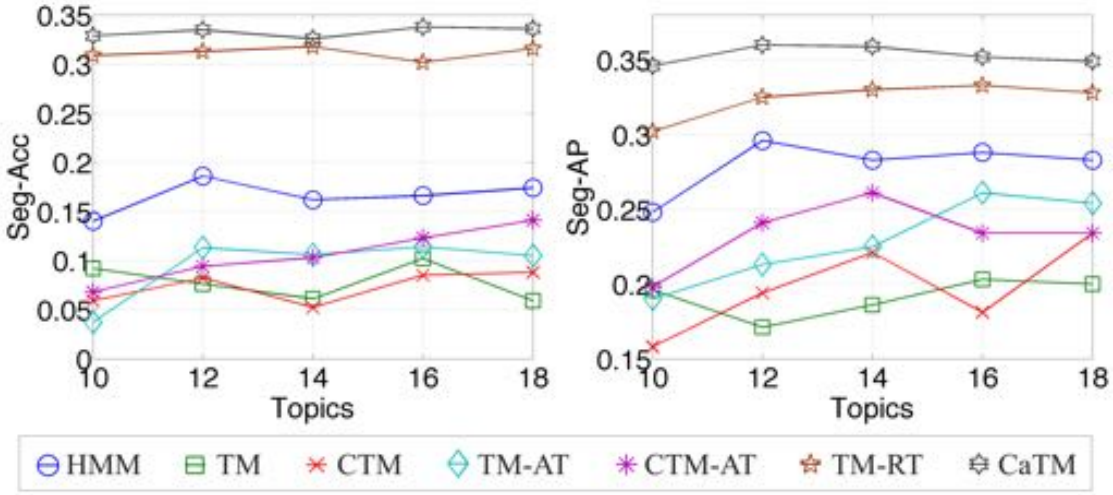


Figure 4.8: Online segmentation Acc/AP varied with the number of topics in 'office' dataset.

are in different views and motions. 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. 5.6. 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. In order to compare the effect of using information from RGB-D videos, we also evaluate our model CaTM and HMM using the popular RGB features for action recognition (CaTM-DTF and HMM-DTF in Table 5.2). Clearly, the proposed human skeleton and RGB-D features outperform the DTF features as more accurate human motion and object are extracted.

How well did our new application of action patching performs? From Table 5.2, we find that the approaches learning the action relations mostly give better patching performance. This is because the learned co-occurrence and temporal structure strongly help indicate which actions are forgotten. Our model

Figure 4.9: **Visualization of the learned topics using our model.** For better illustration, we decompose the segments with the same topic into different modes (shown two) and divide a segment into three stages in time. The clips from different segments in the same stage are merged by scaling to the similar size of human skeletons.

capturing both the short-range and long-range action relations shows the best results.

Figure 4.10: **Examples of every action class in our dataset.** The left is RGB frame and the right is depth frame with human skeleton (yellow).

4.9 Summary

In this chapter, we presented an algorithm that models the human activities in a completely unsupervised setting. We showed that it is important to model the long-range relations between the actions. To achieve this, we considered the

video as a sequence of action-words, and an activity as a set of action-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 recognition, as well as the action patching. For evaluation, we also contributed a new challenging RGB-D activity video dataset.

CHAPTER 5

UNSUPERVISED LEARNING FOR REMINDING HUMANS OF FORGOTTEN ACTIONS

5.1 Introduction

In robot perception, 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. It turns out that the average adult forgets three key facts, chores or events every day [3]. For example in Fig. 5.1, 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 chapter, we focus on detecting these forgotten actions in the complex human activities for a robot, which learns from a completely unlabeled set of RGB-D videos.

There are a large number of works on vision-based human activity recognition for robots. These works infer the semantic label of the overall activity or localize actions in the complex activity for better human-robot interactions [84, 4, 29], assistive robotics [58, 140]. Given the input RGB/RGB-D videos [120, 69, 25], or 3D human joint motions [89, 102], or from other inertial/location sensors [26, 93], they train the perception model using fully or weekly labeled actions [69, 22, 54], or locations of annotated human/their interactive objects [122, 97]. Recently, there are some other works on anticipating human activities for reactive robotic response [70, 58]. However, to enable a robot to remind people of forgotten things, it is challenging to directly use these approaches especially in a completely unsupervised setting.

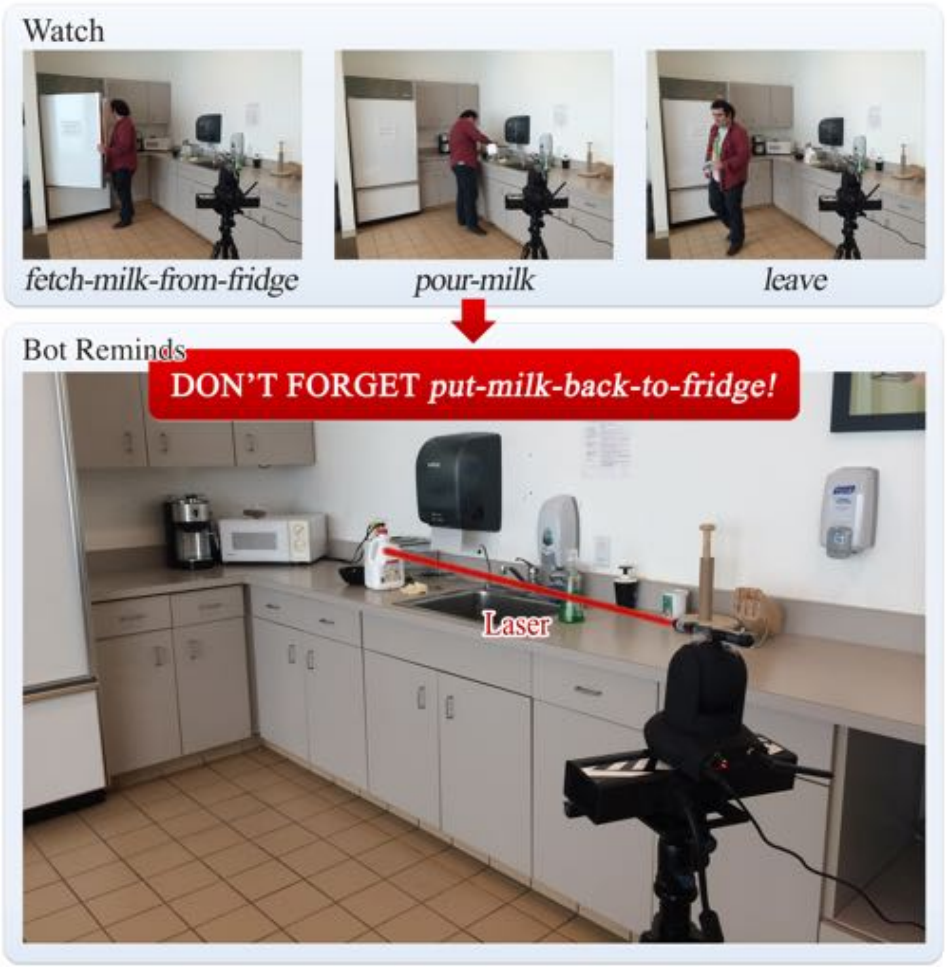
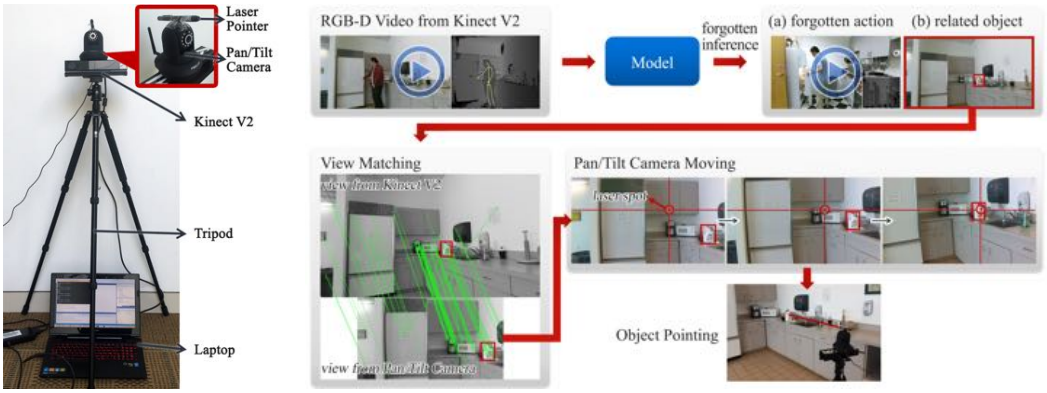


Figure 5.1: Our Watch-Bot watches what a human is currently doing, and uses our unsupervised learning model to detect the human's forgotten actions. Once a forgotten action detected (*put-milk-back-to-fridge* in the example), it points out the related object (*milk* in the example) by the laser spot in the current scene.

Our goal is to 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 work) mounted with a laser pointer, and a laptop (see Fig. 5.2(a)). This setup can be easily deployed on any assistive robot. Taking the example in Fig. 5.1, 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, it would first detect the forgotten action and the related object (the milk), given the input RGB-D frames



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

Figure 5.2: (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 pan/tilt until the laser spot lies in the bounding box of the target object.

and human skeletons from the Kinect; then map the object from the Kinect's view to the camera's view; finally pan/tilt the camera until its mounted laser pointer pointing to the milk.

In real robotic applications, people perform a very wide variety of actions. These are hard to learn from existing videos on the Internet and there are few with annotations of actions or objects. So we propose a probabilistic learning model in a completely unsupervised setting, which can learn actions and relations directly from the data without any annotations, only given the input RGB-D frames with tracked skeletons from Kinect v2 sensor.

We model an activity video as a sequence of actions, so that we can understand which actions have been taken, *e.g.*, the example activity contains four actions: *fetch-milk-from-fridge*, *pour*, *put-milk-back-to-fridge*, and *leave*.[1] For detect-

[1]In the training, we do not know these action semantic labels. Instead we assign the action cluster index.

ing the forgotten action and reminding, we model the co-occurrence between actions and the interactive objects, as well as the temporal relations between these segmented actions, *e.g.*, action *fetch-milk-from-fridge* often co-occurs with and is temporally after action *put-milk-back-to-fridge*, and object *milk* occurs in both actions. Using the learned actions and relations, we infer the forgotten actions and localize the related objects, *e.g.*, *put-milk-back-to-fridge* might be forgotten as previously seen *fetch-milk-from-fridge* before *pouring*, and seen *leaving* indicates he really forgot to do, also *milk* is the object interacted in the forgotten action.

We evaluate our approach extensively on a large RGB-D human activity dataset recorded by Kinect v2 [135]. The dataset contains 458 videos of human daily activities as compositions of multiple actions interacted with different objects, in which people forgot actions in 222 videos. We show that our approach not only improves the action segmentation and action cluster assignment performance, but also obtains promising results of forgotten action detection. Moreover, we show that our Watch-Bot is able to remind humans of forgotten actions in the real-world robotic experiments.

5.2 Related Work

Most previous works focus on recognizing human actions for both robotics [84, 69, 25] and computer vision [65, 121, 5]. They model different types of information, such as the temporal relations between actions [101, 129], the human and the interactive object appearances and relations [71, 129]. Yang *et al*. [140] presented a system that learns manipulation action plans for robot from uncon-

strained youtube videos. Hu *et al*. [54] proposed an activity recognition system trained from soft labeled data for the assistant robot. Chrungoo *et al*. [29] introduced a human-like stylized gestures for better human-robot interaction. Piyathilaka *et al*. [102] used 3D skeleton features and trained dynamic bayesian networks for domestic service robots. However, it is challenging to directly use these approaches for inferring the forgotten actions.

Recently, there are works on anticipating human activities and they performed well for assistant robots [70, 58]. They modeled the object affordances and object/human trajectories to discriminate different actions in past activities and anticipate future actions. However, in order to detect forgotten actions, we also need to consider actions after it such as *boiling water* indicates *filling kettle* before it.

The output laser spot on object is also related to the work 'a clickable world' [96], which selects the appropriate behavior to execute for an assistive object-fetching robot using the 3D location of the click by the laser pointer. Differently, we keep the laser pointer fixed on top of the camera, and pan/tilt the camera iteratively to point out the target object using a real-time view matching.

Most of these works rely on supervised learning given fully labeled actions, or weakly supervised action labels, or locations of human/their interactive objects. Differently, our robot uses a completely unsupervised learning setting that trains model only on Kinect's output RGB-D videos. Our model is based on our previous work [135], which presents a Casual Topic Model to model action relations in the complex activity. In this work, we further introduce the human interactive object and its relations to actions, so that the robot can localize the related object. We then design a robotic system using the model to kindly re-

mind people.

5.3 Watch-Bot System

We outline our Watch-Bot system in this section (see Fig. 5.2). Our goal is to detect what people forgot to do given the observation of his poses and interacted objects. The robot consists of a Kinect v2 sensor, a pan/tilt camera mounted with a laser pointer, and a laptop. The input to our system is RGB-D human activity videos with the tracked 3D joints of human skeletons from Kinect v2. Then we use an unsupervised trained learning model (see Section 5.4) to infer the forgotten action and localize the related object in the Kinect's view. After that, we map the object bounding box from the Kinect's view to the camera's view. Finally, we pan/tilt the camera until the laser spot lies within the target object in its view (see Section 5.5).

Video Representation. To detect the action structure in the complex activity video, we propose a video representation that draws parallels to document modeling in the natural language [20] (illustrated in Fig. 5.3). We first decompose a video into a sequence of overlapping fixed-length temporal clips. We then extract the human-skeleton-trajectory features and the interactive-object-trajectory features from the clips. In order to build a compact representation of the activity video, we represent it as a sequence of words. We use *k*-means to cluster the human-skeleton-trajectories/interactive-object-trajectories from all the clips to form a *human-dictionary* and an *object-dictionary*, where we use the cluster centers as *human-words* and *object-words*. Then, the video can be represented as a sequence of human-word and object-word indices by map-

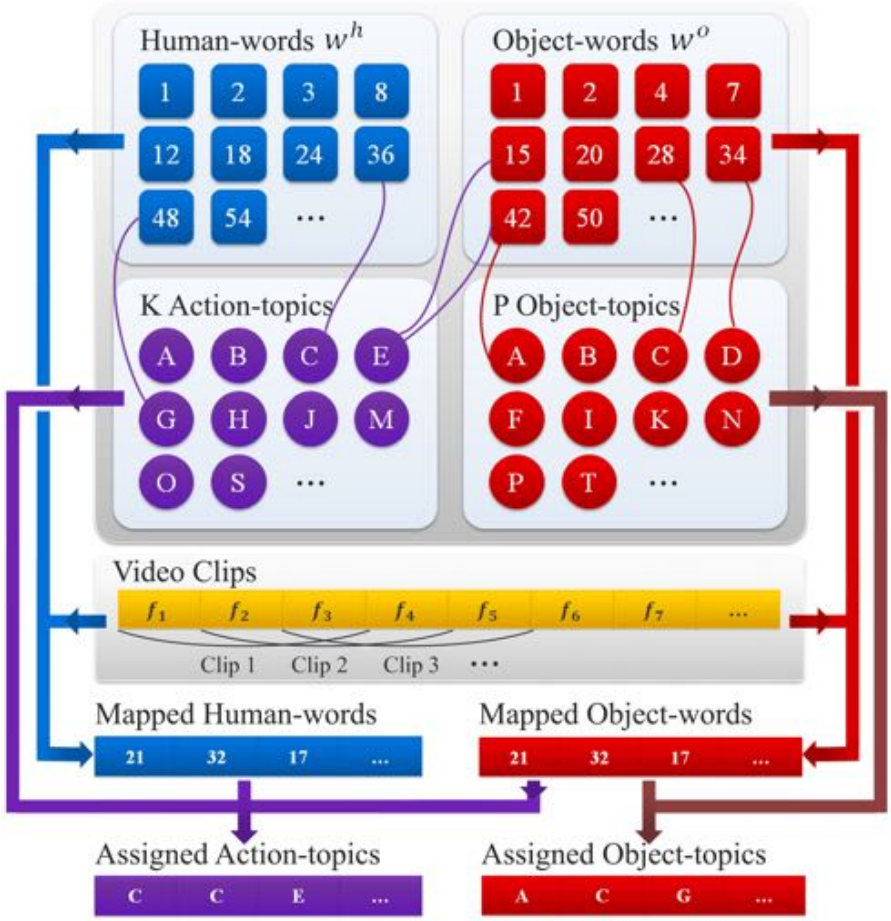


Figure 5.3: Video representation in our approach. A video is first decomposed into a sequence of overlapping fixed-length temporal clips. The human-skeleton-trajectories/interactive-object-trajectories from all the clips are clustered to form the human-dictionary/object-dictionary. Then the video is 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. Also, an activity video is about a set of action-topics/object-topics indicating which actions are present and which object types are interacted.

ping its human-skeleton-trajectories/interactive-object-trajectories to the nearest human-words/object-words in the dictionary. Also, an activity video is 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 interacted. We use the visual features as described in Section 4.4.

5.4 Learning Model

We present a new unsupervised model for our Watch-Bot. The graphic model is illustrated in Fig. 5.4 and the notations are in Table 5.1. Our model is able to infer the probability of forgotten actions using the rich relationships between actions and objects.

We learn the model from a training set of D unlabeled videos. 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_{nd}^h mapped to the human-dictionary and an object-word w_{nd}^o 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_{nd}^o from P latent object-topics, indicating which object-topic is interacted within the clip. The assignments are denoted as $z_{nd}^{(1)}$ and $z_{nd}^{(2)}$. We use superscripts $(1), (2)$ to denote action-topics and object-topics respectively. After assignments, in a video, continuous clips with the same action-topic compose an action segment. All the segments assigned with the same action-topic from the training set compose an action cluster.

As shown in Fig. 5.4, the generative process of our model is as follows. 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

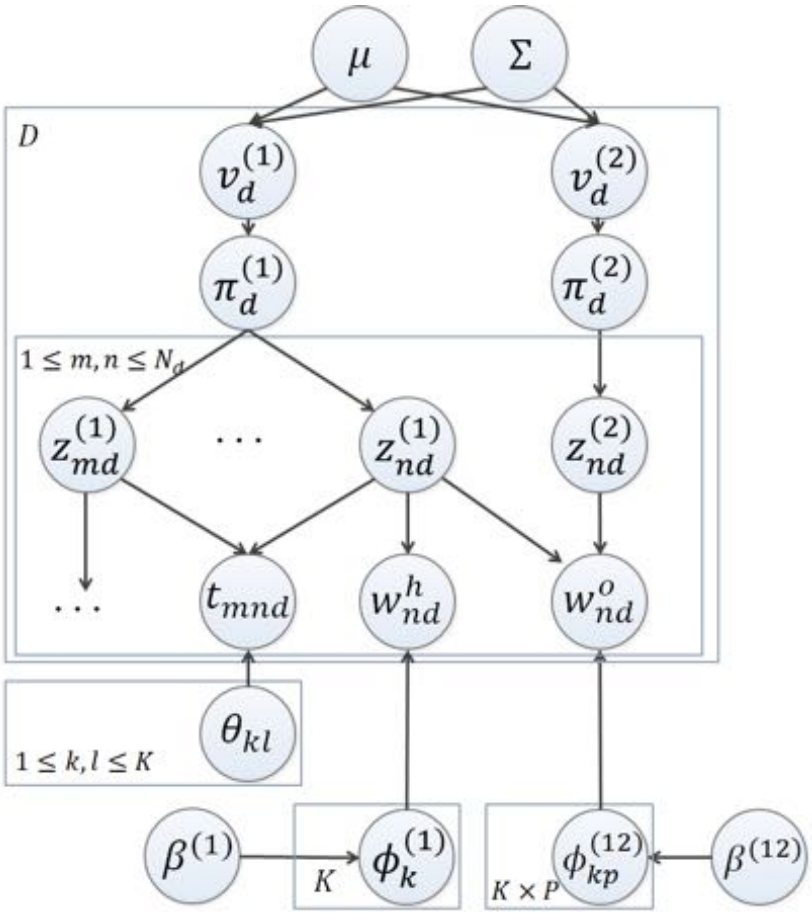


Figure 5.4: The probabilistic graphic model of our approach.

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 *reading*. So we consider the object-word distribution for different combinations of the action topic and the object topic.

The co-occurrence such as action *put-down-items* and action *take-items*, object *book* and action *reading*, is useful to recognizing the co-occurring actions/objects and gives a strong evidence for detecting forgotten actions. We model the co-occurrence by drawing their priors from a mixture distribution. In the graphic model, $\pi_{kd}^{(1)}, \pi_{pd}^{(2)}$ decide the probability of action-topic k and object-topic p occurring in a document d, where $\sum_{k=1}^{K} \pi_{kd}^{(1)} = 1, \sum_{p=1}^{P} \pi_{pd}^{(2)} = 1$. We construct the prob-

Table 5.1: 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^h_{nd}	n-th human-word in d-th video;
w^o_{nd}	n-th object-word in d-th video;
$z^{(1)}_{nd}$	action-topic assignment of c_{nd};
$z^{(2)}_{nd}$	object-topic assignment of w^o_{nd};
t_{nd}	normalized timestamp of of c_{nd};
t_{mnd}	$= t_{md} - t_{nd}$ the relative time between c_{md} and c_{nd};
$\pi^{(1)}_{:d}, \pi^{(2)}_{:d}$	the probabilities of action/object-topics in d-th document;
$v^{(1)}_{:d}, v^{(2)}_{:d}$	the priors of $\pi^{(1)}_{:d}, \pi^{(2)}_{:d}$ in d-th document;
$\phi^{(1)}_k$	multinomial human-word distribution from action-topic k;
$\phi^{(12)}_{kp}$	multinomial object-word distribution from action-topic k and object-topic p;
μ, Σ	multivariate normal distribution of $v_{:d} = [v^{(1)}_{:d}, v^{(2)}_{:d}]$;
θ_{kl}	relative time distribution of t_{mnd}, between action-topic k, l;

abilities using a stick-breaking process as in [135], where $v^{(1)}_{kd}, v^{(2)}_{pd}$ serve as the priors. Then we draw the packed vector $v_{:d} = [v^{(1)}_{:d}, v^{(2)}_{:d}]$ from a multivariate normal distribution $N(\mu, \Sigma)$, which captures the correlations between action-topics and object-topics.

The temporal relations between actions are also useful to discriminating the actions using temporal ordering and inferring the temporal consistent forgotten actions. So we model the relative time of occurring actions as in [135]. In detail, let $t_{nd}, t_{md} \in (0, 1)$ be the absolute time stamp of n-th clip and m-th clip, 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, $t_{mnd} \sim \Omega(\theta_{z^{(1)}_{md}, z^{(1)}_{nd}})$, where $\theta_{z^{(1)}_{md}, z^{(1)}_{nd}}$ are the parameters. $\Omega(\theta_{k,l})$ are K^2 pairwise action-topic specific relative time distributions defined by a product of a Bernoulli distribution which gives the probability of action k after/before the action l, and a normal distribution which estimates how long the action k is after/before the action l.

5.4.1 Learning and Inference

We use Gibbs sampling [19, 66] to learn the parameters and the infer the hidden variables from the posterior distribution of our model. The word w^h_{nd}, w^o_{nd} and the relative time t_{mnd} are observed in each video. 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. The topic priors $v^{(1)}_{:d}, v^{(2)}_{:d}$ can be sampled by a Metropolis-Hastings independence sampler [44] as in [135]. Following the convention, 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 Latent Dirichlet Allocation (LDA) [20] by calculating the posterior distribution of $z^{(1)}_{nd}, z^{(2)}_{nd}$:

$$
\begin{aligned}
&p(z^{(1)}_{nd} = k | \pi^{(1)}_{:d}, z^{(1)}_{-nd}, z^{(2)}_{nd}, t_{nd}) \\
&\propto \pi^{(1)}_{kd} \omega(k, w^h_{nd}) \omega(k, z^{(2)}_{nd}, w^o_{nd}) p(t_{nd} | z^{(1)}_{:d}, \theta), \\
&p(z^{(2)}_{nd} = p | \pi^{(2)}_{:d}, z^{(2)}_{-nd}, z^{(1)}_{nd}) \propto \pi^{(2)}_{pd} \omega(z^{(1)}_{nd}, p, w^o_{nd}), \\
&\omega(k, w^h_{nd}) = \frac{N^{-nd}_{kw^h} + \beta^{(1)}}{N^{-nd}_{k} + N_w \beta^{(1)}}, \\
&\omega(k, p, w^o_{nd}) = \frac{N^{-nd}_{kpw^o} + \beta^{(12)}}{N^{-nd}_{kp} + N_o \beta^{(12)}}, \\
&p(t_{nd} | z^{(1)}_{:d}, \theta) = \prod_{m}^{N_d} \Omega(t_{mnd} | \theta_{z^{(1)}_{md}, k}) \Omega(t_{nmd} | \theta_{k, z^{(1)}_{md}}),
\end{aligned}
\tag{5.1}
$$

where N_w, N_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,

Algorithm 3 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 5.4.1.
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, each not assigned action-topic $k_m \in [1:K] - K_e$, and each object-topic $p_m \in [1:P]$:
Compute the probability defined in Eq. 5.2;
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 $forget\_score(p)$;
7. If $forget\_score(p)$ is smaller than a threshold, claim no action forgotten and return;
8. Segment the current frame to super-pixels using edge detection [36] as in Section 5.3;
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.

and N_k^{-nd}/N_{kp}^{-nd} denotes the number of total words assigned with action-topic k/action-topic k and object-topic p. $z_{-nd}^{(1)}/z_{-nd}^{(2)}$ denotes the topic assignments for all words except $z_{nd}^{(1)}/z_{nd}^{(2)}$.

In Eq. (5.1), note that the topic assignments are decided by which actions/objects are more likely to co-occur in the video (the occurrence probabilities $\pi_{kd}^{(1)}/\pi_{kd}^{(2)}$), the visual appearance of the word (the word distributions $\omega(k, w_{nd}^h), \omega(k, p, w_{nd}^o)$) and the temporal relations (the relative time distributions $p(t_{nd}|z_{:d}^{(1)}, \theta)$). The time complexity of the sampling per iteration is $O(N_d D(\max(N_d K, P))$.

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

5.5 Forgotten Action Detection and Reminding

In this section, we describe how we apply our model in our robot to detecting the forgotten actions and reminding people. It is more challenging than conventional action recognition, since what to infer is not shown in the query video. Therefore, unlike the existing models on action relations learning, our model learns rich relations rather than the only local temporal transitions. As a result, those actions occurred with a relatively large time interval, occurred after the forgotten actions, as well as the interactive objects can also be used to detect forgotten actions, *e.g.*, a *put-back-book* might be forgotten as previously seen a *fetch-book* action before a long *reading*, and seen a *book* and a *leaving* action indicates he really forgot it.

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. 3). We first use our model to segment the query video into action segments (step 1,2 in Alg. 3), and then infer the most possible forgotten action-topic and the related object-topic (step 4 in Alg. 3). 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. 3). 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. 3). After that, we map the bounding box of the object from the Kinect's view to the camera's view. Finally, we pan/tilt camera until its laser pointer points out the related object in the current scene.

Forgotten Action and Object Inference. We first introduce how we infer the forgotten action-topic and object-topic using the dependencies in our learned

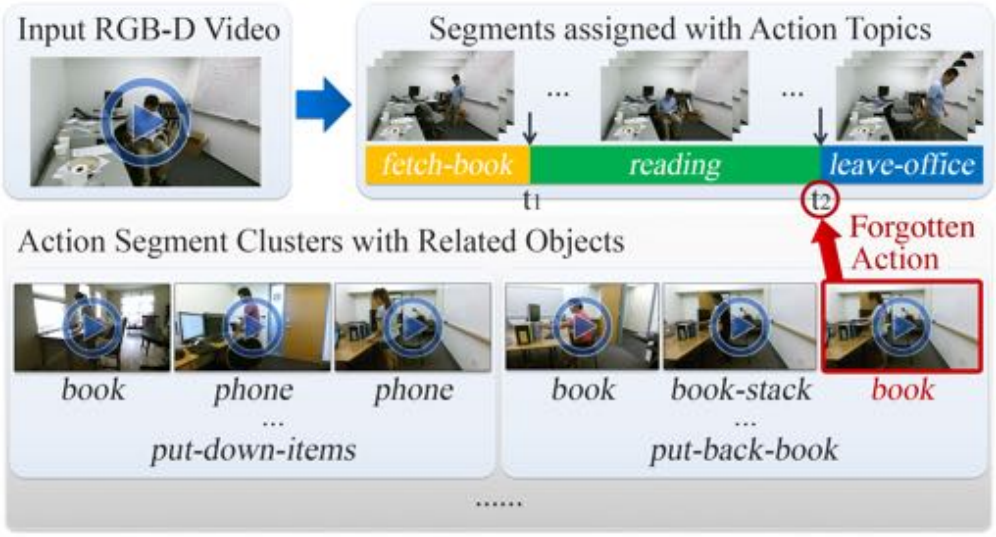


Figure 5.5: **Illustration of forgotten action and object detection using our model.** Given a query video, we infer the forgotten action-topic and object-topic in each segmentation point (t_1, t_2). Then we select the top segment from the inferred action-topic's segment cluster with the inferred object-topic with the maximum $forget\_score$.

model. After assigning the action-topics and object-topics to the query video q, we consider adding one additional clip $\hat{c}$ consisting of $\hat{w^h}, \hat{w^o}$ into q in every action segmentation point t_s (see Fig 5.5). Then the probabilities of the missing action-topics k_m with object-topics p_m in each segmentation point t_s can be computed following the posterior distribution in Eq. (5.1):

$$
\begin{aligned}
& p(z^{(1)}_{\hat{c}} = k_m, z^{(2)}_{\hat{c}} = p_m, t_{\hat{c}} = t_s | other) \\
& \propto \pi^{(1)}_{k_m d} \pi^{(2)}_{p_m d} p(t_s | z^{(1)}_{:d}, \theta) \sum_{w^h, w^o} \omega(k_m, w^h) \omega(k_m, p_m, w^o), \\
& s.t. \quad t_s \in T_s, \; k_m \in [1:K] - K_e,
\end{aligned} \tag{5.2}
$$

where T_s is the set of segmentation points (such as t_1, t_2 in Fig. 5.5) and K_e is the set of existing action-topics in the video (*fetch-book, etc.* in Fig. 5.5). Thus $[1:K] - K_e$ are the missing topics in the video (*put-down-items, etc.* in Fig. 5.5). $p(t_s | z^{(1)}_{:d}, \theta), \omega(k_m, w^h), \omega(k_m, p_m, w^o)$ can be computed as in Eq. (5.1). Here we marginalized $\hat{w^h}, \hat{w^o}$ to avoid the effect of a specific human-word or object-word.

Note that, in Eq. (5.2), the closer topics would have higher probabilities $\pi^{(1)}_{kd}, \pi^{(2)}_{pd}$ to co-occur in this query video as they are drawn from the learned joint distribution. The action-topics which are more consistent with the learned temporal relations would have higher probability $p(t_s|z^{(1)}_{:d}, \theta)$. The marginalized 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.

Forgotten Action and Object Detection. We then introduce how we retrieve a top action segment from the training database. We first select the top three tuples (k_m, p_m, t_s) using the above probability. These action segments consist a forgotten action candidate segment set Q. We then retrieve the segment from Q with the maximum $forget\_score(p) = ave(\mathbb{D}(f_{pm}, f_{qf}), \mathbb{D}(f_{pm}, f_{qt})) - max(\mathbb{D}(f_{pf}, f_{qt}), \mathbb{D}(f_{pt}, f_{qf}))$, where $\mathbb{D}(,)$ is the average pairwise distances between frames, $ave(,), max(,)$ are the average and max value. The front and the tail of the forgotten action segment f_{pf}, f_{pt} need to 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}. The middle of the forgotten action segment f_{pm} need to be different to f_{qt}, f_{qf}, as it is a different action forgotten in the video[2]. 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 related forgotten object in the current scene. We segment the current frame into super-pixels as in Section 5.3, then search the nearest super-pixels using the extracted object in the top retrieved action, finally merge the adjacent super-pixels and bound the largest one with a bounding box.

[2]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.

Real Object Pointing. We describe how we pan/tilt the camera to point out the real object. 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 we fix the position of the laser spot in the pan/tilt camera view, 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. 5.2). 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.

5.6 Experiments

5.6.1 Dataset

We evaluate our Watch-Bot in a challenging human activity RGB-D dataset [135] consisting of 458 videos of about 230 minutes in total recorded by the Kinect v2 sensor. Each video in the dataset contains 2-7 actions interacted with different objects (see examples in Fig. 4.10). We asked 7 subjects to perform human daily

activities in 8 offices and 5 kitchens with complex backgrounds and recorded the activities in different views. It is composed of fully annotated 21 types of actions (10 in the office, 11 in the kitchen) interacted with 23 types of objects. The participants finish tasks with different combinations of actions and ordering. Some actions occur together often such as *fill-kettle* and *boil-water*, while some are not always together. Some actions are in a fix order such as *turn-on-monitor* and *turn-off-monitor* while some occur in random order. Also, in the dataset, people forgot actions in 222 videos. There are 3 types of forgotten actions in 'office' and 5 types in 'kitchen'.

5.6.2 Baselines

We compare four unsupervised approaches. They are Hidden Markov Model (HMM) [16], LDA topic model [20], our previous work Causal Topic Model(CaTM) [135] and our Watch-Bot Topic Model (WBTM). We use the same human skeleton and RGB-D features introduced in Section 5.3. In LDA, actions and objects are modeled independently as the priors of action/object assignments are sampled from a fix Dirichlet prior and there is no relative time between actions modeled. For HMM, similarly we set action states which generates both human and object trajectory features of each clip, and object states which generates object trajectory features. Since there is no object modeled in CaTM, we only evaluate its activity related performance.

In the experiments, we set the number of action-topics/object-topics and states for HMM equal to or more than ground-truth action/object classes. For LDA, CaTM and our WBTM, the clip length is set to 20 frames, densely sampled

by step one and the size of human/object dictionary is set to 500. The forgotten action candidate set for different approaches consists of the segments with the inferred missing topics by transition probabilities for HMM, the topic priors for LDA. After inference, we use the same forgotten action and object detection method as introduced in Section 5.5.

5.6.3 Evaluation Metrics

We test in two environments 'office' and 'kitchen'. In each environment, the dataset is split into a train set with mostly 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 mostly forgotten videos (office: 89, kitchen 113). We train the models in the train set and evaluate the following metrics in the test set.

Action Segmentation and Cluster Assignment. As in evaluation for unsupervised clustering, we map the action cluster in the train set to the ground-truth action labels by counting the mapped frames between action-topics and ground-truth action classes as in [135] . Then we can use the mapped action class label for evaluation.

We measure the performance in two ways. Per frame: we compute *frame-wise accuracy (Frame-Acc)*, the ratio of correctly labeled frames. Segmentation: we consider a true positive if the union/intersection of the detected and the ground-truth segments is greater than 40% as in [101]. We compute *segmentation accuracy (Seg-Acc)*, the ratio of the ground-truth segments that are correctly detected and *segmentation average precision (Seg-AP)* by sorting all action segments

Table 5.2: Action segmentation and cluster assignment results, and forgotten action/object detection results.

'office'(%)	Seg-Acc	Seg-AP	Frame-Acc	FA-Acc	FO-Acc
HMM	19.4	23.1	27.3	32.2	20.4
LDA	12.2	19.6	18.4	15.7	10.5
CaTM	32.9	34.6	38.5	41.5	-
WBTM	**35.2**	**36.0**	**41.2**	**46.2**	**36.4**
'kitchen'(%)	Seg-Acc	Seg-AP	Frame-Acc	FA-Acc	FO-Acc
HMM	17.2	18.8	20.3	12.4	5.3
LDA	6.7	17.1	14.4	10.8	5.3
CaTM	29.0	25.5	34.0	20.5	-
WBTM	**30.7**	**28.5**	**36.9**	**24.4**	**20.6**

using the average probability of their words' topic assignments. All above three metrics are computed by taking the average of each action class.

Forgotten Action and Object Detection. We measure the *forgotten action detection accuracy (FA-Acc)* by the portion of correct detected forgotten action or correctly claiming no forgotten actions. We consider the output forgotten action segments by the compared approaches containing over 50% ground-truth forgotten actions as correct. We measure the *forgotten object detection accuracy (FO-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%.

5.6.4 Results

Table 5.2, Fig. 5.6 and Fig. 5.7 show the main results of our experiments. We discuss our results in the light of the following questions.

How well did forgotten action/object detection perform? In Table 5.2, we can see that our model achieves a promising results for complex activities with

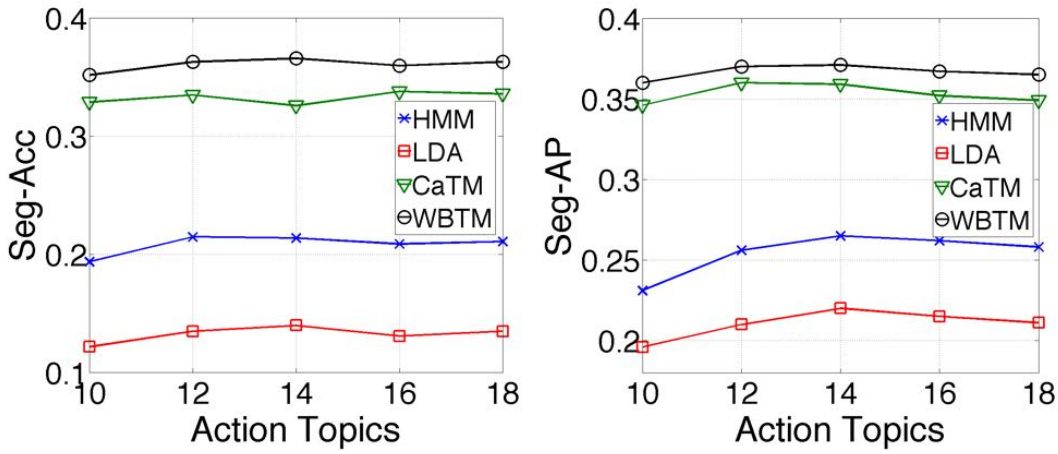


Figure 5.6: Action segmentation Acc/AP varied with the number of action-topics in 'office' dataset.

multiple objects in variant environments in the completely unsupervised setting. Our models CaTM and WBTM show better performance than traditional uncorrelated topic model LDA, since the co-occurrence and temporal structure are well learned. They outperform HMM, since we consider both the short-range and long-range action relations while HMM only considers the local neighboring states transitions. Our WBTM model improves the performance over CaTM on action clustering and forgotten action detection, also is able to detect the forgotten object, because action and object topics are factorized and their relations are well modeled.

How important is it to consider relations between actions and objects? From the results, we can see that the model which did well in forgotten action detection also performed well in detecting forgotten object. Since our model well considers the relations between the action and the object, it shows better performance in both forgotten action and forgotten object detection than HMM and LDA which models action and object independently as well as CaTM which only models the actions.

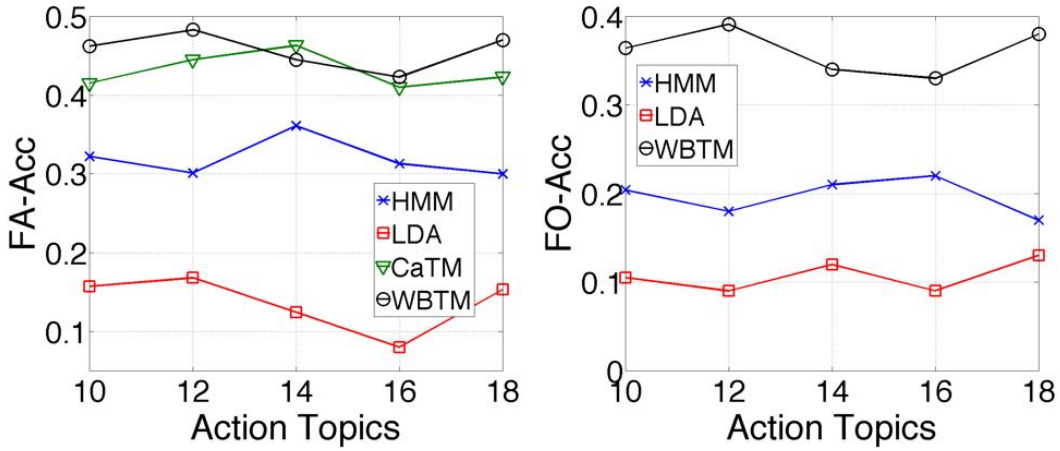


Figure 5.7: Forgotten action/object detection accuracy varied with the number of action-topics in 'office' dataset.

Figure 5.8: An example of the robotic experiment. The robot detects the human left the food in the microwave, then points to the microwave.

How successful was our unsupervised approach in learning meaningful action-topics? From Table 5.2 and Fig 5.6, we can see that the unsupervised learned action-topics can be semantic meaningful even though ground-truth semantic labels are not provided in the training. It can also be seen that, the better action segmentation and cluster assignment performance often indicates better forgotten action detection performance, since actions in the complex activity should be first well segmented and discriminated for next stage forgotten action/object detection.

How did the performance change with the number of action-topics? We plot the performance curves varied with the action-topic number in Fig. 5.6 and Fig. 5.7. It shows that the performance does not change much with the action-topics. This is because a certain action might be divided into several action-

Table 5.3: Robotic experiment results. The higher the better.

	Succ-Rate(%)	Subj-AccScore(1-5)	Subj-HelpScore(1-5)
HMM	37.5	2.1	2.3
LDA	29.2	1.8	2.0
WBTM	**62.5**	**3.5**	**3.9**

topics but more variations are also introduced.

5.6.5 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. 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 5.3 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. 5.8 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.

5.7 Summary

In this chapter, we enabled a Watch-Robot to automatically detect people's forgotten actions. We showed that our robot is easy to setup and our model can be trained with completely unlabeled videos without any annotations. We modeled an activity video as a sequence of action segments, which we can understand as meaningful actions. We modeled the co-occurrence between actions and the interactive objects as well as the temporal relations between these segmented actions. Using the learned relations, we inferred the forgotten actions and localized the related objects. We showed that our approach improved the unsupervised action segmentation and cluster assignment performance, and was able to detect the forgotten action on a complex human activity RGB-D video dataset. We showed that our robot was able to remind people of forgotten actions in the real-world robotic experiments by pointing out the related object using the laser pointer.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we presented how to use unsupervised learning to model the complex spatial, temporal and semantical structures of human environments and human actions, so as to give a better understanding for robots. We showed that modeling these structures in perception algorithms is useful in robotics and how robotic systems use our algorithms to improve the real-world applications.

We first introduced a hierarchical semantic labeling approach to modeling semantic and spacial relations of objects for robotic perception. Using the semantic hierarchy, robot is able to recognize the environments relevant to the tasks. We built a semantic hierarchy graph to represent the 'is part of' and 'is type of' relationships and proposed a novel CRF based approach which relates pixel-wise and pair-wise observations to labels. We encoded hierarchical labeling constraints into the model while keeping inference tractable. We showed the labeling performance improvement of our algorithm in off-line experiments and the high success rate in the real-world robotic scenarios.

We then presented a human centred co-segmentation method to automatically co-segmenting the common semantic regions from a set of images. We generated a set of object proposals as foreground candidates from the images and discovered the rich internal structure of these proposals using a proposed fully connected CRF auto-encoder. We showed that leveraging the interactions between humans and objects improved the co-segmentation accuracy signifi-

cantly.

We described an unsupervised causal topic model, which learns high-level co-occurrence and temporal relations between the actions. We modeled the video as a sequence of short-term action clips, which contains human-words and object-words. So an activity is about a set of action-topics and object-topics indicating which actions are present and which objects are interacted with. We introduced an unsupervised casual topic model relating the words and the topics. In the experiments, we showed the flexibilities in consideration of different structures using our models and modeling the long-term temporal relations and co-occurrence of actions gives the best results. We also contributed a new challenging RGB-D activity video dataset which contains several human daily activities as compositions of multiple actions interacting with different objects.

Finally, we presented a novel robotic system using our unsupervised structured learning based perception algorithms. The robot was able to detect what humans forgot to do by watching their activities, and if necessary reminds the person using a laser pointer to point out the related object. We showed the promising result in the robotic experiments.

6.2 Future Work

The following are different directions to pursue in the future:

6.2.1 Deep Structures in Feature Encoding and Decoding

Deep structures have been proved to improve the learning ability significantly recently. In our work of fully connected CRF auto-encoder, both encoding and decoding part can be extended to deep structures using convolutional neural networks, recurrent neural networks or deep mixture models *etc*. Though the unsupervised structured learning using deep structures is still an open challenging problem, the way we leverage the fully connected CRF encoding in the unsupervised learning is promising to capture the fully structured information in the final encoding stage when introducing the deep structures.

6.2.2 Extending to Semi-Supervised Learning

Semi-supervised learning is able to leverage a small set of labeled data to guide the learning and improve the performance. Our casual topic model and fully connected CRF auto-encoder is not hard to be extended to the semi-supervised setting and have the potentials to use them to improve the performance. For the casual topic model, we can fix the sampling using the ground-truth of topic assignments, so that the relations we learned can be guided to the right direction without getting stuck in the local minimum. For the fully connected CRF auto-encoder, the initialization can be done by supervised learning of the encoding parameters, which might give a better and faster learning of the next unsupervised learning stage of the whole model.

6.2.3 Practical Robotic Applications

Applying perception algorithms in real-world robotic applications are challenging problems. When developing the perception algorithms, we need to think in the perspective of applications just like our task relevant hierarchical semantic labeling. We showed that robot can do a better job on navigation, manipulation, and reminding using a better perception algorithm. Robots will work better for people in cars, houses, offices if using unsupervised learning to improve the perception of robot from large-scale data from different domains in these applications. In the practical applications such as automatic lighting, temperature control, home appliance control at smart homes or navigation/planning in smart car systems, how can the perception algorithms be improved when humans are in the loop? How can we develop a better planning of the robot to improve the perception? More practical questions like these, which are out of the perception algorithm itself but closely related to perception and final applications, need to be answered.

BIBLIOGRAPHY

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

[2] Mixed integer programming solver. http://tomopt.com/tomlab/products/cplex/.

[3] 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.

[4] Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason Corso, and Venkat Krovi. Estimating human dynamics on-the-fly using monocular video for pose estimation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2012.

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

[6] E. A. Akkoyunlu. The enumeration of maximal cliques of large graphs. *SIAM J. Comput.*, 2(1):1–6, 1973.

[7] Dewayne Rocky Aloysius. *Bron-Kerbosch Algorithm*. PopulPublishing, 2012.

[8] Waleed Ammar, Chris Dyer, and Noah A Smith. Conditional random field autoencoders for unsupervised structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[9] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *IJRR*, 32(1):19–34, 2013.

[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[11] Shayan Modiri Assari, Amir Roshan Zamir, and Mubarak Shah. Video classification using semantic concept co-occurrences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] Alper Aydemir, Kristoffer Sjöö, John Folkesson, Andrzej Pronobis, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *ICRA*, 2011.

[13] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Scene segmentation using temporal clustering for accessing and re-using broadcast video. In *Multimedia and Expo (ICME), IEEE International Conference on*, 2015.

[14] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[15] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *International Journal of Computer Vision (IJCV)*, 93(3):273–292, 2011.

[16] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[17] Subhabrata Bhattacharya, Mahdi M. Kalayeh, Rahul Sukthankar, and Mubarak Shah. Recognition of complex events: Exploiting temporal dy-

namics between underlying concepts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

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

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

[20] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[21] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[22] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision (ECCV)*, 2014.

[23] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.

[24] GabrielJ. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*. 2008.

[25] Guang Chen, Manuel Giuliani, Daniel S. Clarke, Andre K. Gaschler, and Alois Knoll. Action recognition using ensemble weighted multi-instance learning. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[26] Liming Chen, J. Hoey, C.D. Nugent, D.J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):790–808, 2012.

[27] Stephen Xi Chen, Arpit Jain, Abhinav Gupta, and Larry S Davis. Piecing together the segmentation jigsaw using context. In *CVPR*, 2011.

[28] Wei-Chen Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *CVPR*, 2013.

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

[30] Vincent Delaitre, Josef Sivic, and Ivan Laptev. Learning person-object interactions for action recognition in still images. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

[31] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[32] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[33] Jia Deng, Jonathan Krause, Alex Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*, 2012.

[34] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010.

[35] D.Koller and N.Friedman. Probabilistic graphical models: Principles and techniques. In *MITPress*. 2009.

[36] Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *International Conference on Computer Vision (ICCV)*, 2013.

[37] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *European Conference on Computer Vision (ECCV)*, 2009.

[38] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res. (JMLR)*, 12:2121–2159, 2011.

[39] Clemens Eppner and Oliver Brock. Grasping unknown objects by exploiting shape adaptability and environmental constraints. In *IROS*, 2013.

[40] V. Escorcia and J.C. Niebles. Spatio-temporal human-object interactions for action recognition in videos. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 2013.

[41] Tanveer A Faruquie, Prem K Kalra, and Subhashis Banerjee. Time based activity inference using latent dirichlet allocation. In *British Machine Vision Conference (BMVC)*, 2009.

[42] Huazhu Fu, Dong Xu, Stephen Lin, and Jiang Liu. Object-based rgbd image co-segmentation with mutex constraint. In *CVPR*, 2015.

[43] Huazhu Fu, Dong Xu, Bao Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, 2014.

[44] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[45] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[46] Karl Granström, Jonas Callmer, Fabio T. Ramos, and Juan I. Nieto. Learning to detect loop closure from range data. In *ICRA*, 2009.

[47] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013.

[48] Hu He and Ben Upcroft. Nonparametric semantic segmentation for 3d street scenes. In *IROS*, 2013.

[49] E. Herbst, P. Henry, and D. Fox. Toward online 3-d object segmentation and mapping. In *ICRA*, 2014.

[50] Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D flow: Dense 3-d motion estimation using color and depth. In *ICRA*, 2013.

[51] Lauren Hinkle and Edwin. Predicting object functionality using physical simulations. In *IROS*, 2013.

[52] Minh Hoai, Zhen zhong Lan, and Fernando De la Torre. Joint segmentation and classification of human actions in video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[53] D.S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[54] Ninghang Hu, Zhongyu Lou, Gwenn Englebienne, and Ben Kröse. Learning to recognize human activities from soft labeled data. In *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[55] Zhaoyin Jia, Andrew Gallagher, Ashutosh Saxena, and Tsuhan Chen. 3d-based reasoning with blocks, support, and stability. In *CVPR*, 2013.

[56] Yun Jiang, Hema Koppula, and Ashutosh Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *CVPR*, 2013.

[57] Yun Jiang and Ashutosh Saxena. Infinite latent conditional random fields for modeling environments through humans. In *Robotics: Science and Systems (RSS)*, 2013.

[58] Yun Jiang and Ashutosh Saxena. Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs. In *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[59] Simon Jones and Ling Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[60] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[61] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[62] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J. Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *IJRR*, 31(2):216–235, 2012.

[63] Vadim Kantorov and Ivan Laptev. Efficient feature extraction, encoding and classification for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[64] Dov Katz and Oliver Brock. Interactive segmentation of articulated objects in 3d. In *Workshop on Mobile Manipulation at ICRA*, 2011.

[65] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *European Conference on Computer Vision (ECCV)*, 2007.

[66] Dae I. Kim and Erik B. Sudderth. The doubly correlated nonparametric topic model. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[67] Gunhee Kim, Eric P. Xing, Li Fei-Fei, and Takeo Kanade. Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion. In *ICCV*, 2011.

[68] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman. Human focused action localization in video. In *International Workshop on Sign, Gesture, and Activity (SGA) in Conjunction with ECCV*, 2010.

[69] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from RGB-D videos. *I. J. Robotic Res.*, 32(8):951–970, 2013.

[70] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[71] Hema Swetha Koppula and Ashutosh Saxena. Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation. In *International Conference on Machine Learning (ICML)*, 2013.

[72] H.S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.

[73] Dimitrios Kottas and Stergios Roumeliotis. Exploiting urban scenes for vision-aided inertial navigation. In *RSS*, 2013.

[74] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

[75] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[76] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. In *ICRA*, 2014.

[77] I. Laptev and P. Perez. Retrieving actions in movies. In *International Conference on Computer Vision (ICCV)*, 2007.

[78] Maria Teresa Lazaro, Lina María Paz, Pedro Pinies, José A. Castellanos, and G. Grisetti. Multi-robot SLAM using condensed measurements. In *IROS*, 2013.

[79] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. In *RSS*, 2013.

[80] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[81] Yen-Yu Lin, Ju-Hsuan Hua, Nick C. Tang, Min-Hung Chen, and Hong-Yuan Mark Liao. Depth and skeleton associated action recognition without online accessible rgb-d cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[82] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[83] Malte Lorbach, Sebastian Höfer, and Oliver Brock. Prior-assisted propagation of spatial information for object search. In *IROS*, 2014.

[84] 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.

[85] A. Lucchi, P. Marquez-Neila, C. Becker, Y. Li, K. Smith, G. Knott, and P. Fua. Learning structured models for segmentation of 2-d and 3-d imagery. *Medical Imaging, IEEE Transactions on*, 34(5):1096–1110, 2015.

[86] Marianna Madry, Carl Henrik Ek, Renaud Detry, Kaiyu Hang, and Danica Kragic. Improving generalization for 3d object categorization with global structure histograms. In *IROS*, 2012.

[87] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007.

[88] Marco Manfredi, Costantino Grana, and Rita Cucchiara. Learning superpixel relations for supervised image segmentation. In *International Conference on Image Processing (ICIP)*, 2014.

[89] A. Mansur, Y. Makihara, and Y. Yagi. Action recognition using dynamics features. In *International Conference on Robotics and Automation (ICRA)*, 2011.

[90] S. Mathe and C. Sminchisescu. Actions in the Eye: Dynamic Gaze Datasets and Learnt Saliency Models for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2014.

[91] Fanman Meng, Hongliang Li, Guanghui Liu, and King Ngi Ngan. Object co-segmentation based on shortest path algorithm and saliency model. *Multimedia, IEEE Transactions on (TMM)*, 14(5):1429–1441, 2012.

[92] Michael Milford. Vision-based place recognition: how low can you go? *IJRR*, 32(7):766–789, 2013.

[93] Jun-Ki Min and Sung-Bae Cho. Activity recognition based on wearable sensors using selection/fusion hybrid ensemble. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1319–1324, 2011.

[94] Lopamudra Mukherjee, Vikas Singh, Jia Xu, and MaxwellD. Collins. Analyzing the subspace structure of related images: Concurrent segmentation of image sets. In *ECCV*. 2012.

[95] J. Neira, A.J. Davison, and J.J. Leonard. Guest editorial special issue on visual SLAM. *Robotics, IEEE Transactions on*, 24(5):929–931, 2008.

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

[97] Bingbing Ni, Vignesh R. Paramathayalan, and Pierre Moulin. Multiple granularity analysis for fine-grained action detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[98] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision (ECCV)*, 2010.

[99] Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. From large scale image categorization to entry-level categories. In *ICCV*, 2013.

[100] Dejan Pangercic, Moritz Tenorth, Benjamin Pitzer, and Michael Beetz. Semantic object maps for robotic housework - representation, acquisition and use. In *IROS*, 2012.

[101] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

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

[103] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[104] Xiaofeng Ren, Liefeng Bo, and D. Fox. RGB-D scene labeling: Features and algorithms. In *CVPR*, 2012.

[105] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006.

[106] Michael Ruhnke, Liefeng Bo, Dieter Fox, and Wolfram Burgard. Compact RGB-D surface models based on sparse coding. In *AAAI*, 2013.

[107] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[108] Bryan C. Russell and Antonio Torralba. Building a database of 3d scenes from user annotations. In *CVPR*, 2009.

[109] S. Sadanand and **J. J. Corso**. Action bank: A high-level representation of activity in video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[110] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *CVPR*, 2013.

[111] A. Saxena, S.H. Chung, and A. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[112] A. Saxena, J. Driemeyer, and A.Y. Ng. Robotic grasping of novel objects using vision. *IJRR*, 27(2):157, 2008.

[113] A. Saxena, M. Sun, and A.Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31(5):824–840, 2009.

[114] Bernt Schiele. A database for fine grained activity detection of cooking activities. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[115] Ozan Sener, Amir Roshan Zamir, Chenxia Wu, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic action discovery from video collections. *CoRR*, 2016.

[116] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *RSS*, 2013.

[117] Qinfeng Shi, Li Cheng, Li Wang, and Alex Smola. Human action segmentation and recognition using discriminative semi-markov models. *International Journal of Computer Vision (IJCV)*, 93(1):22–32, 2011.

[118] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGB-D images. In *ECCV*, 2012.

[119] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[120] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *International Conference on Robotics and Automation (ICRA)*, 2012.

[121] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[122] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[123] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.

[124] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[125] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[126] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *ECCV*. 2010.

[127] Nam N. Vo and Aaron F. Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[128] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[129] Xiaoyang Wang and Qiang Ji. A hierarchical context model for event recognition in surveillance video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[130] Xuerui Wang and Andrew 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.

[131] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. Modeling 4d human-object interactions for event and object recognition. In *ICCV*, 2013.

[132] T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, and J.B. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *ICRA*, 2013.

[133] C. Wu, J. Zhang, O. Sener, B. Selman, S. Savarese, and A. Saxena. Watch-n-Patch: Unsupervised Learning of Actions and Relations. *ArXiv e-prints*, 2016.

[134] Chenxia Wu, Ian Lenz, and Ashutosh Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. In *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[135] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. Watch-n-patch: Unsupervised understanding of actions and relations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[136] Chenxia Wu, Jiemi Zhang, Ashutosh Saxena, and Silvio Savarese. Human centred object co-segmentation. In *Tech Report*, 2016.

[137] Chenxia Wu, Jiemi Zhang, Bart Selman, Silvio Savarese, and Ashutosh Saxena. Watch-bot: Unsupervised learning for reminding humans of forgotten actions. In *International Conference on Robotics and Automation (ICRA)*, 2016.

[138] Di Wu and Ling Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[139] Yang Yang, Imran Saleemi, and Mubarak Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1635–1648, 2013.

[140] Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In *AAAI*, 2015.

[141] Bangpeng Yao and Li 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. (TPAMI)*, 34(9):1691–1703, 2012.