LEARNING DESCRIPTIVE MODELS OF OBJECTS AND ACTIVITIES FROM EGOCENTRIC VIDEO

A Thesis
Presented to
The Academic Faculty

by

Alireza Fathi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
August 2013

LEARNING DESCRIPTIVE MODELS OF OBJECTS AND ACTIVITIES FROM EGOCENTRIC VIDEO

Approved by:

Professor James M. Rehg, Adviser
School of Interactive Computing
Georgia Institute of Technology

Professor Gregory D. Abowd
School of Interactive Computing
Georgia Institute of Technology

Professor Aaron Bobick
School of Interactive Computing
Georgia Institute of Technology

Professor Thad Starner
School of Interactive Computing
Georgia Institute of Technology

Professor Martial Hebert
Robotics Institute
Carnegie Mellon University

Professor Antonio Torralba
CSAIL
Massachusetts Institute of Technology

Date Approved: 6 June 2013

ACKNOWLEDGEMENTS

Many thanks to my mother Sussan. I owe you all of this. Also thanks to my father Mohammad and my sister Shaghayegh.

Thanks to all of my advisers and mentors from whom I learned a lot: Jim Rehg, Jessica Hodgins, John Krumm, Frank Dellaert, Greg Mori and Tamara Smyth.

I want to thank all my thesis committee members: Gregory Abowd, Martial Hebert, Aaron Bobick, Antonio Torralba and Thad Starner.

I want to also thank David Forsyth, Fei-Fei Li, Deva Ramanan, Xiaofeng Ren, Nassir Navab, Bill Triggs and Stan Sclaroff who helped me at different stages of my education with their letters and support.

Contents

List of Tables

List of Figures

SUMMARY

Recent advances in camera technology have made it possible to build a comfortable, wearable system which can capture the scene in front of the user throughout the day. Products based on this technology, such as GoPro and Google Glass, have generated substantial interest. In this thesis, I present my work on egocentric vision, which leverages wearable camera technology and provides a new line of attack on classical computer vision problems such as object categorization and activity recognition.

The dominant paradigm for object and activity recognition over the last decade has been based on using the web. In this paradigm, in order to learn a model for an object category like coffee jar, various images of that object type are fetched from the web (e.g. through Google image search), features are extracted and then classifiers are learned. This paradigm has led to great advances in the field and has produced state-of-the-art results for object recognition. However, it has two main shortcomings: a) objects on the web appear in isolation and they miss the context of daily usage; and b) web data does not represent what we see every day.

In this thesis, I demonstrate that egocentric vision can address these limitations as an alternative paradigm. I will demonstrate that contextual cues and the actions of a user can be exploited in an egocentric vision system to learn models of objects under very weak supervision. In addition, I will show that measurements of a subject's gaze during object manipulation tasks can provide novel feature representations to support activity recognition. Moving beyond surface-level categorization, I will showcase a method for automatically discovering object state changes during actions, and an approach to building descriptive models of social interactions between groups of individuals. These new capabilities for egocentric video analysis will enable new applications in life logging, elder care, human-robot interaction, developmental screening, augmented reality and social media.

Chapter I

INTRODUCTION

Egocentric vision is an area which aims at studying the new capabilities that wearable camera technology enables in the field of computer vision (See the first[1] and the second[2] workshops on egocentric vision, and the paper by Kanade and Hebert [86]). The recent advances in camera technology provide the possibility of continuously capturing video from what a person sees throughout everyday life. This new way of sampling the visual world, as an alternative to the currently dominant web-based paradigm, creates the opportunity to attack the fundamental problems in computer vision from a novel perspective.

This thesis aims at developing object and activity recognition techniques that leverage the capabilities that are enabled by egocentric vision. Some of these capabilities are as follows:

- Egocentric vision provides a *continuous* and *rich* paradigm for sampling the visual world in contrast to the discrete web-based approach.
- In egocentric vision, first-person *attentional cues* can be leveraged to discover objects and recognize activities.
- Egocentric video corresponds to what a particular person sees. As a result, the algorithms and techniques can be *personalized* to the characteristics and preferences of that person.

In Sections 1.1, 1.2 and 1.3 we explain each of the above bullet points in detail. In Section 1.4, we point out a few other characteristics of the egocentric vision paradigm. Then in Section 1.5, we go over some of the challenges of recognition in egocentric videos. In Section **??**, we describe some of the potential applications of recognizing objects and activities in egocentric video. Finally, in Section 1.6, we lay out the content of this thesis.

1.1 A Continuous and Rich Sample of Visual World

The dominant paradigm for object and activity recognition over the last decade has been based on using the web. In the web-based paradigm, in order to learn a model for an object category like "coffee jar", various images of that object type are fetched from the web (e.g. through Google image

[1] http://www.seattle.intel-research.net/egovision09/

[2] http://egovision12.cc.gatech.edu/

search) to build a positive training set. Furthermore, images of other object categories are used as the negative training set. Then features are extracted from these images and finally classifiers are trained for each object type using machine learning techniques. This is the gist of the web-based paradigm. This paradigm has led to great advances in the field and has produced state-of-the-art results for object recognition. However, it has two main shortcomings:

- Objects on the web appear in isolation. The context of daily usage is missing from the web images. In particular it is hard to answer the following questions by using web images: (1) What actions does a particular object afford? (2) How does this object change as a result of an action? (3) What are the objects that often interact with this object during daily activities?

- Web data does not represent what we see every day. For example, look at the images of the coffee jar returned in the first row of Google search shown in Figure 1, and compare them to the coffee jar that is seen by the first-person in Figure 2. Images on the web are often taken from the canonical view of the objects, while in every-day life we see objects from different view points and in different states and situations. In addition, web data doesn't contain a uniform sample of everyday life. For example, there are 21,000,000 videos of "wedding" on YouTube while there are only 280,000 videos for "eating lunch". Around 100 times more videos for wedding. This is while we eat lunch every day, but we go to a wedding once in every few years!

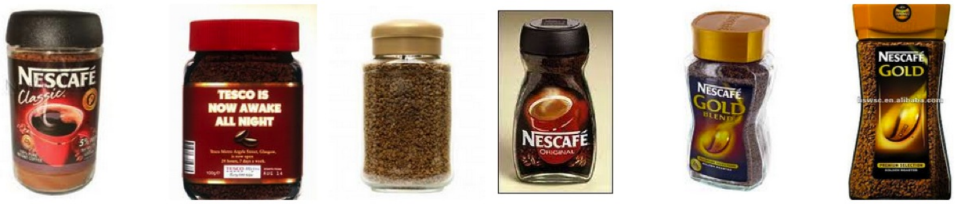


Figure 1: Images of the coffee jar returned in the first row of Google search.

As an alternative paradigm, egocentric vision captures the objects in the context of their usage in daily activities. In daily life, objects change states when they go through actions. For example, a "closed coffee jar" becomes an "open coffee jar" as a result of the action of "opening". These actions and their effects to objects and the scene is captured in egocentric videos. In addition, objects are seen from different viewing angles and in different states. However, on the web, as seen in the images of Figure 1, objects often only appear in one particular state, and from one particular point of view. Furthermore, the web space is discrete, i.e., the images corresponding to the transition from one state to another are missing on the web. As a result, it is a hard task to learn descriptive models of objects and actions from web data, while egocentric data is very rich for this purpose, as is shown in Figure 3.

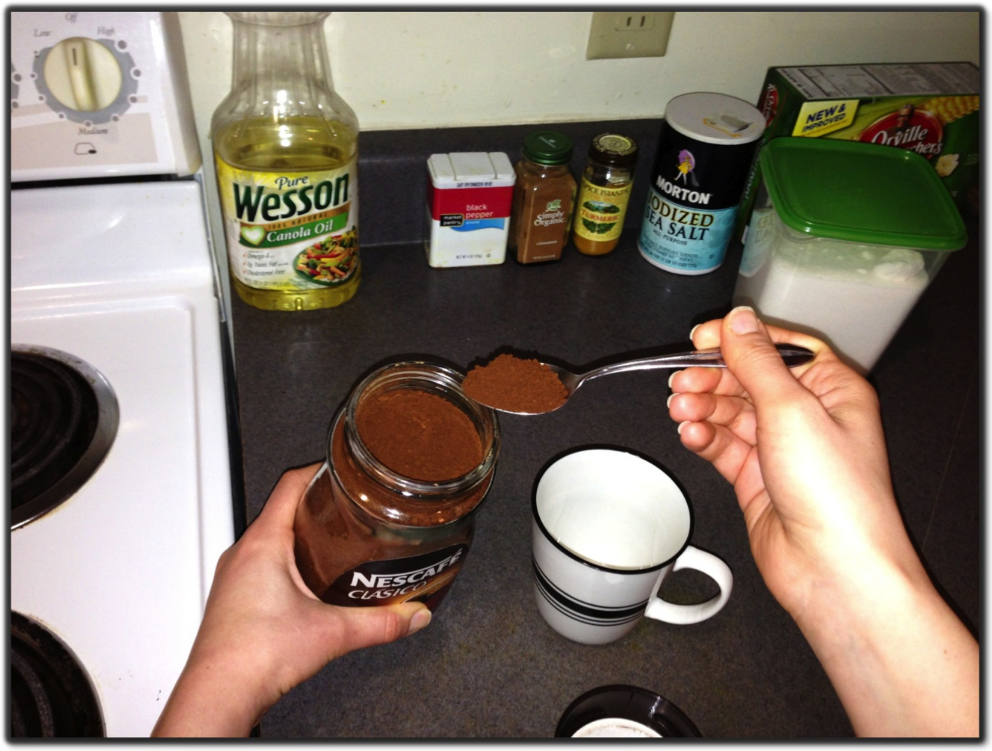


Figure 2: An image of a coffee jar from an egocentric video.

Egocentric video captures the relationships between objects, hands and actions of daily living. Hands manipulate objects and perform the actions. Actions change the state of the objects and change the pose of the hands. Objects and materials interact with each other, change shape and change states. In Chapter 7, we demonstrate that leveraging these relationships improves the recognition accuracy for both objects and actions. For example, “cup” is an object that often occurs in the activity of “making coffee”, but it rarely is used in activity of “making cheese sandwich”. The same way that detecting objects is crucial for recognizing daily activities, activities also provide significant context for recognizing objects.

Furthermore, in Chapter 8, we use the fact that actions change the state of the objects to learn object and material states in egocentric video, and temporally segment the video into action intervals. We discover regions that correspond to object and material states by finding the changes in the environment after an action is performed. Once we learn to detect the state of the objects, we can localize actions that cause the object states to change.

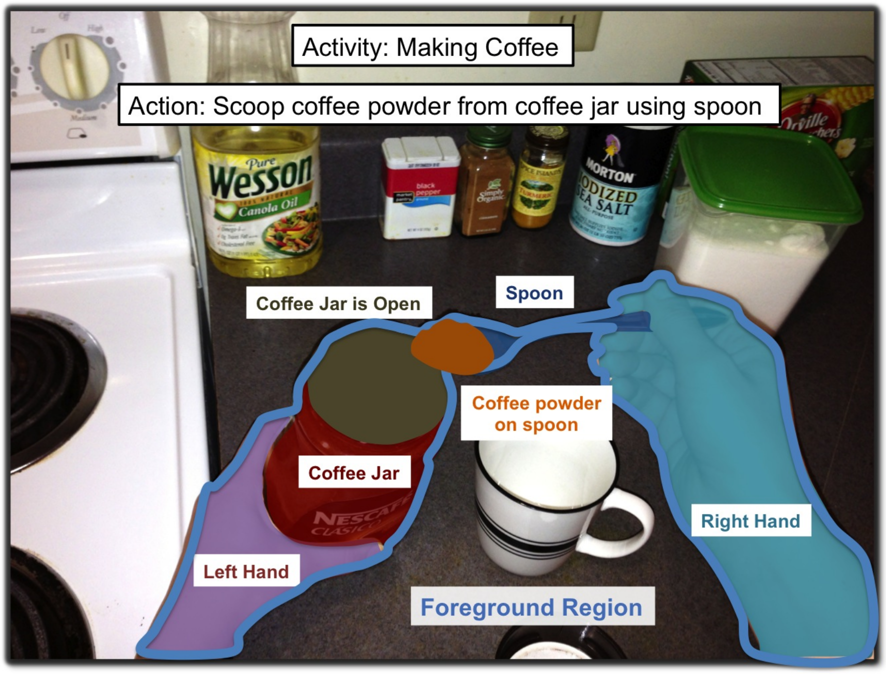


Figure 3: Our goal is to learn descriptive models of objects and activities from egocentric video.

1.2 Attentional Cues

An egocentric vision system can capture attentional cues that are implicitly provided by the first-person at different granularities [169], as depicted in Fig 4. Attentional cues let our algorithms focus on important regions in the video and ignore the locations that are less important.

In order to better understand what we mean by attentional cues, imagine a system without any attentional cue: an abstract camera randomly placed and oriented in space. That camera uniformly samples images from the world and provides no specific information on where the important objects and actions appear in the images. In contrast to this camera, an egocentric camera is not randomly placed and oriented in space, but instead is oriented in the direction that the camera wearer's head is oriented. As a result, the egocentric camera is continuously capturing the first-person's observable space. The content of an egocentric video tells us how the head of the camera-wearer is oriented in space at any given time. The objects that appear in the egocentric video are the ones that are being observed by the first-person and are very likely to be relevant to the activity that is taking place. As a result, a coarse level of attention is provided by the video content which corresponds to the observable space of the first-person.

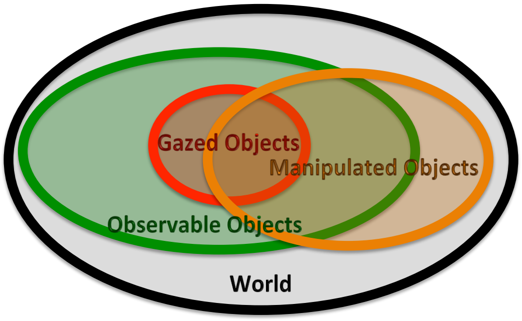


Figure 4: An egocentric vision system can capture human attention at different granularities.

A more fine-grained level of attention is provided by first-person's gaze. An egocentric vision system that is utilized with eye-tracking capability, provides the accurate location that is gazed by the camera wearer (first-person) in the environment. Such a fine-grained attentional cue not only determines the particular object that is attended by the first-person, but also identifies the exact region of that object which is being fixated.

An additional attentional cue for identifying the important regions in the scene is provided by first-person's hands. The objects that are being manipulated by the first-person's hands are often very important for understanding what is happening in the scene. Humans often ignore the objects and actions in the background and only focus on the objects that are being manipulated by their hands [100]. As a result, the regions that are moving by the first-person's hands are often important regions in video.

Throughout this thesis, we will study the value of the attentional cues for recognition of object and activities. We will demonstrate that it is possible to segment the first-person's hands and the region corresponding to active objects in egocentric video. In addition, we will study first-person's gaze and its value for recognizing daily actions in egocentric video. Finally, we will demonstrate the value of attentional cues for recognizing the social behavior of individuals surrounding the first-person in the scene.

1.3 Personalization

A key property of an egocentric system is that it can be personalized to its user in an automated fashion. For example, it is hard to learn models for all the objects in the world. However, each person only uses a small set of objects in her daily life. An egocentric system can adapt to these objects over time by observing first-person's daily activities. There are three main benefits in the personalization of an egocentric vision system: 1) improves recognition of objects and activities; 2) results in a better ranking of interesting moments; and 3) provides the possibility of predicting user's intentions. The key to building a personalized system is leveraging semi-supervised and online learning techniques that can update their learned models over time.

Thus, in Chapter 6 we design an adaptive method that learns daily objects from first-person's actions. We use the fact that handled objects move together with first-person's hands to carve them out of egocentric video and learn models of them. We further develop a weakly supervised learning method that labels the carved objects based on the patterns of object co-occurrence in activities.

Another useful application of personalization is for building a visual memory. For example, if the system knows that the first-person is interested in the moments that involve interaction with family, it can record a diary of those moments.

1.4 Other Characteristics

Egocentric videos of daily activities follow a deterministic structure that can extremely simplify the recognition tasks.

Location Prior: in egocentric video, the activities and important objects often appear in the center of the frames. The left hand often appears on the left side of the frames, and the right hand often appears on the right side of the frames. Similarly, in case of social activities, important faces often appear in the center of the frames. In Chapter 6, we will show that the location of the hands in video is a strong cue for discriminating between different categories of daily actions.

Appearance: in egocentric video, it rarely happens that the important and active objects become occluded. The reason is that the camera is capturing what the first-person is seeing, and the first-person will naturally avoid occlusion during interaction with objects. In addition, since the the camera is on the head, the handled objects are very close to the camera, and as a result they appear quite large in the video. This helps to be able to better recognize the details about these objects.

Time and Place Context: context plays an important role in recognition of egocentric activities. For example, if a person is in the kitchen, she is more likely to prepare a meal rather than to play basketball. Or if a person is in the office, she is more likely to perform reading or writing. Time of the day is another source of contextual information. Often individuals follow a daily routine in their daily activities. For example, the probability of someone going to work in the morning is much higher than them going to work in the evening. In addition, they probably always take the same route and see similar scenes. One can use these repetitive patterns to infer what is happening in a particular day or detect novelties. In this dissertation, we do not use time and place context for recognition, and it remains as an idea for future work.

1.5 Challenges

Egocentric video provides various opportunities that make recognition tasks simpler, but at the same time introduces new challenges.

1.5.1 Weak Supervision

Current approaches to activity and object recognition depend upon large amounts of labeled training data to obtain good performance. Labels can be automatically acquired for web videos from their tags and for movies and sign languages from their scripts and closed-caption text. While these are promising approaches, the required transcripts are not generally available for home videos. As a result, we need to seek other ways for learning objects and activities in egocentric videos. We overcome this problem by developing unsupervised and semi-supervised methods for segmentation and recognition of hands, objects and object states. We further use the interaction of detected objects and hands, as well as patterns of first-person attention to recognize daily activities.

1.5.2 Limited View

Many actions such as running, walking and jumping involve the body movement of the subject. It is very challenging to recognize these actions without seeing the limbs and the joints of the subject in the video. In egocentric setting, since the camera is worn by the subject, recognizing these actions become challenging. In particular, recognizing posture-defined activities is challenging because the egocentric view does not provide enough information about limbs' movement and their pose for recognition of these actions (for further information on this type of activities, refer to our older papers [54, 53] and the recent paper by folks at MSR Cambridge [158]). There has been a few approaches for recognizing these activities in egocentric video: (1) combining information captured by the egocentric camera with that of static cameras that are mounted in the scene; (2) using optical flow and other video-based features for recognizing these actions [92]; and (3) attaching various cameras to the first-person's limbs and joints [132].

1.6 Thesis Overview

1.6.1 Thesis Statement

Descriptive models of objects and activities can be learned in a weakly supervised setting by leveraging *attentional cues* that are available in egocentric video.

1.6.2 Summary of Contributions

There are two main keywords in the thesis statement that capture the main contributions of this document: (1) descriptive models and (2) attentional cues.

Descriptive Models:

- We develop algorithms that go beyond surface level recognition of activities. Our algorithms describe activities in more details based on their sub-actions, the location of the hands, and the participating objects and their states.

- It is shown that the activity, action and object labels are closely related and provide context for each other. In particular, it is demonstrated that inferring activity, action and object labels together results in better performance in comparison to inferring each separately.

- Left hand, Right hand and the active objects can be segmented from background in egocentric videos of daily activities. Our proposed method is completely unsupervised.

- Daily objects can be discovered from first-person's interactions with the world. Our method learns object models given the weak supervision available from their patterns of co-occurrence in daily activities.

- In addition to recognizing social activities, the faces are detected and recognized, their attention is estimated, and roles are assigned to the individuals in the scene. Furthermore, a social network of friends is built by counting the number of times they appear in each other's views.

- The relationship between actions and the state of the objects is leveraged to discover the object states and recognize the actions. Furthermore, it is shown that activities can be temporally segmented into actions by detecting the state of the objects in the environment at each frame.

Attentional Cues:

- The close relationship between first-person's gaze and actions is leveraged to show: a) a small circle around the gaze point is a strong determinant of the action, b) knowing the action label can improve gaze prediction and c) simultaneous inference of gaze and action results in better action recognition results.

- It is shown that implicit cues that are available from first-person's body such as head motion and hand location can be used to improve gaze prediction accuracy. Improved gaze prediction results in improved action recognition.

- A method is proposed for estimating where the faces are looking in the scene.
- It is shown that social interactions can be modeled based on the patterns of attention over time.
- Mutual eye-contact between the camera wearer and other individuals can be recognized by analyzing the orientation of their faces in egocentric videos.

Datasets:

- The collection of one of the first egocentric daily activity datasets: GTEA [56].
- The collection of the first-person social interaction at Disney Parks dataset [51].
- The collection of the GTEA Gaze and GTEA Gaze+ datasets [52], which contain first-person videos of daily activities with overlaid point of gaze in each frame. These datasets are the first in their kind.

1.6.3 Document Outline

In this thesis we develop descriptive models for two common categories of activities that we perform everyday: (1) interaction with objects and (2) interaction with people. Our goal is not only to recognize these activities, but also to develop methods that attempt to understand their details and semantics. We believe daily activities often consist of three main elements: objects, hands and faces, as depicted in Fig 5. The egocentric setting enables detection of these elements in video by providing first-person attention as an invaluable source of information. Interaction between these elements leads to recognition of daily activities.

Interaction with People:

Chapter 4 - Understanding Social Behavior in the Lab describes a method for understanding children's behavior in a setting following a standard psychology protocol. In this setting, the wearable eye tracking glasses are worn by an examiner who interacts with a child. A method is proposed for measuring the child's capabilities in making eye-contact with the examiner. Creating eye-contact is one of the necessary social skills that are developed during childhood, the lack of which can be an identifier of developmental disorders.

Chapter 5 - Understanding Social Behavior in the Wild leverages egocentric vision for understanding the social behavior of the individuals surrounding the first-person throughout the day. In this chapter, in addition to the first-person's attention, the attention of other individuals in the

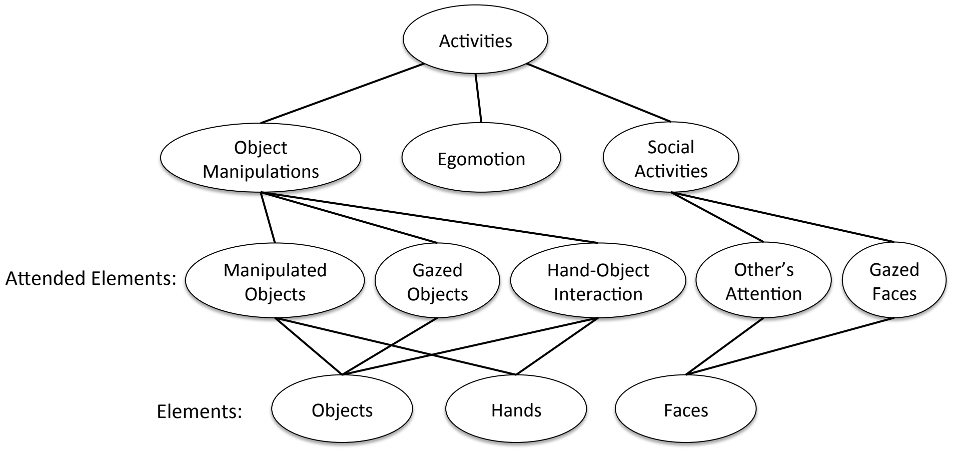


Figure 5: Egocentric activities are highly structured. We believe daily activities consist of three main elements: objects, hands and faces. Egocentric setting provides a framework combines these elements with first-person attention to recognize activities.

scene is also estimated. The dynamics of the group attention is modeled to recognize the type of social interaction in the scene.

Interaction with Objects:

Chapter 6 - Learning about Objects in Context describes a weakly supervised method for learning to recognize objects in egocentric videos. Objects that are used by the first-person throughout daily activities can be segmented out from the video, and their appearance can be learned. In addition, the label of these objects can be discovered by exploring the patterns of object occurrence in activities of daily living. For example, "bread" is used in making a peanut-butter sandwich and also in making a cheese sandwich. At the same time there are activities like making tea in which "bread" is not used. We can find regions in the videos of these activities that follow these patterns and most likely correspond to "bread".

Chapter 7 - Objects, Actions and Activities: Closing the Loop introduces a joint model for inferring objects, actions and activities all together. In this chapter, a multi-level graphical model is built for this purpose and approximate inference in this model is described. It is shown that the context provided by each element enhances the ability to recognize the others.

Chapter 8 - Learning the Function of Objects describes a method for learning finer-grained details about objects and actions. In previous chapters, objects and actions are segmented and recognized. In this chapter, the state of the objects (e.g. open vs close), existence of materials (e.g. coffee powder in spoon) and the reaction of the environment to the performance of actions (e.g. pouring milk into cup changes the state of the environment from including an empty cup to including a cup that contains milk) are learned.

Chapter 9 - Leveraging Gaze to Predict Action and Intention shows that having access to gaze measurements can significantly improve the action recognition results. In addition, when the gaze measurements are not available, gaze points can be modeled as latent variables and inferred during the test, which again results in significant gain in action recognition results. In this chapter, there are two methods proposed for gaze prediction, and their results are compared to each other.

Chapter II

PREVIOUS WORK

In this chapter, we discuss the relationship between this work with other areas of related work. Egocentric vision is an emerging area in computer vision. Every year around ten papers addressing this topic appear in top vision conferences (CVPR, ICCV and ECCV). It is possible to categorize these papers into roughly three main groups:

- Recognition of Activities
- Gaze in Egocentric Vision
- Day-long Video Summarization

In this chapter, we describe the related work to each of these topics.

2.1 Recognition of Activities

Action and activity recognition have been the subject of a vast amount of research in computer vision literature [124, 175]. Action recognition methods use various kinds of features to represent actions: features based on the entire human figure [21, 47], local space-time features [103, 130], mid-level features [54], features based on interaction of objects and hands [68, 125] and features based on point trajectories [121]. Even though space-time features and tracklets have obtained impressive results on challenging and realistic datasets [104, 145], they are not associated with semantic descriptions. In this thesis, our goal is to develop semantically-meaningful features that model actions based on the interaction between hands, objects and faces.

We categorize the previous work on activity recognition into three groups based on the kind of action classes they study. The first class of works consider body movements such as walking, running, etc, in which no other objects are involved other than the human body [21, 47, 54]. The second class of works consists of actions such as drinking, smoking, opening, etc in which the object context plays an important role [125, 186]. The third category contain group activities and social interactions [98, 34]. In this thesis our focus is on the last two categories.

2.1.1 Body-Posture based Activities

There are only a few works that study the recognition of body-posture based activities such as running, jogging, and sign languages from first-person view. Kitani et al. [92] recognize atomic actions like move left/right in outdoor activities such as snowboarding. Kemp [90] learns a kinematic model of the wearer via body mounted absolute orientation sensors and a head-mounted camera. Starner et al. [166] recognize American sign languages from the view of a head-mounted camera.

2.1.2 Object Manipulation Tasks

There have been various attempts in the past to model object context for action recognition. Mann et al. [117] use kinematic and dynamic properties of objects to understand their interactions. Moore et al. [125] use object context to classify hand actions. Li and Fei-Fei [111] use the object categories that appear in an image to identify an event. They provide ground truth object labels during learning in order to categorize object segments. Wu et al. [186] perform activity recognition based on temporal patterns of object use, using RFID-tagged objects to bootstrap the appearance-based classifiers. Ryoo and Aggarwal [151] combine object recognition, motion estimation and semantic information for the recognition of human-object interactions. Gupta et al. [68] use a Bayesian approach to analyze human-object interactions with a likelihood model that is based on hand trajectories. Marszalek et al. [119] demonstrate that the use of scene context improves action recognition performance. Yang et al. [189] treat the pose of the person in an image as a latent variable and use it to enhance action recognition. Yao and Fei-Fei [190] use the mutual context of object and human pose to recognize activities in images.

All of these methods use static cameras mounted in the environment. However, to capture daily activities of a person, even if the office and the home are densely instrumented with cameras, the system needs to go through the non-trivial challenge of focusing on hands and objects and coping with body occlusions. This makes wearable cameras an inevitable alternative. Ren and Gu [149] show that figure-ground segmentation improves object detection results in egocentric setting. Spriggs et al. [165] recognize egocentric meal preparation activities using visual data and other sensors. Pirsiavash and Ramanan [138] recognize activities of daily living by detecting active objects in video. This work is closely related to ours. The main difference is that our method is weakly supervised and does not require the labor intensive task of object annotation in the training set.

Segmentation of Active Objects: In order to detect the active objects, in Section 6.1, we introduce a method that segments the foreground containing hands and manipulated objects from background in egocentric videos. Foreground segmentation is a well addressed problem for fixed-location cameras. Various techniques have been developed, such as adaptive mixture-of-Gaussian model [167]. However, problem is much harder for a moving camera and is usually approached by motion segmentation given sparse feature correspondences (e.g. [179, 87, 187]). The most relevant work to our background subtraction method is Ren and Gu [149]. Given ground-truth segmentations, they learn a classifier on motion patterns and foreground object prior location, specific to their egocentric camera. In comparison, our segmentation method is completely unsupervised and achieves a higher accuracy. More recently Li and Kitani [109] propose a method for segmenting first-person's hands in egocentric videos.

Weakly Supervised Learning: in contrast to web data that is naturally associated with tags and annotations, it is hard to get labels for the daily egocentric videos. Thus, weakly supervised learning becomes an important topic when studying the recognition problems in egocentric vision. Furthermore, in general, reducing the amount of required supervision is a popular topic in various fields including computer vision, given the expense of labeled image data.

We develop a semi-supervised learning method for detecting objects in egocentric activities of daily living. Recent works have tried to provide different sources of automatic weakly supervised annotations by using web data [17, 58, 110] or cheap human annotation systems such as Amazon's Mechanical Turk [146]. Others have studied probabilistic clustering methods such as pLSA and LDA for unsupervised discovery of object topics from unlabeled image collections [162].

Unsupervised methods are not necessarily appropriate for learning object categories, since they have no guarantee of finding topics corresponding to object classes. An alternative approach is to expand a small set of labeled data to the unlabeled instances using semi-supervised learning. Fergus et. al [57] leverage semantic hierarchy from WordNet to share labels between objects.

More recently Multiple Instance Learning (MIL) has shown a great promise as a weakly supervised paradigm in both machine learning and computer vision communities [6, 79, 27, 180, 32]. In MIL, labels are provided for bags containing instances (e.g. images containing objects). These information are then leveraged to classify instances and bags. Buehler et. al [27] localize signs in footage recorded from TV with a given script. Vijayanarasimhan and Grauman [180] learn discriminative classifiers for object categories given images returned from keyword-based search engines. Prest et al. [140] learn object class detectors frm weakly annotated video.

In this thesis, we show that egocentric video captured from a wearable camera can potentially provide a new framework for learning objects given very small amount of supervisory information. We present results on reliably carving object classes out of large amount of daily video data by leveraging the domain constraints provided by our domain.

2.1.3 Social Activities

Some recent works address the problem of recognizing group activities such as standing in line and crossing street in images and videos. Lan et al. [98] use a discriminative latent SVM model to recognize the group activities in images based on individual actions and pairwise context. Choi et al. [34] recognize group activities in videos using features which capture the relative location of pedestrians in space and time. Patron-Perez et al. [133] extract features for human interactions like hand shaking from where the faces are oriented to. Ni et al. [129] recognize group activities in surveillance videos from self, pair and group localized causalities. Morariu and David [126] recognize multi-agent events in scenarios where structure is imposed by rules that agent must follow. A more recent work by Ramanathan et al. [144] look at the roles of the individuals in a social setting.

Our method differs in three ways from these works: (1) our videos are recorded from a first-person camera in which the bodies of other individuals are usually off-camera but faces and first-person head movement are easy to detect, (2) our focus is on categorizing extended social interactions such as conversations, and (3) we assign roles to individuals using patterns of attention and first-person movement. There are previous work that estimate where people are looking in the scene [16, 118]. However, our method goes beyond these works by showing that these attention patterns can be used for recognizing social interactions.

There has been a recent interest in building the social network of individuals present in movies or other types of video using computer vision techniques. Choudhury [35] recovers the social network and patterns of influence between individuals. Yu et al. [195] use face recognition and track matching to associate people together in videos using an eigen vector analysis method which they call modularity-cut. Ding and Yilmaz [43] group the movie characters into adversarial groups. In contrast to these works, our primary goal is to identify specific categories of social interaction and not estimate the social network structure for a group of individuals.

2.2 Gaze in Egocentric Vision

There are two main sources of information for identifying the object or the face that is being concentrated by the first-person: gaze and object-hand interaction. A system that captures human gaze, can identify the object that is being attended by first-person at any given time. In this thesis, not only we propose to use gaze for improving activity recognition, but also we propose to learn to predict human gaze. A system that can predict human gaze, knows where the important features for activity recognition lie in the video frames.

Previous gaze allocation models are usually derived from static picture viewing studies. This has led to methods for computation of image saliency [82, 147, 81] which use low-level image features such as color contrast or motion to provide a good explanation of how humans orient their attention. However, these models fail for many aspects of picture viewing [191] and natural task performance. Recently researchers have been using higher level information for gaze prediction. Torralba et al. [173] use global scene context features to predict the image regions fixated by humans performing natural search tasks. Judd et al. [85] show that incorporating top-down image semantics such as faces and cars improves saliency estimation in images.

Gaze for Analyzing Activities: There is a rich literature on using eye movement to analyze behaviors. Pelz and Consa [134] show that humans fixate on objects that will be manipulated several seconds in the future. Tatler et al. [172] state that high acuity foveal vision must be directed to locations that provide the necessary information for completion of behavioral goals. Einhauser et al. [48] observe that object level information can better predict fixation locations than low-level saliency models. Bulling et al. [29] look at eye movement patterns for recognizing reading. Liu and Salvucci [112] use gaze analysis for human driver behavior modeling. Land and Hayhoe [100] study gaze patterns in daily activities such as making peanut-butter sandwich and making tea. Researchers have shown that visual behavior is a reliable measure of cognitive load [168], visual engagement [163] and drowsiness [154]. Bulling and Roggen [28] analyze gaze patterns to identify whether individuals remember faces or other images.

Mishra and Aloimonos [122] introduce a method for segmenting objects using a given point inside them. The motivation for their method is that the gaze points usually fall on objects, and they can be used as seed points for segmenting objects in the scene. We believe the philosophy behind their algorithm has fundamental problems, since gaze points usually fall on the object edges and corners. Yi and Ballard [193] is the closest work to ours. They use a wearable eye-tracking system and wearable sensors on the hands to detect the grasped and gazed object for recognizing daily actions. In contrast to [193], we do not instrument hands with sensors. In this thesis, we propose to use gaze for recognition of daily activities and further we propose to predict gaze in videos of daily living in the context of first-person action. Furthermore, we propose a method for simultaneously inferring gaze and first-person action in egocentric videos. Furthermore, Bulling et al. [29] look at eye-movement patterns for recognizing reading. Liu and Salvucci [112] use gaze analysis for human driver behavior modeling.

Gaze for Identifying Developmental Disorders: A large body of behavioral research indicates that individuals with diagnoses on the autism spectrum disorder (ASD) have atypical patterns of eye gaze and attention, particularly in the context of social interactions [38, 106, 155]. Eye tracking studies using monitor-based technologies suggest that individuals with autism, both adults [93] as well as toddlers and preschool-age children [31, 84], show more fixations to the mouth and fewer fixations to the eyes when viewing scenes of dynamic social interactions as compared to typically developing and developmentally delayed individuals. Importantly, atypical patterns of social gaze may already be evident by 12 to 24 months of age in children who go on to receive a diagnosis of autism (e.g. [205, 184]).

Gaze for Interaction with Children: Several eye tracking methods for infants in daily interactions have been proposed in [137, 131, 60, 67]. In particular, Noris et. al [131] presented a wearable eye tracking system for infants and compared the statistics of gaze behavior between typical developed children and children with ASD in a dyadic interaction. Guo and Feng [67] measured the joint attention during a storybook reading, by showing the same book on two different screens and simultaneously track the eye gaze of the parent and his child by two eye trackers. However, these previous work either need a specially designed wearable eye trackers for infants [131, 137, 60, 123], or limit the eye tracking on a computer screen [67]. Our method, instead, only utilizes one commercial eye tracker for adult and is able to detect eye contacts between a child and an adult in natural dyadic interactions.

2.3 Day-long Video Summarization

An egocentric camera that is worn every-day by an individual captures thousands of hours of video. It is not possible for a human to go through all of these videos in a limited amount of time. As a result, video summarization is a key topic in egocentric vision. In this section we review the previous methods for video summarization. These works are related to our problem of finding the interesting moments in day-long videos at Disney parks, described in Chapter 5.

Key Frame Extraction: These works first segment the video into smaller clips such as scenes, shots, sub-shots, etc. A scene is all that happens in a particular location. A shot is shorter than a scene. There are three kinds of boundaries between shots: (a) cut: instantaneous change form one frame to another, (b) wipe: one shot gradually replaces another shot and (c) dissolve: one shot gradually appears while the other fades out.

Each shot or sub-shot is presented by a representative image of it, called the key-frame. Video segmentation is performed based on the similarity of shots. Various similarity measures are developed based on color [200], motion [200, 185] and objects [91].

Zhang [200] segment the video into shots, and represent each shot using key-frames. The key-frames are represented using visual features such as color, motion, etc. Shots are compared based on the similarity of their key-frames. The similarities are used for retrieving and browsing shots and key-frames. Wolf [185] uses motion analysis to select the key frames. Their algorithm computes optical flow for all frames, and selects the frames with the local minimum amount of motion in each shot. The reason for their strategy is that they believe actors pause when they want to emphasize gestures. Kim and Hwang [91] extract edges and filter the non-moving edges to segment objects of interest. They pick a key frame either if the number of regions between consecutive frames change (a new event), or if the shape of objects changes (an action). Dufaux [46] extracts the best shot and the best key frame by combining motion and spatial activity measures with skin color and face detection scores.

Most of the key frame detection methods first extract shots from the video and then extract key-frames from shots, however, there are works which extract key frames directly from the entire video [64].

There exist other methods used for selecting key frames. DeMenthon et al. [39] and Zhao et al. [201] use feature vector space based key frame selection. These methods represent each frame as a feature point in a multi dimensional feature space. A clip produces a curve in the space. They pick key frames based on curve properties such as sharp corners and direction change.

Montage of Still Images: Irani et al. [80] proposed to use image mosaics for summarizing video instead of traditional key frame extraction. They find the affine transformation between successive frames, choose one frame as the reference frame, and project other frames into the plane of reference coordinate system. Aner and Kender [7] further extended their method by presenting a new type of video abstraction. Caspi et al. [30] introduce dynamic stills as a better representation of summarized video. The advantage of these methods over key frames is that they provide better visualization of the motion and video content, however, they are not applicable in all settings, focus on human movements and fail in case of occlusions.

Collection of Short Clips: There are works which extract shorter clips such as shots and sub-shots and generate algorithms to pick the most important clips in order to render the summarized video. Ngo et al. [127] cluster the shots into multiple groups using normalized clustering. Shots are further down grained into sub-shots. They pick the representative shots from each cluster while keeping the temporal order of events. They decrease the length of shots by replacing them with the sub-shot that has the maximum attention score. Pfeifer et al. [136] use multiple cues such as scene boundaries, actions, mood (clips should represent the emotions existing in the longer video), dialog, etc. to extract representative clips from the video. Li and Sezan [108] use a model-based summarization. They summarize football games, by modeling the game as a sequence of plays, and detecting the start and end of play. There are papers such as Petrovic et al. [135] that use fast forward to shorten the video. In these method, the important events are shown slower while the unimportant ones are shown faster.

Saliency: Kang et al. [88] make a video montage by picking the highly salient space-time regions in order to summarize video. They use gradient to compute the saliency measure. They segment the video into layers of saliency and pick the more salient layers to form the target video. Other methods exist for finding salient regions in images and videos. Heidemann [75] uses color symmetry to find focus of attention in images. Itti and Baldi [81] present a system which is capable of detecting novel regions (regions of surprise) in video. They measure the performance of their system by comparing it to human gaze. They find a region surprising if the posterior of model given new data ($P(M|D)$) becomes far different than prior over model ($P(M)$). This is one step after Itti's work on finding salient regions in images [82]. Another well known but old biologically inspired visual attention system is Ahmad [4]. Ma et al. [116] combine different attention models to summarize videos. They compute attention measures of appearance, motion, camera movement, face, etc. and evaluate their results using user studies. In addition, people have looked at object of interest in video to decide on important parts [113]. In this paper they extract patches, then get the user input which determines in which frames the object of interest exists and in which it doesn't. Then they use MIL to find out the region which corresponds to object of interest. Then they summarize video by removing the frames in which the object of interest is less likely to appear.

Language: Analysis of audio and language can play an important role in video summarization. Smith and Kanade [164] integrate audio, video and textual information by coming up with heuristic summarization rules. They perform a user study for validating their experiments.

Simultaneous Spatial and Temporal Summarization: Pritch et al. [142] present a system using static (publicly mounted webcam) cameras. They extract the background by computing the median of pixel values. Given the background, they can segment the foreground regions. They present each object with a tube extending in time. They create a short video that best captures all of these tubes, without letting them overlap. Simakov et al. [160] summarize images or videos, in a way that the summarized smaller version video contains the set of features that existed in the original one, and at the same time does not introduce new features which did not exist in the original video.

User Study: Syeda-Mahmood and Ponceleon [170] incorporate user study for both annotating training data and for evaluating the performance. The user browses a long video by moving the browser forward and spending more time watching the interesting parts. They use these information to produce a measure for video summarization. Babaguchi et al. [11] learn people's personal preferences. They use the search and browsing patterns of individuals to create a profile for them. Each subject's profile contains a set of key words and weights associated with them, capturing their interests. They use profiles for subjective video summarization. Shamma et al. [157] experiment that the average length of the videos shared by people using instant messaging is about 5 minute. In this work they argue for moving from video content understanding towards systems and algorithms that can utilize the information about how the media content is used. Yu et al. [194] use the logs of user interaction with video data for summarization. They split the videos into shots and observe user behaviors such as watching shots, jumping from shots to each other and etc. to rank the shots.

Egocentric Vision: Aghazadeh et al. [3] detect novel scenarios from everyday activities. Lu and Grauman [115] and Lee et al. [105] summarize an egocentric video based on the important objects and faces. Singletary and Starner [161] use face detection to identify when first-person is involved in a social interaction.

2.3.1 Life-time Logging

Jim Gemmell and his colleagues at MSR created a life logging system, which records subject's data during her/his life using a variety of means including cameras, GPS, microphones, emails, documents, etc. They store this data and provide practical ways for fetching it upon user request. Aris et al. [9] bind GPS information with photos taken over time to provide a search method using time and location. Gemmel et al. [62] provide a nice quote from a friend to argue for using passive recording of daily life: "When I had my first child, I bought a camera and took many pictures. But eventually I realized I was living behind the camera and no longer taking part in special events. I gave that up - now I don't have nearly as many pictures of my second child". They present a lifetime recording system, consisting of a camera, accelerometers, infrared sensor and etc. Their system takes images when there is a change in image light, basically when the view or the environment is changed. Hodges et al. [77] perform a study on a patient with Amnesia and show that the life logging system significantly help her in recalling the events.

There are works on life long logging from other groups as well. Blanke and Schiele recognize daily routines such as working and commuting by using occurrence statistics of low level activities. Doherty and Smeaton [44] segment the saved images from MSR's SenseCam into events. The boundary of events are the images which do not match the adjacent images. The philosophy behind their work is that humans remember activities by splitting them into events [197]. Doherty and Smeaton in a more recent work [45] provide a method for augmenting the life logged data. They take logged images and use them to find similar images from similar events from the web.

2.3.2 Social Networks

There are recent works which address the problem of recognizing group activities such as talking, standing in line, crossing street, etc both in images and videos. Lan et al. [98] use a discriminative latent SVM model to recognize the group activities in images based on individual actions and pairwise context. Choi et al. [34] recognize group activities in videos using features which capture the relative location of pedestrians to a center one in space and time. They use random forests to both learn the binning parameters as well as the classifier. There are other less related works as well such as [126, 76]. Zen and Ricci [199] formulate the task of discovering high level activity patterns as a prototype learning problem.

Gibson [63] describes the conversational interactions as a set of turn taking scenarios. This paper provides great understanding of social interactions in general. Yu et al. [195] use face recognition and track matching to associate people. They analyze the associations using an eigen vector analysis method which they call modularity-cut. Ding and Yilmaz [43] group the movie characters into adversarial groups by analyzing the relations between characters and further determine the leader of each group.

Maria-Jimenez et al. [118] estimate face orientations in movies and use it to infer whether subjects are looking at each other. Dhar et al. [42] use features representing the composition of image, content of image and illumination to predict the aesthetics and interestingness of an image. Wang et al. [182] recognize the social relationships between people in the images based on the relative location of faces.

Chapter III

EGOCENTRIC VISION SENSORS

Recent advances in camera technology have resulted in creation of various tiny, light and high-quality cameras that are used in various kinds of wearable devices. Wearable cameras have been around for a long time in the lab setting. However, more recently they have become more popular and have started to become a part of everybody's daily life. Back in the 70's, Collins of the Smith-Kettlewell Institute of Visual Sciences developed a five pound wearable with a head-mounted camera that converted images into a 1024-point, 10" square tactile grid on a vest. In the 90's, Michael Land created one of the first head-mounted systems that consisted of a camera and an eye-tracker, as shown in Figure 6. Starner et al. [166] were also some of the first to use wearable cameras. They recognized activities of a person performing American Sign Language (ASL) in the captured videos.

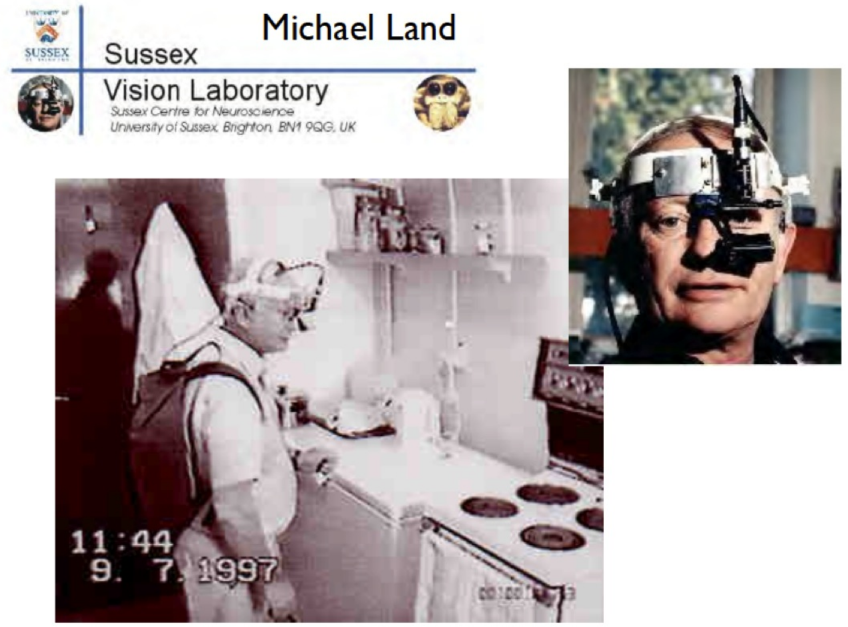

Michael Land
Sussex
Vision Laboratory
Sussex Centre for Neuroscience
University of Sussex, Brighton, BN1 9QG, UK
11:44
9. 7. 1997

Figure 6: Michael Land used a wearable device that recorded the scene in front of the user and would measure the point of the gaze.

The number of wearable devices has increased significantly over the last few years. In this section we will provide a brief description for some of these devices and my experience in using a few of them.

3.1 GoPro

GoPro[1] offers a series of small and high-quality cameras. These cameras can be mounted on various body parts or objects such as a helmet, chest and snow board which makes them very flexible. The GoPro Hero 1 has a fish-eye lens with a 110' view angle. The view angle has increased to 170' in the more recent Hero 2 and 3 cameras. The quality of the video is in HD, with minimal motion blur and other artifacts. The quality of the video is much better outdoors, and the video sometimes becomes very dark indoors. The main advantage of the GoPro camera is its wide field of view and its high quality video. On the other hand, its disadvantage is its bulkiness, and the fact that other individuals in the scene become very aware of it. The battery lasts for about 2 hours during continuous video capture. The data is recorded on a SD card. A 16GB SD card suffices for storing 2 hours of HD video. The price of GoPro is around 200$.

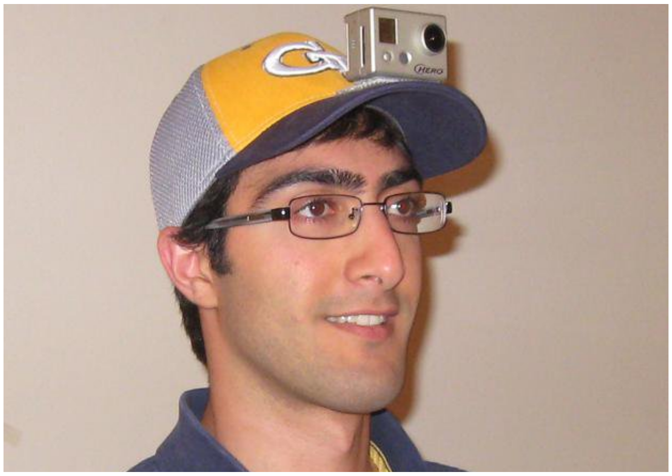


Figure 7: We used GoPro for collecting the GTEA dataset [56] and the Social Interaction at Disney park dataset [51]. We created a simple wearable device by mounting a GoPro camera on a cap using velcro tapes.

We used a GoPro camera for collecting the GTEA [56] dataset[2]. We mounted a GoPro camera on a cap as viewed in Figure 7. The fish-eye lens helps to capture a wide field of view, which is necessary for observing first-person's hands during daily activities. The fish-eye distortion creates some challenges for using the data, which we dealt with by camera calibration and image undistortion. At Disney Research Pittsburgh, we used a set of 10 GoPro cameras for collecting a dataset of social interactions at Disney Theme Parks [51]. Similar to the setup in Figure 7, we mounted the cameras

[1] http://gopro.com

[2] Mounting a GoPro on a cap using velcro was Takaaki Shiratori's idea. He helped me by providing the necessary tools for making a simple wearable device.

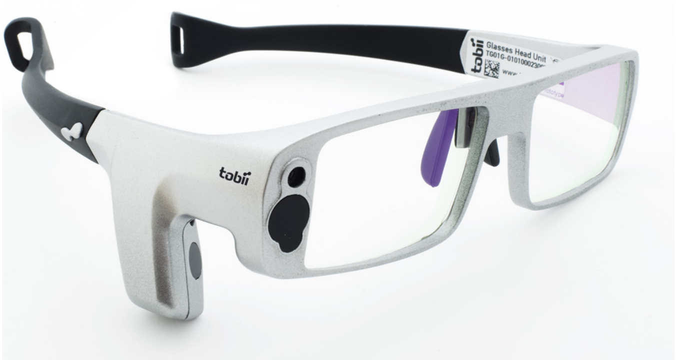


Figure 8: Tobii eye-tracking glasses.

on caps and handed them to the subjects. Each subject wore the camera throughout the whole day, and changed the battery and memory stick every 2 hours. Each subject was provided with 5 SD cards and 5 charged batteries for the data recording. A dataset by Pirsiavash and Ramanan [138] also uses GoPro cameras.

3.2 Tobii Eye-Tracking Glasses

Tobii[3] offers various kinds of eye-tracking products. Most of their devices are static and monitor-based. However, they have a few mobile eye-tracking systems as well. We used Tobii eye-tracking glasses for capturing GTEA Gaze dataset [52], as shown in Figure 8. The system consists of an outward looking camera that captures the scene in front of the user, and an inward looking infrared camera that tracks the subject's right eye. The glasses connect to a pocket size recording device. Before or after data collection, the system needs to be calibrated in order to correctly estimate the gaze point. Calibration is very intensive and becomes very hard for some subjects. The resolution of the video is 640×480 and the frame rate is 30 fps. The video quality is low and there exist severe motion blur and interlacing effects. On the plus size, the gaze tracking is accurate in comparison to the wearable gaze-tracking devices of other companies. The view field of the camera is around $60' \times 40'$. The price of Tobii eye-tracking glasses is around 30,000$.

3.3 SMI Eye-Tracking Glasses

SMI[4] produced a device similar to Tobii's eye-tracking glasses in January 2012 (Figure 9). They have tried to fix some of the issues that exist in the Tobii's system. In particular, their system is easier to calibrate, records a video in HD, and the glasses are more tolerable on the face. The SMI system has two eye-tracking infrared cameras looking at both eyes which results in an easier

[3] http://www.tobii.com

[4] http://www.eyetracking-glasses.com

calibration in comparison to Tobii's system. An issue that exists in the SMI glasses is that the video is blurred and dark on the frame boundaries. We have collected the GTEA Gaze+ dataset [52] using SMI's device. We have further collected a dataset of examiners interacting with toddlers following a standard psychology protocol. In our experiments, the examiner is wearing the device. The price of SMI eye-tracking glasses is around 24,000$.

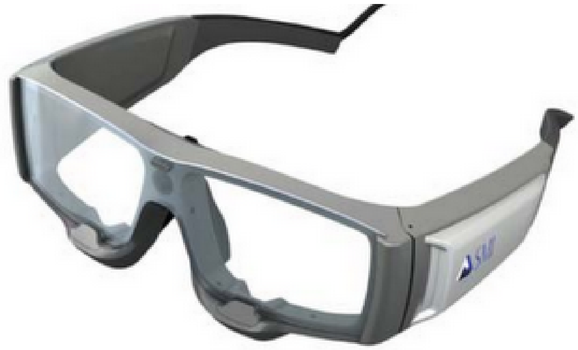


Figure 9: SMI eye-tracking glasses.

3.4 Pivothead Glasses

Pivothead[5] has introduced a relatively cheap pair of glasses that have an outward looking camera that captures the scene in front of the user. Obviously this is a cheaper system in comparison to SMI and Tobii because it doesn't track the eyes. The video quality is HD and it can capture for an hour. The only issue with the pivothead glasses is that the camera's field of view is very narrow, even narrower than that of SMI and Tobii systems. The price of Pivothead glasses is around 300$. They also provide a 100$ device which can transmit video to a laptop in realtime.

3.5 Google Glass

Google Glass is about to become available for public use. The system records a 720p video and takes 5-megapixel images. It can connect to the internet and any bluetooth-capable phone. In addition, Glass has a heads up display (HUD) creating an illusion equivalent to viewing a 25-inch high definition screen from eight feet away. It has 16 GB of RAM, 12 GB of which are usable for apps. Furthermore, it has a microphone, similar to all of the previously mentioned wearable devices. An image of Google Glass is shown in Figure 10.

[5] `http://pivothead.com`

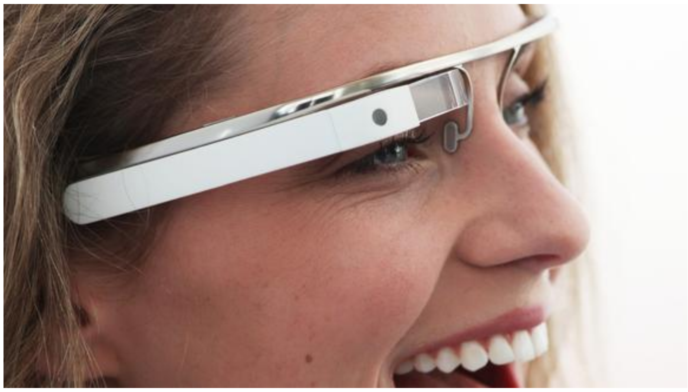


Figure 10: Google Glass.

3.6 Looxcie

Looxcie[6] offers a tiny camera that can be mounted on a person's ear and look out at the world at roughly eye-level. The camera captures a video at 30 fps at 640 × 480 resolution. The battery lasts for 3-5 hours. Lee et al. [105] use these cameras for data collection. The main advantage of this camera is its long battery life and its simple mounting scheme.

3.7 SenseCam

SenseCam is a life-logging camera with fish-eye lens and other sensors such as accelerometers, heat sensing and audio. The device is usually worn around the neck, and takes pictures once every few seconds. The battery lasts for a full day. Obviously the fact that this device does not capture video is its main disadvantage in comparison to previously mentioned devices.

3.8 HUDs

There are various options for heads up displays if one wants to build their own device. TacEye[7] is probably the most appropriate solution. Their systems cost around 3,000$. Other options are Liteye systems[8] and Recon Ski Goggles[9].

3.9 Fit-PC

One of the main parts of a real-time wearable device is its processing unit. The processing unit needs to be portable, light, and should not develop too much heat. The best known option in the market is Fit-PC[10] which costs around 600$.

[6] http://www.looxcie.com
[7] http://www.six-15.com
[8] http://www.liteye.com
[9] http://www.reconinstruments.com
[10] http://www.fit-pc.com/web

Chapter IV

UNDERSTANDING SOCIAL BEHAVIOR IN THE LAB

In the next two chapters, we will show that egocentric vision is a powerful domain for understanding the social behavior of the first-person. The work in this chapter is performed in collaboration with my colleagues Zhefan Ye and Yin Li. We have studied social behaviors in two settings: a) a controlled interaction between an examiner and a child in the lab and b) interactions between individuals in the wild. We will describe our method and experiments for analyzing social behavior in the lab in this chapter, and we will follow up with the social behaviors in the wild in Chapter 5. Similar to daily object manipulation activities that consist of objects, hands and their interactions, daily social activities consist of faces and their interactions, as depicted in Fig 11. The patterns of individuals' attention over time provides an invaluable cue for recognizing the type of social interaction.

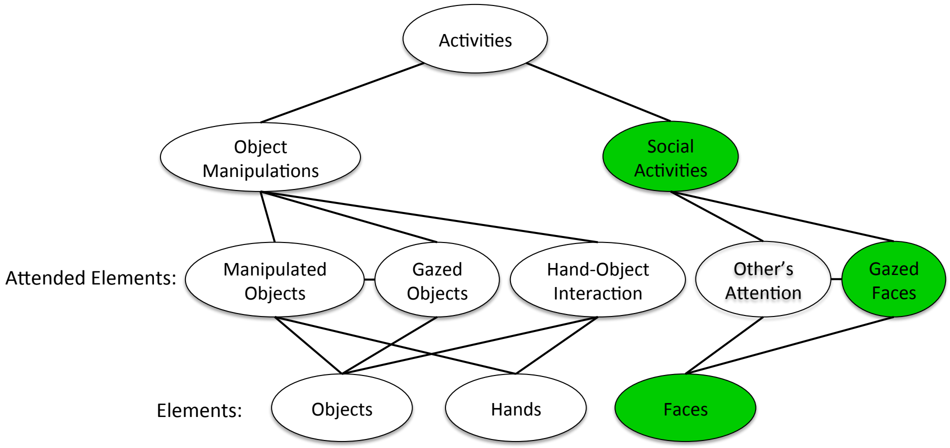


Figure 11: Social activities consist of faces and their attentions. In this chapter we introduce a method that uses faces and their attention, as well as first-person attention to recognize daily social activities.

In this chapter, we describe a system for detecting eye contacts between two individuals, which is based on the use of a single wearable gaze-tracking system. Eye contact is an important aspect of face-to-face social interactions, and measurements of eye contacts are important in a variety of contexts. In particular, atypical patterns of gaze and eye contacts have been identified as potential early signs of autism, and they remain important behaviors to measure in tracking the social development of young children. Our focus in this chapter is on the detection of eye contact events between an adult and a child. These gaze events arise frequently in clinical settings, for example when a child is being examined by a clinician or is receiving therapy. Social interactions in naturalistic settings, such as classrooms or homes, are another source of important face-to-face interactions between a child and their teacher or care-giver.

In spite of the developmental importance of gaze behavior in general, and eye contact in particular, there currently exist no good methods for collecting large-scale measurements of these behaviors. Classical methods for gaze studies require either labor-intensive manual annotation of gaze behavior from video, or the use of screen-based eye tracking technologies, which require a child to examine content on a monitor screen. Manual annotation can produce valuable data, but it is very difficult to collect such data over long intervals of time or across a large number of subjects. Furthermore, it is difficult for an examiner to record instances of eye contact at the same time that they are interacting with a child, and it is often difficult for an external examiner to assess the occurrence of eye contacts. In contrast, monitor-based gaze systems are both accurate and easily scalable to large numbers of subjects, but they are generally not suitable for naturalistic, face-to-face interactions. We present preliminary findings from a wearable system for automatically detecting eye contact events that has the potential to address these limitations.

The existence of wearable gaze tracking glasses, produced by manufacturers such as Tobii or SMI, have created the possibility to measure gaze behaviors under naturalistic conditions. However, the majority of these glasses are currently being marketed for use by adults, and their form-factors preclude them being worn by children. It remains a technical challenge to adapt wearable eye-tracking technology for successful use by child subjects, and there always remains the possibility that a subject will simply refuse to wear the glasses, regardless of how light-weight or comfortable they may be. It is therefore useful to consider an alternative, noninvasive approach in which a single pair of gaze tracking glasses are worn by a cooperative adult subject, such as a clinician or teacher, and used to detect eye contact events. This is possible because, in addition to capturing the gaze behavior of the adult subject, these glasses also capture an outward facing video of the scene in front of the subject. This video will contain the child's face in the case of a face-to-face interaction, and analysis of this video can be used to infer the child's gaze direction.

We present a system for eye contact detection which employs a single pair of gaze tracking glasses to simultaneously measure the adult's point of gaze (via standard gaze estimation) and the child's gaze direction (via computer vision analysis of video captured from the glasses). We describe the design and preliminary findings from an initial research study to test this solution approach. We believe this is the first work to explore this particular approach to eye contact detection.

System Setup: In order for an automatic system to detect eye contacts in a dyadic interaction, it needs to be capable of measuring each individual's gaze and determining if they are looking at each other. Here we describe some of the possible hardware and software options for building a system with such capabilities, and discuss the motivations for our system design.

Multiple Static Cameras: The most straightforward option is to instrument the environment with multiple cameras that are simultaneously capturing the faces in the scene. After synchronization of the cameras, we can then use available software for face detection and analysis, including face orientation estimation, eye detection, and gaze direction tracking, in order to identify eye contacts. However, this approach is fraught with practical difficulties: (1) the size of the interaction space is limited to the area that can be covered by a fixed set of cameras, (2) people might occlude each other in the cameras' views, (3) faces might be distant from the cameras and appear with low resolution, making the analysis of gaze extremely difficult, and (4) faces might not appear in frontal view, which makes eye detection and gaze estimation impractical. In addition to these issues, the cameras must be calibrated and the locations of the faces in the scene must be known in order to correctly interpret the computed gaze directions.

Mutual Eye Tracking: It is possible to track both the examiner's and the child's eyes using electrooculography based eye tracking systems or video based eye tracking systems. Electrooculography (EOG) based eye tracking systems place several electrodes on the skin around the eyes, and use the measured potentials to compute eye movement. In video based systems, infrared light is used to illuminate the eye, producing glints that can be used for gaze direction estimation. Unfortunately, it is probably not realistic to expect children to tolerate wearing either of these eye-tracking systems. This suggests a system based on a wearable eye-tracking device that can be worn by the adult examiner.

Examiner Eye Tracking (Our System): If only the adult examiner is able to wear the eye-tracking device, we need to record the first-person view video, i.e. egocentric video of the examiner in order to estimate the gaze of the child. We propose to detect eye contacts between the child and the adult examiner using the data captured from a wearable eyetracking device that is worn by the adult examiner. We use SMI's wearable eye-tracking glasses for this purpose. The device is similar in appearance to regular glasses, with an outward looking camera that captures a video of the scene in front of the examiner. They further have two infrared cameras that look at wearer's eyes and estimate their gaze location in video from the outward-facing camera. Our system uses a state of the art face detection system to detect the child's face in the egocentric video, and further estimate the child's gaze orientation in 3D space. Eye contact is detected if the child's gaze direction faces towards the camera, and examiner's gaze location falls on the child's face in the video.

This chapter contains the methods and experiments from the following paper: Ye et al. [192].

4.1 Wearable Eye Tracking

We use the SMI eye tracking glasses in this work. To track the eye gaze, the glasses use active infrared lighting sources. The surface of a cornea can be considered as a mirror. When light falls on the curved cornea of the eye, the corneal reflection, also known as a glint occurs. The gaze point can thus be uniquely determined by tracking the glints using a camera [70]. The SMI glasses track both of the two eyes with automatic parallax compensation at a sample rate of 30Hz, and record the high definite (HD) egocentric video at a resolution of 1280 960 for 24 frames per second. The field of view of the scene camera is 60 degree (horizontal) and 46 degree (vertical). The output of the eye tracking is the 2D gaze point on the image plane of the egocentric video. The accuracy of gaze point is within 0.5 degree. Figure 12 demonstrates the configuration of the SMI glasses in our experiment.

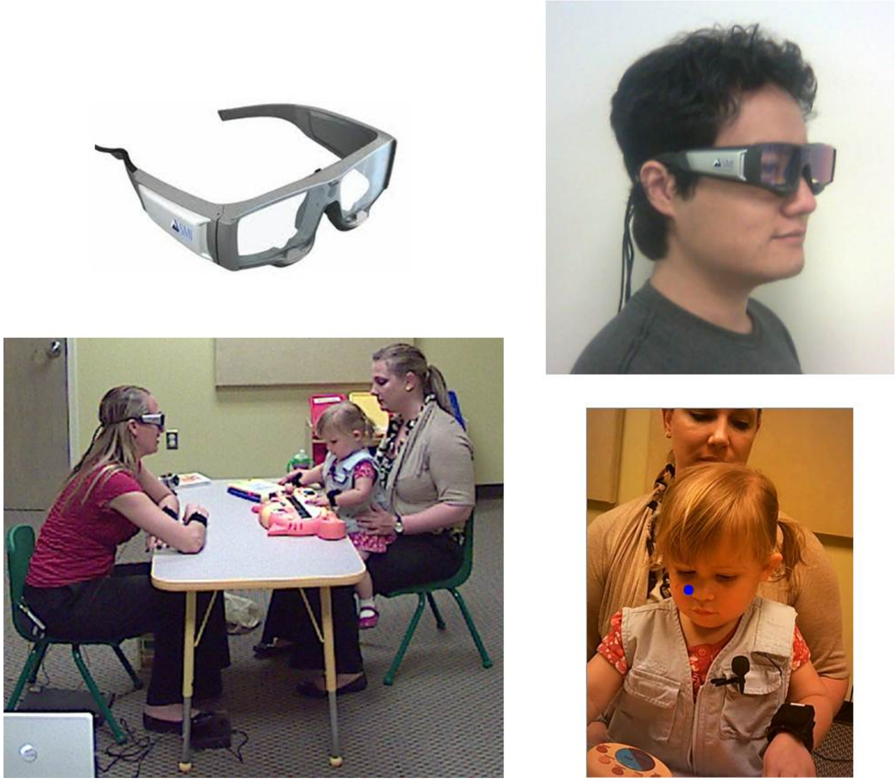


Figure 12: Experiment setup of our method. The examiner wears the SMI glasses and interacts with the child. The glasses are able to record the egocentric video with the gaze data. This figure is generated by Yin Li.

4.2 Face Analysis

The problem of finding and analyzing faces in the video is a well established topic in computer vision. The state-of-the art face detection and analysis algorithms are able to localize the face and facial parts (eyebrow, eye, nose, mouth and face contour) for real world scenarios. Recently, gaze estimation using the 2D appearance of the eye has been proposed [70]. Now, we can estimate a rough 3D gaze direction based on a single image of an eye with sufficient resolution. The core idea is learning the appearance model of the eye for different gaze directions from a large number of training samples.

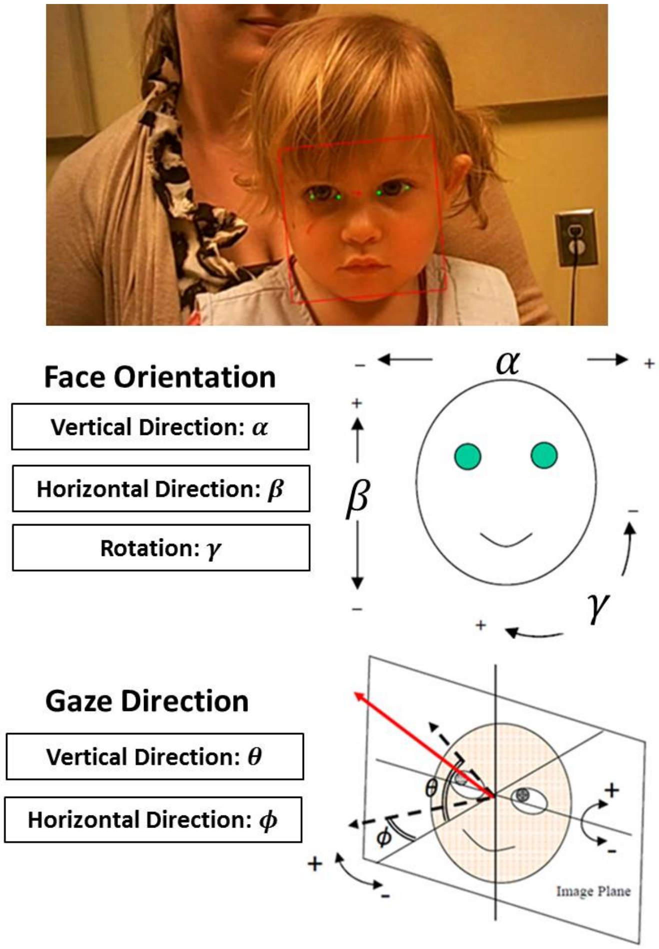

Face Orientation
Vertical Direction: α
Horizontal Direction: β
Rotation: γ
α
β
γ
Gaze Direction
Vertical Direction: θ
Horizontal Direction: ϕ
θ
ϕ
Image Plane

Figure 13: Face analysis results by the OKAO vision library, including the bounding box, facial parts, head pose and gaze directions. Red bounding box shows the the position and 3D orientation of the face. The green dots are the four corners of the eyes. The red line demonstrates the eye gaze direction. This figure is generated by Yin Li.

We rely on a commercial software, the OMRON OKAO vision library[1], to detect and analyze

[1] `http://www.omron.com/r_d/coretech/vision/okao.html`

the child's faces in the adult's egocentric video. The software takes the video as the input, localizes all faces and facial parts in the video, and estimates 3D head pose and gaze direction if the eyes can be found in the current face. As we observed from the experiments, though far from perfect, the software provides promising results, especially for the near frontally-presented faces. A illustration of the detection results can be found in Figure 13. The average processing time for the HD video is 15 frames per second.

4.3 Dataset 4: Examiner-Child Interaction

Our experiment is designed for two objectives: 1) to record the video and gaze data with minimum obtrusiveness for the children; 2) to allow the analysis of the data for eye contact detection.

Protocol: We designed an interactive session (5-8 minutes) for this purpose. In our setting, the examiner would wear the SMI eye tracking glasses and interact with the child, who was sitting across from her at a small table. A number of toys were also provided for casual play. We made sure that the examiner wear the glasses at the beginning of the session, such that it would not be a distraction for the child. In addition, the examiner was required to provide online annotation of each occurrence of eye contact by pressing a foot pedal. The gaze was tracked and the egocentric video was recorded during the interaction. We would expect a high quality image of the child's face in the egocentric video in this setting. The OKAO vision library was further applied to the video to obtain face information of the child, including the location and orientation of the face and the 3D direction of the gaze. The adult gaze information provided by the SMI glasses and the face information by the OKAO library was then used to determine moments of eye contact.

Participants: We report the result of a preliminary study based on a typically developing female subject, age 16 months. The recorded session lasts for roughly 7 minutes. Meanwhile, we continue to collect data and expect a larger sample in the future.

4.4 Detecting Eye-Contact

Our eye contact detection algorithm combines the eye gaze of the examiner (given by the SMI glasses) and the face information of the child, including eye gaze (given by the face analysis on the egocentric video). We extract features from both the gaze and face information for each frame of the video and train a classifier to detect the existence of an eye contact in this particular frame.

Feature Extraction: The extracted feature set includes the relative location (RL) of the examiner gaze point with respect to the child's eye center, the 3D gaze direction (GD) of the child with respect to the image plane (up and down/left and right), the 3D head pose of the child, i.e. head orientation (HO) and confidence of the eye detection (CE). The final feature is an 8 dimensional vector for each frame of the video, as shown in Figure 14.

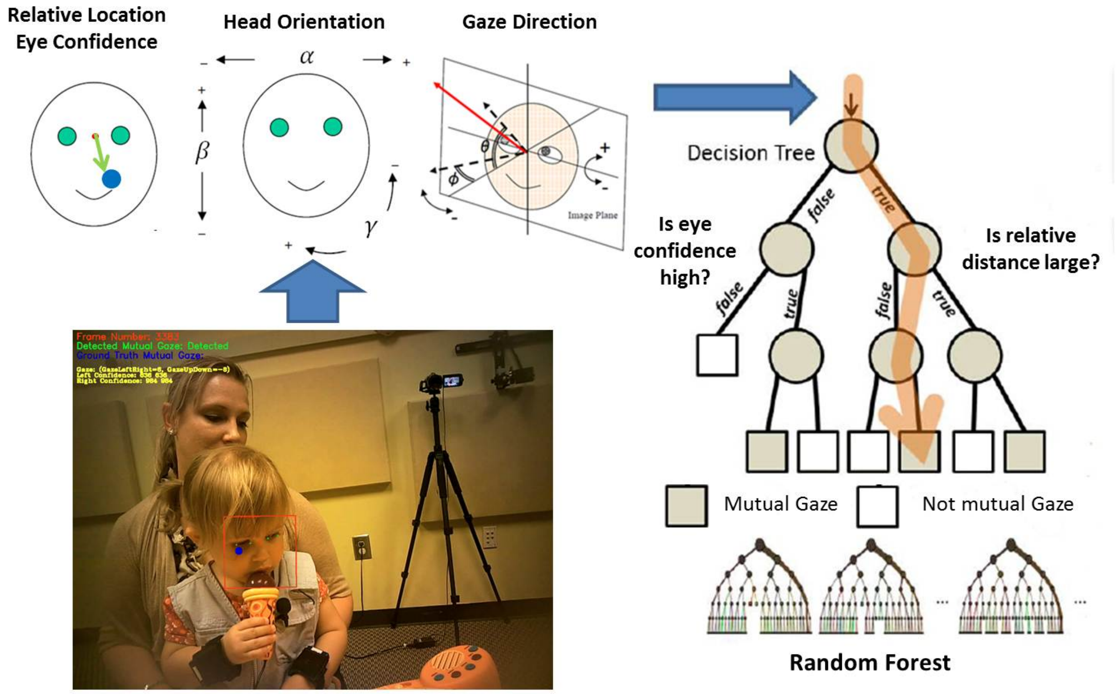
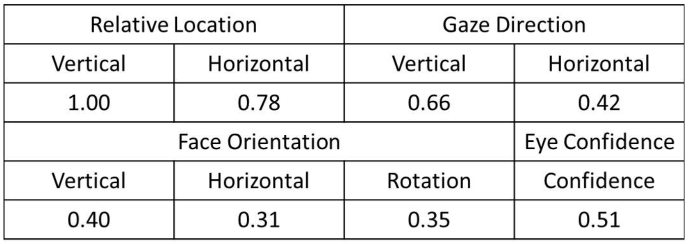


Figure 14: Overview of our approach. We combine the gaze data from the SMI glasses and face information from OKAO vision library. Features as the relative location, gaze direction, head pose and eye confidence are extracted and fit into a random forest. This figure is generated by Yin Li.

Eye Contact Detection: The problem of eye contact detection can be considered as a binary classification problem with the human annotation as the ground truth label. For each frame in the video and given the feature, our method need to decide whether there is an eye contact between the examiner and the child. We observe that even a simple rule would lead to reasonable results. For example, we threshold over the RL and GD, so that the eye contact is detected when the examiner's gaze point is close to the child's eyes and the child's gaze is facing toward the examiner. The rule can thus be encoded by a decision tree, as shown in Figure 14. However, both the adult's gaze data and the child's face information contain some errors. For example, the gaze estimation by the OKAO library includes substantial frame-to-frame variation. And it is inaccurate when the child is not facing toward the examiner or the facial part is not correctly located (See the second row of Figure 18).

To deal with all these problems, we train a random forest for regression [23] on the feature vectors using human annotations. A random forest is essentially an ensemble of single decision trees, as illustrated in Figure 14. It captures different models of the data, with each model a simple



Relative Location		Gaze Direction	
Vertical	Horizontal	Vertical	Horizontal
1.00	0.78	0.66	0.42
Face Orientation			Eye Confidence
Vertical	Horizontal	Rotation	Confidence
0.40	0.31	0.35	0.51

Figure 15: Ranking of the features by the random forest. The scores are normalized, such that 1 indicates the most important feature. Results by Zhefan Ye and Yin Li.

decision tree, and allows us to analysis the importance of different features (see [23] for the details of random forest). We train the model on the training set (see experiments of Section 4.5 for more details). The model can then detect eye contact in each frame. We leave the further temporal integration, such as an Hidden Markov Model, as future work.

4.5 Experiment 5: Detecting Eye-Contact

The human annotation by the foot pedal is considered as the ground truth for training and testing. We randomly select a subset (60%) of the data as the training set and the rest for testing, and train a random forest with 5 trees with the maximum height of 6. All results are averaged over 20 runs.

4.5.1 Features

We first analyze the importance of features with respect to eye contact detection. The random forest algorithm output the importance score of each feature based on its discriminative power. The result is shown in Figure 15. We find the three most important features are relative locations (both vertical and horizontal) of adult's gaze and vertical gaze direction of the child. The ranking yields an intuitive explanation: 1) the examiner gaze given by the SMI glasses is more reliable than the child's gaze given by OKAO vision library; 2) vertical gaze shifts have higher scores than horizontal ones in our experiments, since the former one is more frequent than the later one when the participants play with the toys on the desk.

4.5.2 Detection Performance

We consider eye contact detection as a binary classification problem, where the positive samples occupy a small portion of data. Therefore, the detection performance can be measured by precision and recall, defined as

$$\begin{aligned} Precision &= \frac{\#\, correct\, mutual\, gaze}{\#\, detected\, mutual\, gaze} \\ Recall &= \frac{\#\, correct\, mutual\, gaze}{\#\, real\, mutual\, gaze} \end{aligned}$$

Precision measures the accuracy of the detected eye contacts by the algorithm. Recall describes how well the algorithm is able to find all ground truth eye contacts. Please note that the human annotation is not prefect. This is reported by the examiner as not able to capture every eye contact due to heavy cognitive loading. Another possible problem is the reaction delay during the boundary of eye contacts. A second rater for the annotation of eye contacts would help to disambiguate these errors, which would be considered in future work.

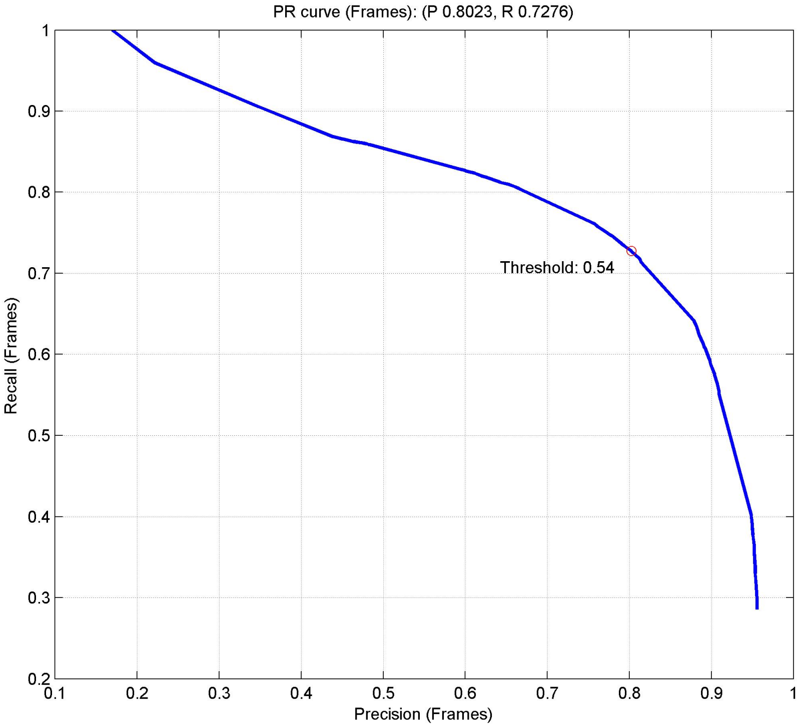
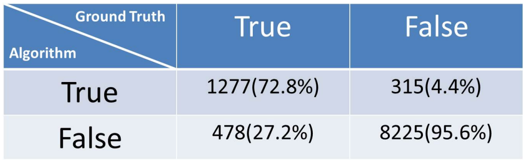


Figure 16: Precision recall curve of our eye contact detection algorithm. Results by Zhefan Ye and Yin Li.

The precision recall curve of our eye contact detection algorithm is shown in Figure 16. Each point on the curve is a pair of precision and recall by selecting different threshold on the regression results. We choose the threshold with the highest F1 score (the harmonic mean of precision and recall) in Figure 16. The optimal threshold is 0.54 that best balances between two different type of errors. For this threshold, the overall performance is reasonably good with the precision 80% and recall 72%.



Ground Truth / Algorithm	True	False
True	1277(72.8%)	315(4.4%)
False	478(27.2%)	8225(95.6%)

Figure 17: Confusion matrix of the frame level eye contact detection results. Results by Zhefan Ye and Yin Li.

The confusion matrix for the optimal threshold is also shown in Figure 17. Our algorithm has more false negative than false positive. The main reason of the errors, as we find in the data, is that the OKAO vision library fails to estimate the correct gaze direction in the video and thus the algorithm fails to detect eye contacts. Some of the successful and failure cases of our algorithm is demonstrated in Figure 18, which displays preliminary results from a second subject.

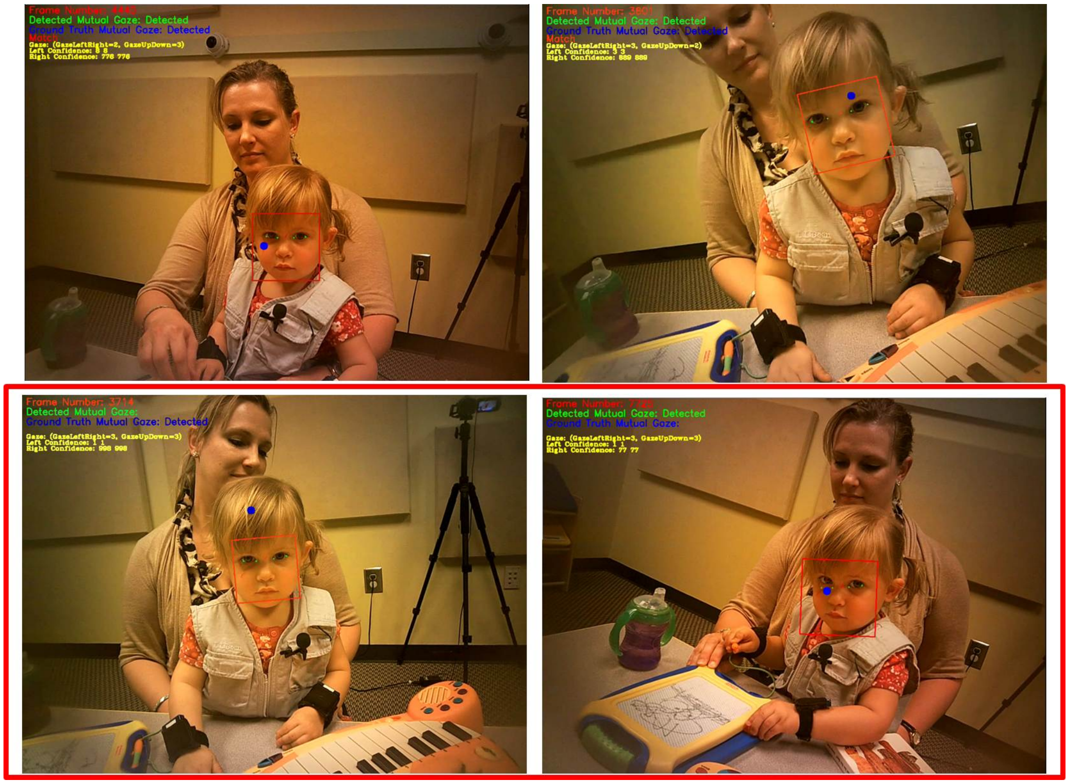

Frame Number: 4445
Detected Mutual Gaze: Detected
Ground Truth Mutual Gaze: Detected
Match
Gaze: (GazeLeftRight=2, GazeUpDown=3)
Left Confidence: 8 8
Right Confidence: 776 776
Frame Number: 3801
Detected Mutual Gaze: Detected
Ground Truth Mutual Gaze: Detected
Match
Gaze: (GazeLeftRight=3, GazeUpDown=2)
Left Confidence: 3 3
Right Confidence: 889 889
Frame Number: 3714
Detected Mutual Gaze:
Ground Truth Mutual Gaze: Detected
Gaze: (GazeLeftRight=3, GazeUpDown=3)
Left Confidence: 1 1
Right Confidence: 998 998
Frame Number: 7725
Detected Mutual Gaze: Detected
Ground Truth Mutual Gaze:
Gaze: (GazeLeftRight=3, GazeUpDown=3)
Left Confidence: 1 1
Right Confidence: 77 77

Figure 18: Example of successful (first row) and failure (second row) cases of our algorithm. The figure is generated by Zhefan Ye and Yin Li.

4.6 Conclusion and Future Work

We have described a system for detecting eye contact events based on the analysis of gaze data and video collected by a single pair of wearable gaze tracking glasses. Our system can be used to monitor eye contact events between an adult clinician, therapist, teacher, or care-giver and a child subject. We present encouraging preliminary experimental findings based on an initial laboratory evaluation.

Our ultimate goal is to have a system that can be used by psychologists. However, currently the robustness and accuracy of our algorithm is still far from human performance. A simplification for making our system affordable and easy to use for public is to replace the SMI eye-tracking glasses with a cheap wearable camera. Then instead of finding the moments of mutual eye-contact, we find the moments when the child looks into examiner's eyes. Since our main interest is in analyzing child's behavior, we believe it is fine to give up on examiner's gaze.

Chapter V

UNDERSTANDING SOCIAL BEHAVIOR IN THE WILD

In this chapter, we address the problem of detecting and characterizing social interactions in the wild. Our method is based on the attention of the faces and their interactions, as depicted in Figure 19. The patterns of individuals' attention over time provides an invaluable cue for recognizing the type of social interaction.

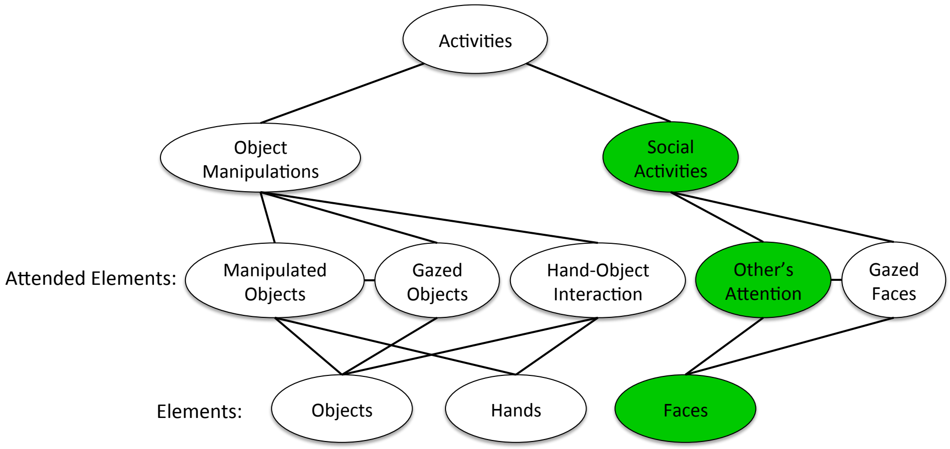


Figure 19: Social activities consist of faces and their attentions. In this chapter we introduce a method that uses faces and their attention, as well as first-person attention to recognize daily social activities.

Recognizing social interactions has various applications. A potential application is mining the interesting moments and activities from the videos recorded through out the day. For example a day at an amusement park or a zoo. Too often the desire for a tangible video record of such an outing results in one or more individuals playing the role of "group videographer" and spending much of their time behind the viewfinder of a camcorder. This videographer role may prevent these individuals from fully participating in the group experience. More importantly, the interesting moments and shared experiences that are the most significant often occur spontaneously, and can be easily missed. After the joke and the laughter have passed, it is too late to turn on the camcorder. This dilemma is summed up nicely by a quote from [62]: "When I had my first child, I bought a camera and took many pictures. But eventually I realized I was living behind the camera and no longer taking part in special events. I gave that up - now I don't have nearly as many pictures of my second child.”

The recent popularity of high-quality wearable camcorders such as the Go-Pro have created an opportunity to revisit the problem of experience capture. However, continuous capture of video footage at a park or some other outing will also result in hours of footage that is uninteresting: walking between rides, standing in line, etc. Our thesis is that the presence or absence of social interactions is an important cue as to whether a particular event is likely to be viewed as memorable. We believe social interactions, such as having a conversation, are tightly coupled with whether a moment is worth keeping.

Another application of recognizing social activities is for child screening and identifying developmental disorders such as autism. In these scenarios, not only we are interested in the type of social interaction, but further in the quality of child behavior and shift of attention. In Chapter 4 we show a method for analyzing temporal patterns of attention for child screening. However, in this chapter our focus is on recognizing categories of social interactions.

We categorize social interactions into three sub-types: dialogue, discussion, and monologue, which characterize whether the interaction involves multiple people (discussion) or a single subject (dialogue) and whether it is interactive (discussion) or largely one-sided (monologue). We present a method for automatically detecting and categorizing social interactions in egocentric video. Our method makes it possible to capture a continuous record of an outing and then distill from it the most salient moments.

First-person video is an obvious choice for capturing personal day-long experiences in an amusement park, or other social events in which thousands of individuals participate. In this context, first-person video provides many advantages in comparison to fixed video recorders: (1) the first-person camera always records where the wearer is attending and provides natural videos of her family and friends, (2) occlusions are less likely in an egocentric setting, because the wearer naturally moves to provide a clear view and (3) it is not practical to simultaneously track all of the individuals in an amusement park and record all of their interactions using static cameras.

Our method uses two sources of information for analyzing the scene in order to detect social interactions: (1) faces and (2) first-person motion. The work-flow of our approach is shown in Fig 20. We transfer detected faces into the 3D scene and estimate their locations and orientations. The location of the faces around the camera wearer provides significant evidence for the type of social interaction. Further, the social interactions are characterized by the patterns of attention shift and turn-taking over time. We therefore estimate these patterns like who looks at who or where, whether a group of individuals look at a common location, etc. We analyze the patterns of attention over

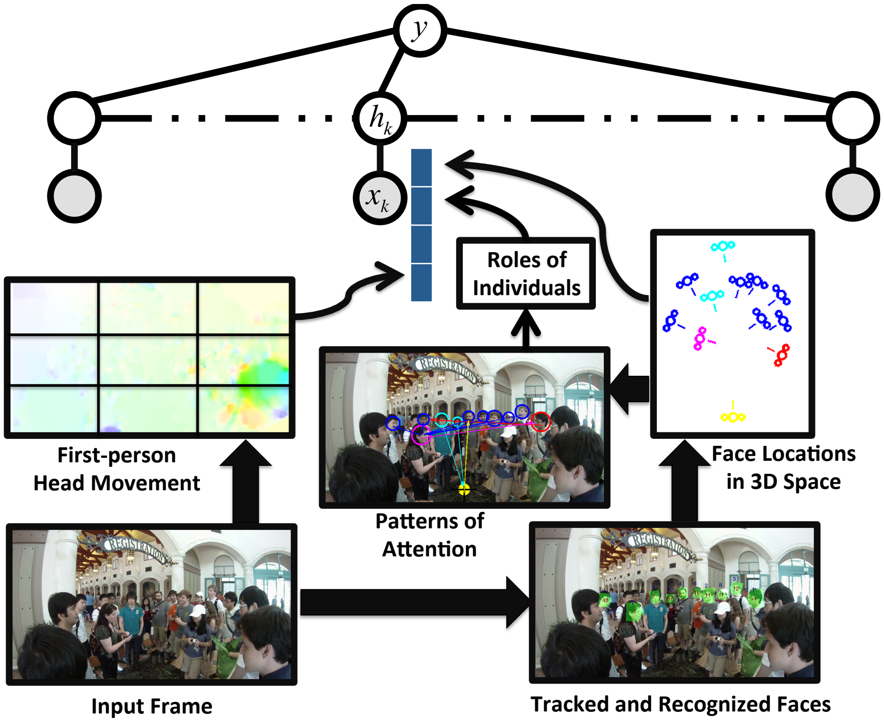

y
h_k
x_k
Roles of Individuals
First-person Head Movement
Patterns of Attention
Face Locations in 3D Space
Input Frame
Tracked and Recognized Faces

Figure 20: Work flow of our method.

time to recognize the type of social interaction. For example, when most of the individuals in a group are looking at a particular person over a long period of time, our algorithm will label this as a monologue. In addition to patterns of face locations and attention, the head movements of the first-person provides additional useful cues as to their attentional focus.

We believe this is the first work to utilize egocentric video in order to detect and categorize social interactions among groups of individuals. Our focus on real-world social events, such as trips to an amusement park, make the task especially challenging due to the complex visual appearance of natural scenes and the presence of large numbers of individuals in addition to the social group of interest. We believe this work can encourage other researchers to tackle this challenging new problem domain, and we will provide a large, extensively-annotated video dataset to support this goal. This chapter makes four contributions: (1) we introduce a method for detection and analysis of social interactions such as monologue, discussion and dialogue, (2) we address this problem from the first-person point of view, which is crucial for capturing individual experience, (3) we present a dataset of 8 subjects wearing a head-mounted camera for a full day at a theme park, containing more than 42 hours of natural and realistic video and (4) we develop a method which estimates the patterns of face attentions in video and analyzes these patterns over time to detect the social interactions. The method in this chapter is going to be presented in CVPR 2012 [51].

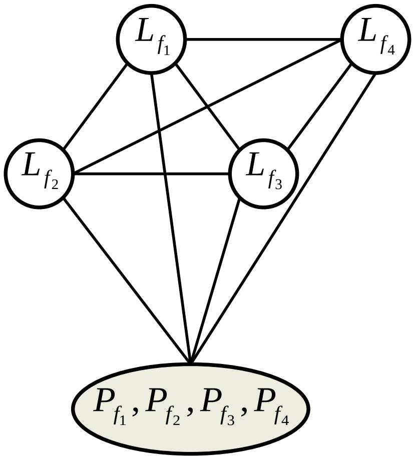


Figure 21: MRF model for inferring where each person is attending. The observations P_{f_i} contain the location and orientation of the face f_i in the scene, and the hidden variables L_{f_i} are the 3D location at which the face f_i is looking.

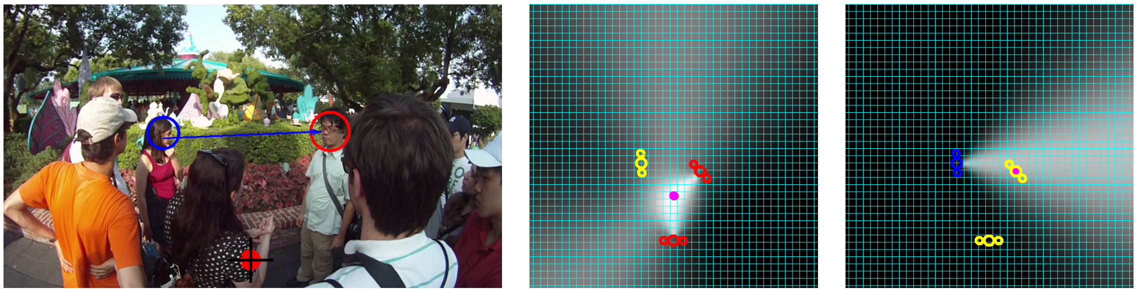
5.1 Faces and Attention

In this section, we describe our method for estimating the location and orientation of faces in space and specifically estimating the patterns of attention (to whom or where in 3D each face is looking). In Section 5.2, we analyze the attention patterns over time to detect the types of social interaction in video. We use faces as our main source of information because (1) faces and their attention patterns play the main role in social interactions and (2) the state of the art computer vision methods for face detection and recognition are more robust in comparison to algorithms for detection of pedestrians or other objects.

Given only one face's location and orientation in the scene, we can estimate its line of sight but it is not possible to estimate where in space it is looking at. However, we show that the context provided by other faces can help to estimate where each faces is attending in space.

We start by tracking the faces in video. Then we identify individuals by clustering the face tracks into multiple bins. In addition, we compute the orientation (yaw, pitch, roll) of every detected face [1]. In each frame, we estimate the location of every face in 3D space with respect to the first-person. Since our videos are recorded from a linear scale fish-eye lens, we can estimate a face's view angle θ from the camera by $\theta = \frac{r}{f}$ where r is the pixel distance of the center of the face from the image center and f is the camera's focal length. We use the height h of a detected face to approximate its distance d from the camera by $d = \frac{c}{h}$ where c is a constant. We estimate c and f by calibrating our

[1] We use Pittpatt software (http://www.pittpatt.com) for face detection and recognition.



(a) (b) (c)

Figure 22: MRF Inference Procedure. Our method groups the faces looking at a common location together. In (a) the color of the circle around the face determines the group it belongs to. The camera wearer and the man on the right are looking at the lady wearing a polkadot shirt. In (b), our algorithm cannot detect lady's face, but realizes that the first-person and the man are looking at the same location in space. In (c), our algorithm estimates that the lady with the red shirt is looking at the man.

cameras, asking multiple subjects to stand at pre-defined locations and orientations with respect to a camera mounted on a tripod. We estimate the face orientations in 3D using its computed orientation in 2D image. Examples are shown in Fig 23(a-c).

Only a subset of individuals present in the scene are visible in each frame. This issue impacts the effectiveness of our attention estimation method. We solve this problem by building a map of faces around the first-person at local time intervals. Our assumption for making these local maps is that the positions of faces around the first-person does not significantly change locally in time. For each interval, we first pick the frame with the maximum number of faces as the reference frame. We initialize the 3D location of the faces in the reference frame. We set the origin of the world coordinate frame to the camera coordinate in the reference frame. We iteratively add the faces in adjacent frames to the map. For each frame, we match the faces to the ones already added to the map based on their assigned cluster number acquired in the recognition process.

The location and orientation of a face in 3D provides us with an approximate line of sight. We use the context provided by all the faces to convert lines of sight into 3D locations. We make three assumptions to achieve this goal: (1) It is more likely that a person looks at something in the direction of her face's orientation, (2) a person looks at a person with a higher probability than at other objects, (3) if other people in the scene are looking at a particular location, then it is more probable for a face that is oriented towards that location to be looking at it as well. Next we describe our method for estimating where faces attend.

5.1.1 Reasoning about People's Attention

Our goal is to find out where each person is attending in 3D space. We build an MRF (Fig 21) in which the observations P_{f_i} contain the location and orientation of the face f_i in the scene, and the hidden variables L_{f_i} are the 3D location at which the face f_i is looking. To make the inference feasible, we discretize the space into a grid at a resolution of $5cm \times 5cm$. Our goal is to estimate at which grid point each face is looking. The label space for each L_{f_i} is the set of grid locations. We have depicted an example in Fig 22.

Our MRF model in the case of four faces is shown in Fig 21. The unary potentials capture the likelihood of looking at a grid cell based on the observations, while the pairwise terms model the context between faces in the scene. The pairwise terms model the likelihood of looking at a grid cell given where other faces are looking.

Unary Potentials: Consist of three terms as follows:

$$\begin{aligned}\phi_U(L_{f_i}, P_{f_1}, P_{f_2}, ..., P_{f_N}) &= \phi_1(L_{f_i}, P_{f_i}) \times \\ & \phi_2(L_{f_i}, P_{f_i}) \times \\ & \phi_3(L_{f_i}, P_{f_1}, ..., P_{f_N})\end{aligned}$$

where f_i represents face i in the scene, L_{f_i} is the location at which f_i is looking at in space, and $P_{f_i} = \begin{bmatrix} V_{f_i} \\ T_{f_i} \end{bmatrix}$ contains the orientation unit vector V_{f_i} and location vector T_{f_i} of the face f_i. The first potential ϕ_1 is modeled as a Gaussian function that computes the possibility of f_i looking at a location ℓ based on f_i's location and orientation in space:

$$\phi_1(L_{f_i} = \ell, P_{f_i}) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left\{-\frac{\|V_{f_i} - \overline{(\ell - T_{f_i})}\|^2}{2\sigma_1^2}\right\}$$

where σ_1 is the standard deviation. The second potential ϕ_2 is modeled as a sigmoid function to put a threshold on how close $L_{f_i} = \ell$ can be to the face f_i, added mainly to avoid a face looking at itself:

$$\phi_2(L_{f_i} = \ell, P_{f_i}) = \frac{1}{1 + \exp\{-(c_2.\|\ell - P_{f_i}\|)\}}$$

where c_2 is a constant. Finally the third term ϕ_3 is meant to bias faces to look at where other faces are in comparison to looking at objects:

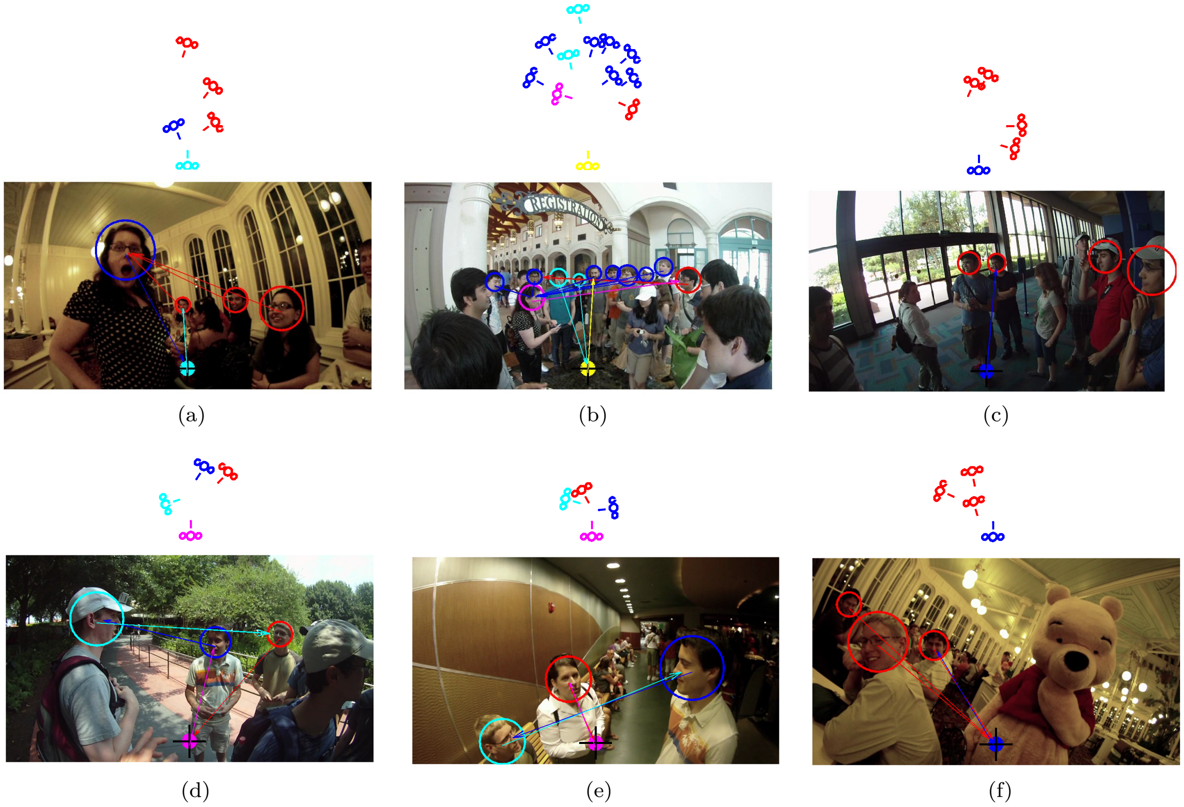


Figure 23: Faces attending to a common location are shown with the same color. The bird's eye view of the location and orientation of faces in 3D space is shown. In addition, faces are connected to the person they are looking at in each image. The first person is shown by a circle at the bottom center of the images. Note that our method can estimate the common attention even if the faces are not looking at a person (c).

$$\phi_3(L_{f_i} = \ell, P_{f_1}, ..., P_{f_N}) \quad = \quad \begin{cases} c_3 & \ell = P_{f_j} \forall j \neq i \\ 1 & otherwise \end{cases}$$

where c_3 is a constant increasing the chance of f_i looking at a location ℓ if another face f_j is at that location.

We set the parameters σ_1, c_2 and c_3 using the training data. We manually annotate faces looking at each other in a set of frames and learn the parameters from these examples.

Pairwise Potentials: The binary potentials capture the interaction between people. They bias the faces towards looking at the same location in the scene. Basically, if others are looking at something in the scene, the probability that another person is looking at the same thing is higher. We define the following function for the binary potentials:

$$\phi_B(L_{f_i} = \ell_1, L_{f_j} = \ell_2) \quad = \quad \begin{cases} c_B & if(\ell_1 = \ell_2) \\ 1 - c_B & if(\ell_1 \neq \ell_2) \end{cases}$$

where c_B is a constant greater than $\frac{1}{2}$ and smaller than 1. We set c_B by cross validation on the annotated examples.

Optimizing the MRF: We need to optimize the MRF to infer the locations $L_{f_i} = \ell$ where each face f_i is attending. There are a large number of possible locations (cells in the grid) and there can be up to 10 faces in a frame in some cases. Because the location at which a face is looking at is dependent on that of other faces, exact inference is intractable. We propose an approximate algorithm to solve this problem which is inspired by the α-expansion method. Our algorithm iteratively groups or separates faces based on whether they are looking at a common location or not.

Our algorithm starts by assigning each face's attention to a location by only optimizing its unary terms. Thus, faces are first assigned to different groups. In the next stage, it considers both unary and pairwise terms and iteratively merges or splits the groups. At each step, it considers a pair of groups and measures if the total MRF energy increases as a result of merging them. If it does, the two groups are merged. Similarly, in each group, it measures whether removing a face increases the total energy. The procedure iterates until convergence. An illustration of this procedure is depicted in Fig 22. Qualitative results are shown in Fig 23.

5.2 Detecting Social Interactions

In this section we describe our approach for detecting and recognizing types of social interactions in day-long first-person videos. We introduce three categories of features and provide an analysis of their capability to describe social interactions: (1) location of faces around the first-person, (2) patterns of attention and roles taken by individuals and (3) patterns of first-person head movement. We use these features in a framework that explores the temporal dependency over time to detect the types of social interactions.

5.2.1 Location of Faces around First-Person

Important evidence for the detection of social interactions is provided by the location of faces in the 3D space around the first-person. This is very similar in nature to the approach of [34], where they use the relative location of pedestrians to categorize group activities. For example, one can imagine that in a monologue, faces tend to appear in a circle around the person who is talking to the rest. In a dialogue a face tends to appear in front of the camera, looking at the first-person. To build location-based features, we divide the area in front of the first-person into 5 angular bins (from -75 to 75 degrees) and 4 distance bins (from 0 to $5m$). Our method counts the number of faces in each bin, and returns a 20 dimensional histogram as a feature.

5.2.2 Attention and Roles

Social interactions are characterized by patterns of attention between individuals over time. When a person speaks, she attracts the attention of others. Once another individual takes the floor, the attention shifts to the new person.

Our idea is that during a social interaction, each individual present in the scene adopts a specific role. For example, in a monologue, there is a particular role that can be assigned to the person who is speaking, and another role played by the individuals listening to the speaker. Analyzing the change in roles over time can describe the patterns of turn taking and attention shift that are crucial elements of social interactions.

We assign roles to individuals based on four features that capture the patterns of attention for each individual x:

- Number of faces looking at x
- Whether first-person looks at x
- If there is mutual attention between x and first-person (both are looking at each other)
- Number of faces looking at where x is attending

We assign a 4 dimensional feature vector to each individual and then cluster all the examples in training sequences to a few bins using k-means. Each bin represents a role. We represent each frame by building the histogram of roles involved in a short interval around that frame.

5.2.3 First-Person Head Movement

A further cue for the categorization of social interactions is provided by the first-person head movement. The movement patterns complement the coarse attention estimations with transition information. In addition, in cases where two individuals are speaking while walking, faces are absent from the video, the first-person head movement provides significant information.

We propose an additional feature to capture first-person head motion patterns. We extract features from dense optical flow [24] at each frame. We split each frame horizontally and vertically into a few sub-windows. We split the flow vector field in each sub-window into horizontal and vertical components, V_x and V_y, each of which is then half-wave rectified into four non-negative channels

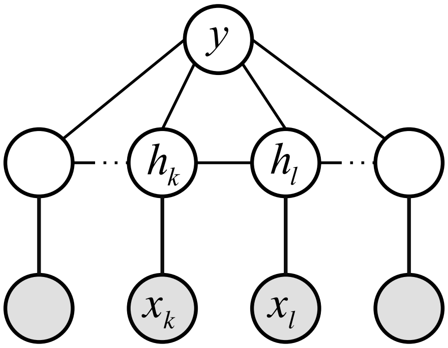


Figure 24: Our model. y is the social interaction label, h_l is the hidden state label assigned to frame l and x_l contains the features extracted from a local window around frame l.

V_{x^+}, V_{x^-}, V_{y^+} and V_{y^-}. We represent each sub-windows with a vector containing the mean value of its motion channels.

5.2.4 Temporal Model

The features described in previous sections encode a local snapshot in time. However, the temporal change in these features is crucial for detection and understanding of social interactions. The intuition behind our solution is that each frame is assigned to a state based on its features and then an interaction type is assigned to the whole sequence based on the state labels and their dependencies. We model our problem with Hidden Conditional Random Field (HCRF) [143] for this purpose. In our model (Fig 24), frames are assigned hidden state labels and these states are connected by a chain over time. In HCRF, state labels are latent variables and are learned by the algorithm.

The HCRF model is learned over the following potential function Ψ:

$$\begin{aligned}\Psi(y, \mathbf{h}, \mathbf{x}; w) &= \sum_{i=1}^{n} w_{h_i}.\varphi_{x_i} + \sum_{i=1}^{n} w_{y,h_i} \\ &+ \sum_{(k,l)\in E} w_{y,h_k,h_l}\end{aligned}$$

where the graph E is a chain with nodes corresponding to hidden state variables, φ_{x_i} contains the feature vector from the small sub-window around frame i, and w contains the parameters of the model, which are learned during training using BFGS optimization. The label assigned to the whole sequence y, takes binary values in case of detection and takes multiple values (dialogue, discussion, monologue, walk dialogue, walk monologue) when trained for recognition. During the test, the label y for which the potential Ψ is maximum is assigned to the sequence.

5.3 Dataset 5: Social Interactions at Disney Parks

To collect our dataset, we sent a group of more than 25 individuals to theme parks for three days. Each day a subset of the individuals used a head-mounted GoPro camera to record throughout the day. Our dataset contains more than 42 hours of video recorded by 8 subjects. The group usually broke into smaller groups during the day. As a result, each video contains a significant amount of experiences that are not present in the other videos. The cameras were fixed on caps. The GoPro cameras capture and store a high definition 1280 × 720, 30 fps video. We extract images at 15 fps from resulting in over two million images in total.

We manually labeled the start and end time of intervals corresponding to types of social interactions throughout the videos. We have six labels: dialogue, discussion, monologue, walk dialogue, walk discussion and background. Each of these interactions can take place at a dinner table with group of friends, while walking, or while standing in a line, etc. We train our social interaction detectors on videos from five subjects and test on videos from the remaining three subjects.

During the three days at Disney park, each day one or more interns wore cameras mounted on baseball caps. The first day Alireza Fathi, Mune, Denis Aleshin, Aline Normoyle and Hussain Raza wore cameras. The data for the first day is captured at Coronado Spring Resort, bus, and Magic Kingdom.

In the second day, four interns wore cameras: Alireza Fathi, Michael Sibley, Michael Julian and Matthew Glisson. The data was captured at Epcot. In the middle of the day Alireza Fathi's camera stopped working, and he used Matthew's camera to record the rest of the day. As a result Matthew's data finishes half way through the day. Further, for some unclear reason Michael Julian's data is very short.

In the last day, only Alireza Fathi wore a camera. The data is captured at Disney Hollywood Studios.

5.3.1 Data Annotation

We annotated different parts of the data from different prospects. These annotations contain: Interestingness, Events, Number of Active Users, Social Interactions, Torso Orientation, and where faces look at.

Interestingness: We annotated the interesting moments in the following datasets. Alireza Day1, Alireza Day2, Alireza Day3, Michael Sibley Day2, Aline Day1, Jim Rehg. Alireza Fathi has also annotated most of the videos in Alireza Day1.

Events: We annotates the kinds of events taking place. The set of events we came up with by looking at the videos are the following:

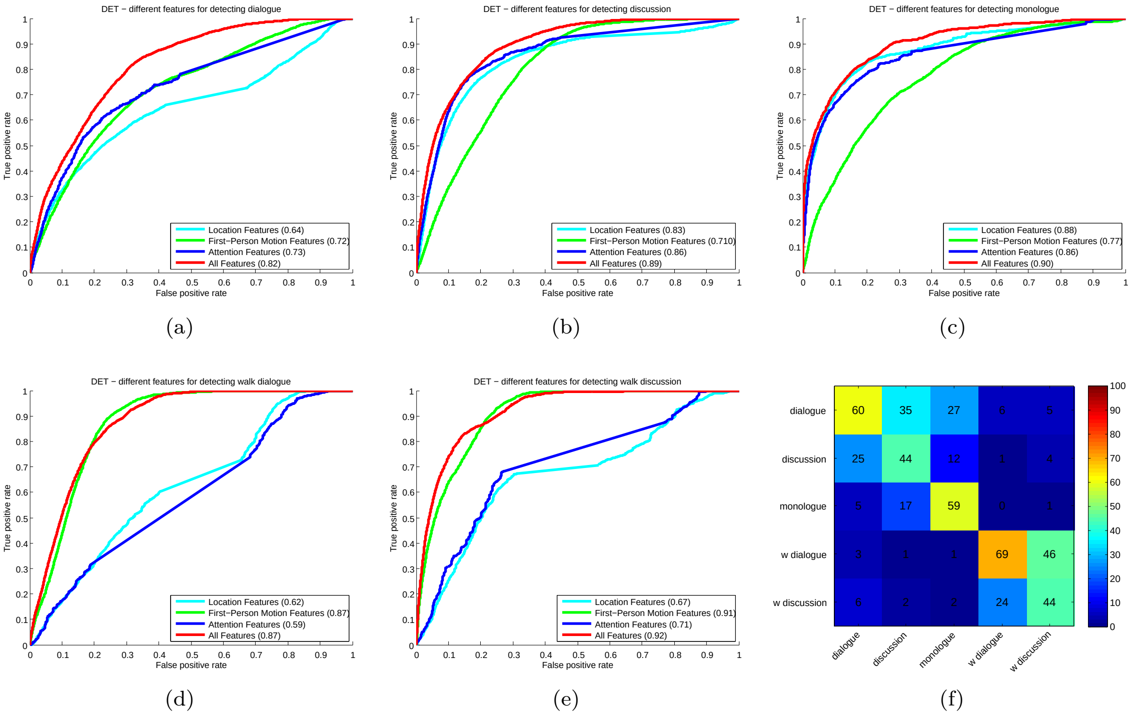
- Walking.
- Waiting in line, either it is waiting for a bus, or waiting to get on a train, or on a ride, etc.
- gathering, which means people, more than 3-4 gather together to talk about something, such as what to do now, whether to go eat, etc.
- sitting in the bus.
- being on a ride in the park such as train, etc.
- buy something from cashier, such as sandwich, drink, ice cream, food, etc.
- having a meal, either alone, or with friends.

Number of Active Users: We annotated how many people including the first person are interacting together. The rule was to only focus on people who are interacting with the first user.

Social Interactions: We annotated some of the videos in Alin Day1 and Alireza Day1 by what social interactions take place. Currently the only labels used are dialogue, monologue and conversation. A dialogue is when the first-person talks to someone else one to one. Conversation is when there is a discussion and multiple people participate. Monologue is when there is one person talking for a significant amount of time and others are only listening.

Torso Orientation: We made a tool in MATLAB that given an image shows a bounding box below each detected face, and asks the user to label it in one of the 8 possible directions. For annotation, 8 circles each of them showing a different orientation are shown and the user clicks on one of them. The 8 orientations are as below: right, 45 degree right front, front, 45 degree left front, left, 45 degree left back, back, 45 degree right back. We annotated some frames from Alin Day1 and some from Alireza Day1.

Where Faces Look: We made a user interface in MATLAB where each face is shown with a circle around it. The user can click on a face and click on a different one to annotate the first face looking at the second face. We annotated some frames from Alin Day1 and some from Alireza Day1. We further setup a camera on a tripod, and have put markers at particular locations on the ground, and has asked some interns to stand at those locations in particular orientations. We use these for calibrating the camera parameters for transferring the faces to 3D.



(a) (b) (c)

(d) (e) (f)

Figure 25: ROC curves of detecting types of social interactions are shown in (a-e). The area under each curve is provided in the figure. In case of dialogue and discussion, the attention features outperform flow and location features. In case of monologue, location features perform the best. First-person motion features significantly outperform the rest in detecting walk dialogue and walk discussion. In addition, we show our recognition results using all features in (f).

5.4 Experiment 6: Detecting Social Interactions

Attention Estimation Results: Example results for face localization and attention estimation are shown in Fig 23. Our method both estimates who is looking at who, and in addition uses the context from the rest of the faces to estimate where in space an individual is attending. For example in Fig 23(c), the group of individuals with red circles around their faces are looking at the lady wearing a white shirt whose face was not detected. Our method realizes that these four individuals are looking at the same location and estimates this location in space. We quantitatively measure the performance of our method. We manually label who each person is looking at in a subset of the frames (about 1000 frames). For each frame, we connect each detected face to the one it is looking at. We split the ground-truth into two sets and use the first set to train the parameters of our model. In 71.4% of the cases our method correctly estimates who is looking at who.

Detection and Recognition of Social Interactions: During training, for each type of social interaction, we randomly select 100 intervals (of 200 frames each) from each subject's video and 300 intervals from the background. As a result, the total number of intervals used for training are

4000. To learn a detector for a particular type of social interaction, we set the label of intervals corresponding to that type to positive and the rest to negative. During the test, we perform the detection on a 200 frame long interval around every frame of the test video. In Fig 25(a-e), we show the performance of our method on detecting different types of social interactions. For each type, we compare the performance of different features. Attention and location based features perform better at detecting dialogue, discussion and monologue, while first-person motion features perform better on walk dialogue and walk discussion. We show that the combination of these features together significantly improves the results for every type of social interaction.

We train a multi-label HCRF model for the recognition of social interactions. In Fig 25(d), we show the confusion matrix for recognizing social interactions. Walk dialogue and walk discussion contain very similar motion patterns and there is a significant confusion between them.

5.5 Other Interestingness Cues

There are cues other than social interactions that are associated with how interesting a moment is. Some of these cues are described below:

Social Relation: Here we show the great potential for building a social network of camera wearer's friends from the daily egocentric videos. We cluster the faces into multiple bins. We manually assign each bin to one of the individuals by looking at the faces it contains. We weigh the connection of a subject (person wearing the camera) to other people based on the number of faces in the cluster corresponding to that individual. The resulting network is illustrated in Fig 26.

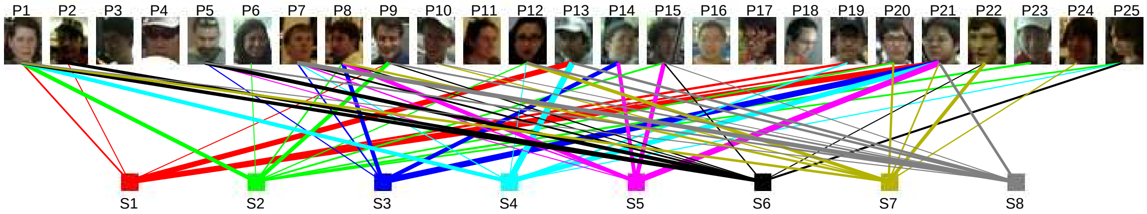


Figure 26: The social network built using our method. The representative faces of persons in the group $P_1 ... P_{25}$ are shown. Subjects wearing the cameras $S_1 ... S_8$ are shown by squares. We weigh the connections based on how frequently a person's face appears in the video captured by a subject. An edge is drawn between a subject and a person if the number of faces are above average. The thickness of edges represent the weight of connections. It is possible to notice some individuals like P_1 who was the tour guide are popular among the subjects. In addition, one can notice the similar connection patterns between S_1 and S_4 who were spending a significant time together throughout the day.

Detecting Head Rotation around Torso: It is useful to understand the relative orientation of the head with respect to the torso, for two reasons: (1) it helps us estimate where people are looking at more accurately and (2) it helps us better realize if a moment is interesting. Sometimes two subjects are sitting parallel and beside each other. In such a case when the two subjects are willing to talk to each other, they usually do not completely rotate their torso. They rather rotate their head but since it is uncomfortable to hold the head 90 degrees around the torso for a long period, they usually rotate their head half way through about 45 degrees. This makes our current algorithm think that the subject is looking at the wrong person, but if we know that the subject's torso is in a different direction, we can make a better judgement. Further people usually do not look behind their shoulder unless an event of their interest takes place.

We are given the head orientation from the software. To find the torso orientation we learn a torso orientation classifier as follows. For each detected face in the image, we automatically crop a windows below the face, rotated based on the pitch of the face. The reason for rotation is that the first person might have her head rotated which causes the camera to be angled from the horizon, and as a result everything appearing rotated in the image. We extract HOG features from these windows, and learn a classifier for 8 different torso orientations (front, left front 45, left, left back 45, back, right back 45, right, right front 45). We use labeled examples to learn a linear SVM classifier for each orientation as positive set and the other orientations as negative. During the test we label the torso with the orientation whose classifier returns the highest score. Some classification results are shown in Figure 27.

Unpredictable Camera Movement: We realized unpredictable camera movement is usually associated with interesting moments such as laughing out loudly, checking all the faces around to share an emotion with them, nodding, etc. At each frame we can compute the camera movement, based on either the average optical flow, or how much the face tracks move on average. We use the second option because computing optical flow is expensive, event though we have run optical flow computation on the cluster but when we came up with this task it was not ready yet.

For the next frame, our assumption is that the camera movement should be similar to that of the previous one. However, this assumption goes wrong in cases such as examples above. We use the absolute difference between the two motion vectors to compute unpredictability. We score each interval by the sum of the unpredictability measures of its frames. We have some video examples for this, but it is hard to show it as images.

5.6 Conclusion and Future Work

We describe a novel approach for detection and recognition of social interactions such as dialogue, discussion, and monologue, in day-long first-person videos. Our method constructs a description of the scene by transferring faces to 3D space and uses the context provided by all the faces to estimate where each person is attending. The patterns of attention are used to assign roles to individuals in the scene. The roles and locations of the individuals are analyzed over time to recognize the social interactions. We believe this is the first work to present a comprehensive framework for analyzing social interactions based on the patterns of attention which are visible in first-person video. We present encouraging results on a challenging new dataset consisting of 42 hours of video captured at a popular amusement park.

Gaze vs. Face Orientation: One of the issues we faced in our method was that face orientation does not always correspond to gaze direction. There is a strong correlation, but it sometimes happens that people move their eyes and look to the sides. We have recently collected some new social interaction data using wearable eye-tracking glasses. In this data, we noticed that first-person's gaze can strongly help to better distinguish between different types of social interaction.

We believe that it is not possible to recognize social interactions solely based on faces and their attention, but there are other important cues such as emotion and voice. As a future work we plan to combine these cues with attentional cues for better understanding the social behavior of individuals in the scene.

Benefits of the Egocentric Vision: In this chapter, our goal was to understand the social experience of a particular person who is wearing the camera. Many social interactions take place in a scene at any given time. However, we are interested in the experience of the person who is wearing the camera throughout the day. The best view for capturing this experience is that person's view.

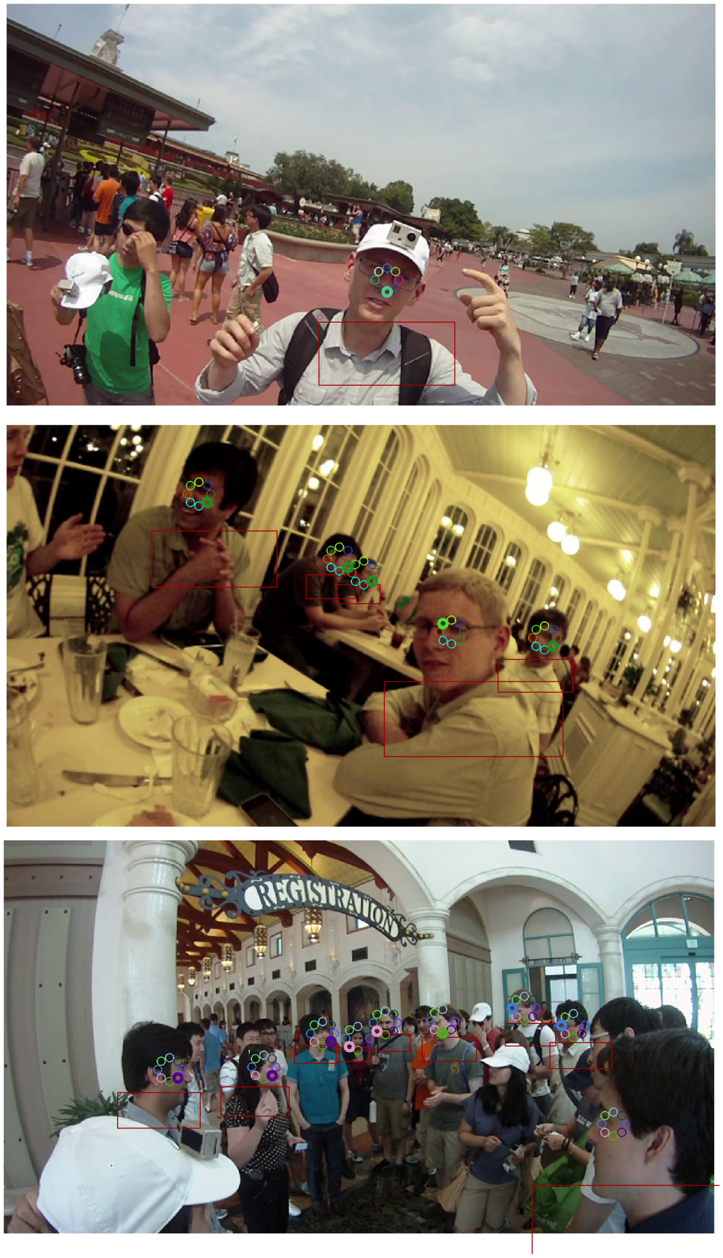


Figure 27: The torso and our classification results. In our experiments we have rotated the torso to align with the head pitch, however, in the images above the torso is not rotated. Each detected face is shown with 8 circles, and the one with which it is labeled is shown bolder. If the person's torso is not seen (out of image) it is not labeled. The circle at the bottom means the torso is looking front, and the one at the top means looking back.

Chapter VI

LEARNING ABOUT OBJECTS IN CONTEXT

In this chapter, our ultimate goal is to automatically learn rich models of object instances from daily egocentric videos. Current approaches to object and activity recognition depend upon large amounts of labeled training data to obtain good performance [104, 40]. In these approaches, labels are often automatically acquired from image tags, movie scripts or close-caption text [27, 119, 17]. However, these annotations are not available for egocentric videos of daily activities.

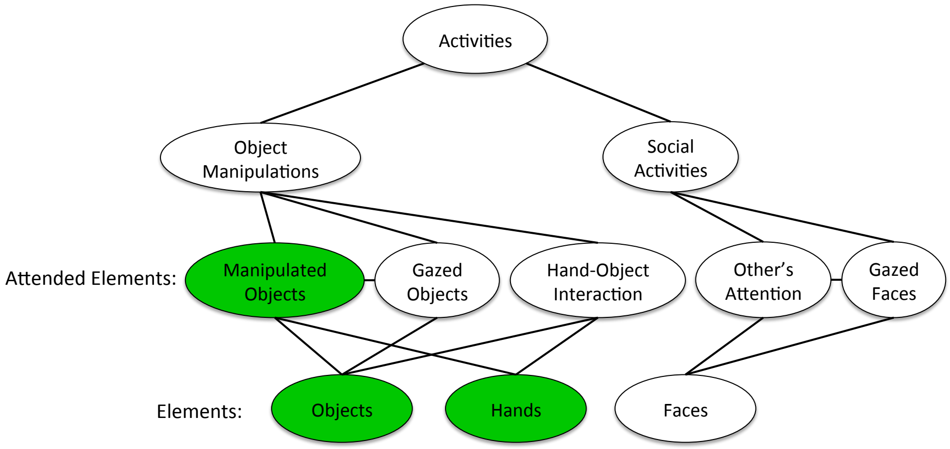


Figure 28: Detecting objects and hands is crucial for recognizing daily object manipulation activities. We develop a semi-supervised method for learning to detect objects in egocentric videos by using the patterns of object co-occurrence in different types of activity.

Here we explore the hypothesis that object instances can be detected and localized simply by exploiting the co-occurrence of objects within and across the labeled activity sequences. This key idea is shown in Figure 29. We assume that we are given a set of training videos which are coarsely labeled with an activity and the list of objects that are employed, but without any object localization information. The difficulty of this learning problem stems from the fact that there are many possible candidate regions which could contain objects of interest, and a standard learning method could succeed only if an extremely large amount of diverse training examples were available.

In contrast to the established third-person video paradigm, the egocentric paradigm makes it possible to easily collect examples of natural human behaviors from a restricted vantage point. The

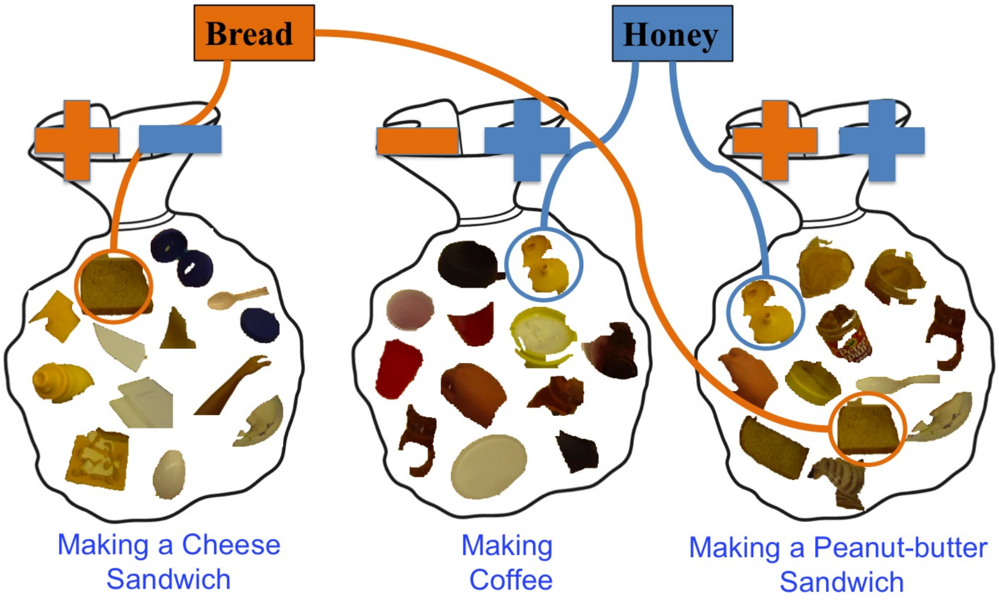


Figure 29: The key idea. Our goal is to discover regions that correspond to an object label (e.g. bread) from the videos that each correspond to an activity. A video can be modeled as a bag of active regions as shown in the image. We use the fact that an object like “bread” is used in some activities and is not used in some other activities. We find regions that consistently appear in the active region of positive activities and do not appear in the negative ones.

stability of activities with respect to the egocentric view is a potentially powerful cue for weakly-supervised learning. Egocentric vision provides many advantages: (1) there is no need to instrument the environment by installing multiple fixed cameras, (2) the object being manipulated is less likely to be occluded by the user’s body, and (3) discriminative object features are often available since manipulated objects tend to occur at the center of the image and at an approximately constant size. Here, we will show how the domain knowledge provided by egocentric vision can be leveraged to build a bottom-up framework for efficient weakly-supervised learning of models for object recognition.

An overview of the method is depicted in Fig 30. Our method consists of two main stages. In the first stage, our goal is to segment the active objects and hands from unimportant background objects. As humans, we can easily differentiate between background objects and the ones we are attending to during the course of an activity. Likewise, the learning method must be able to focus only on the objects being manipulated by the hands in order to be able to accurately distinguish different daily activities. Weakly supervised learning of objects will not be feasible unless we are able to ignore the dozens of unrelated and potentially misleading objects which occur in background. In the second stage of the method, we learn object appearances based on the patterns of object use provided as a weak source of information with the training data. We first use a MIL framework to initialize a few regions corresponding to each object type, and then we propagate the information to other regions using a semi-supervised learning approach. The method and the results of this chapter are previously presented in the following CVPR 2011 paper: Fathi et al. [56].

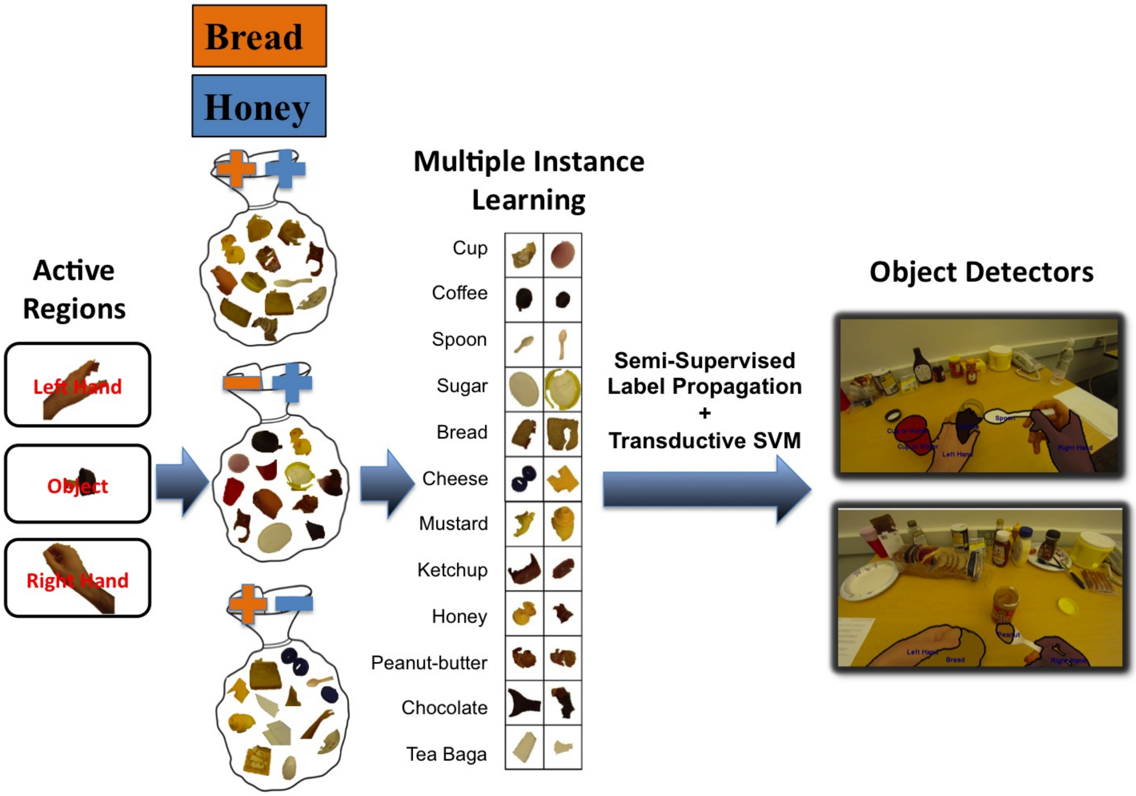


Figure 30: An overview of our weakly supervised object recognition framework is depicted.

6.1 Unsupervised Foreground Segmentation

In this Section we describe a bottom-up segmentation approach which leverages the knowledge from egocentric domain to decompose the video into background, hands and active objects. We first segment the foreground regions containing hands and active objects from background as described in Sec 6.1.1. Then in Sec 6.1.2 we learn a model of hands and separate them from objects, and further refine them into left and right hand areas. In Section 6.2 we show this step is crucial for detection and segmentation of objects in order to parse the activities.

6.1.1 Foreground vs Background Segmentation

Our foreground segmentation method is based on a few assumptions and definitions: (1) we assume the background is static in the world coordinate frame, (2) we define foreground as every entity

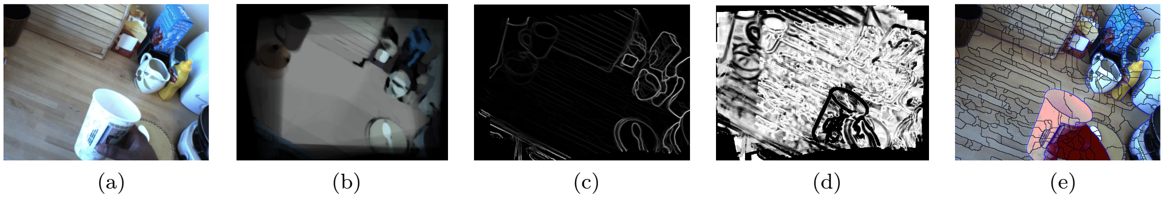


Figure 31: The Background Model. (a) A sample frame from Intel dataset [149], (b) mean color of color-texture background model, (c) mean intensity of background boundary model, (d) the edges corresponding to the object boundary in the sample image do not match the background model, (e) foreground segment is depicted in red.

which is moving with respect to the static background, (3) we assume background objects are usually farther to the camera compared to foreground objects and (4) we assume we can build a panorama of background scene by stitching the background images to each other using an affine transformation. The fourth assumption is basically assuming that the background is roughly on a plane or far enough from camera. An object will be moving with respect to the background when it is being manipulated by hands. When the subject finishes a sub-task and stops manipulating the object, the object will become a part of background again.

Our segmentation method is as follows. We first make an initial estimate of background regions in each image by fitting a fundamental matrix to dense optical flow vectors. We make temporally local panoramas of background given our initial background region estimates. Then we register each image into its local background panorama. The regions in the image which do not match the background scene are likely to be parts of foreground. We connect the regions in sequence of images both spatially and temporally and use graph-cut to split them into foreground and background segments.

We split the video into short intervals and make a couple of local background models for each. The reason is that the background might change over time, for example the subject might finish manipulating an object and leave it on the table, letting it become a part of background. We initially approximately separate foreground and background channels for each image by fitting a fundamental matrix to its optical flow vectors. We compute the flow vectors to its few adjacent frames. For each interval we choose a reference frame whose initial background aligns the best to other frames.

We build two kinds of temporally local models for background (panoramas): (1) a model based on color and texture histogram of regions and (2) a model of region boundaries. To build these models, we fit an affine transformation to the initial background SIFT feature correspondences of each frame in the interval, and the reference frame. We stitch these images using affine transformation. After fixing the images to the reference frame coordinate, we build the color-texture and boundary background models. This is by computing a histogram of values extracted from interval images corresponding to each location in the background panorama. Here we describe these two background models in more details:

Color-Texture Background Model: We segment each image into small super-pixels [8], as shown in Fig 31(e). We represent each super-pixel with a color and texture histogram. We compute texture descriptors [177] for each pixel and quantize them to 256 kmeans centers to produce the texture words. We sample color descriptors for each pixel and quantized them to 128 kmeans centers. We cluster the super-pixels by learning a metric which forces similarity and dissimilarity constraints between initial foreground and background channels. Euclidean distance between super-pixels might not be a good measure, since for example the color of a region on hand might look very similar to a super-pixel corresponding to background. As a result, we learn a metric on super-pixel distance which satisfies the following properties: (1) the distance between two spatially adjacent super-pixels in background is low, (2) the distance between temporally adjacent super-pixels with strong optical flow link is low and (3) the distance between a super-pixel in foreground and a one in background is high. We use the EM framework introduced in Basu et. al [14] to cluster the super-pixels into multiple words using the mentioned similarity and dissimilarity constraints.

We build a histogram of words for each location in the background model from the values that correspond to that location in each interval image. We have depicted the mean color of an example background model in Fig 31(b). Given the computed background model, we estimate the probability of image super-pixels belonging to background by intersecting their color-texture word with the histogram of their corresponding region in background model.

Boundary Background Model: The hierarchical segmentation method of [8] provides a contour significance value for pixels of each image. We transform contour images of each interval to the reference coordinate. For each pixel in the background model we build a histogram of contour values. We have shown average contour intensity for an example image in Fig 31(c). For each super-pixel we measure how well its contour matches to the background model as shown in Fig 31(d). For the super-pixels corresponding to the background, their edges will match the background model with a high probability (some times object edges might create occlusions on background regions), while the super-pixels corresponding to foreground region usually do not match with the background model.

Now that the foreground and background priors are computed for each super-pixel, we connect the super-pixels to each other both spatially and temporally. We connect adjacent super-pixels in each image and set the connection weight based on their boundary significance. We further connect the super-pixels in adjacent frames based on optical flow and SIFT correspondences [114]. We use

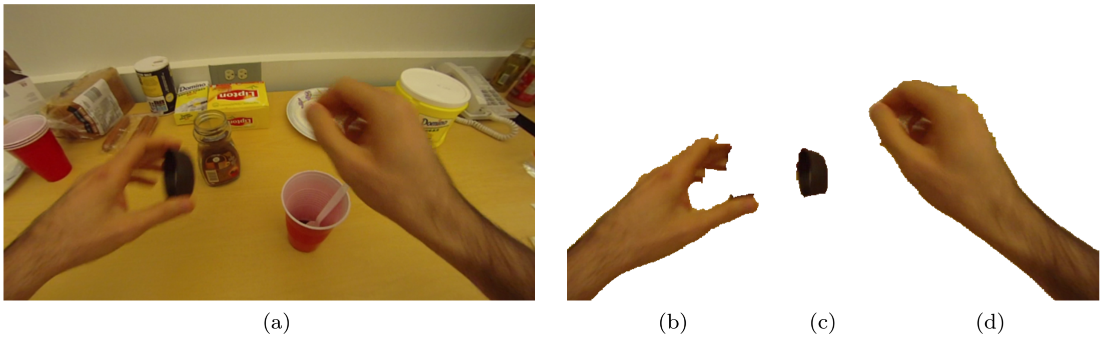


Figure 32: Segmentation results on an image of the Instant Coffee making activity is shown: (a) original image, (b) left hand segment, (c) object segment and (d) right hand segment.

a Markov Random Field model to capture the computed foreground and background priors, as well as spatial and temporal connections between super-pixels. We solve this MRF using graph-cut.

6.1.2 Hands vs Objects Segmentation

We use two pieces of knowledge specific to the egocentric domain to segment the left hand, right hand and the object. We know that the hand presence is dominant in foreground. As objects are manipulated over time, they become a temporary part of foreground, while hands are present most of the time. Given the foreground/background segmentation computed in Section 6.1.1, we build color histograms for foreground and background regions through out each activity.

A super-pixel which has a very high similarity to foreground color histogram is more likely to belong to one of the hands. To set the hand prior for a super-pixel we intersect its color histogram with foreground color histogram and divide it to its intersection score with background color histogram. We set the prior of being object to the median of super-pixel priors for hands. We use graph-cut to segment the foreground into hands and objects.

Given the hand regions extracted in previous step, we segment left and right hands. We use the prior information that in egocentric camera left hand tends to appear in left side of image and right hand usually appears in the right. We set priors on super-pixels based on their location on horizontal axis of image and use graph-cut to segment them into two regions. An example of hand vs objects segmentation is shown in Fig 32.

6.2 Automatic Object Extraction

Given a thousands frame long activity image sequence, our goal is to carve out and label the few participating object categories without having any prior information on their spatial or temporal location. This is a fundamental and challenging problem which is intractable for arbitrary videos given the current state of computer vision technology. However, this problem becomes feasible given the knowledge and constraints existing in egocentric video. Here we describe how our method leverages these information to detect and segment multiple object categories associated with daily activities. The key idea is that each object is used only in a subset of activities and is not included in the rest. An object might be present in the background region of all activity videos, however we use our capability to segment the active object regions to remove the background noise.

The foreground region sometimes contains more than one active object. We split the active object mask into multiple fine regions. Our goal is to learn an appearance model for each object type, and based on that assign each fine region to an object category. To solve this problem, we first initialize each object class by finding a very few set of fine regions corresponding to it. For this purpose, we extend the diverse density based MIL framework of [32] to infer for multiple classes. We need to infer for multiple classes simultaneously in order to discriminate different objects. We further use equality constraints to assign the same object category label to regions with significant temporal connections (with corresponding SIFT feature). These constraints help our method to assign a region to an object class based on the majority votes from its connections.

Given a few regions corresponding to each object class, we propagate these labels to unlabeled foreground regions. Then we learn a classifier for each object class in order to recognize regions in test activities.

6.2.1 Object Initialization

Chen et. al [32] extend ideas from the diverse density framework to solve the MIL problem. Here we further extend their method to (1) handle multiple instance labels simultaneously and (2) apply mutual equality constraints among some instances in each bag. They find a similarity measure between every instance x_i and every bag B_j, using the following equation

$$\begin{aligned} Pr(x_i|B_j) &= s(x_i, B_j) \\ &= max_{x_k \in B_j}\, exp\left(-\frac{\parallel x_i - x_k \parallel^2}{\sigma^2}\right) \end{aligned}$$

where σ is a pre-defined scaling factor. Given m instances in total and l bags, the following $m \times l$ matrix is built:

$$\begin{bmatrix} s(x_1, B_1) & . \; . \; . & s(x_1, B_l) \\ s(x_2, B_1) & & s(x_2, B_l) \\ . & . & . \\ . & . & . \\ . & . & . \\ s(x_m, B_1) & . \; . \; . & s(x_m, B_l) \end{bmatrix} \tag{1}$$

In this matrix each row corresponds to similarities of an instance to all the bags, and each column captures the similarity of each bag to all the instances. For our task, the instances correspond to image regions and bags correspond to activities. Our objective function is to find a sparse set of instances corresponding to each object category which have a high similarity to positive bags and a low similarity to negative bags.

We extend their formulation to infer multiple instance classes simultaneously. Instead of minimizing for a single vector w, we are looking for r sparse vectors w_1, w_2, ..., w_r where each w_c is m dimensional and is positive at a few representative instances of object class c and is zero every where else.

We further add equality constraints $w_c(p) = w_c(q)$ between a pair of elements (p, q) in all $w_c, c = \{1, ..., r\}$ if there is a temporal link between regions corresponding to instances p and q. We optimize the following linear program which minimizes the L1-norm of w_c vectors and returns sparse vectors:

$$min_{w,b,\xi} \left\{ \sum_{c=1}^{r} \mid w_c \mid + C \sum_{j=1}^{l} \xi_j \right\} \tag{2}$$

$$w_c.s(:, j) \geq w_{c'}.s(:, j) + \delta_{c,c',B_j} - \xi_j \; s.t. \; \forall c, c' = \{1, ..., r\}$$

$$w_c(p) = w_c(q) \; s.t. \; \forall c = \{1, ..., r\}, \; (p, q) \in \mathcal{C}$$

$$\xi_j \geq 0 \; s.t. \; \forall j = \{1, ..., l\}$$

where ξ_j is slack variable for bag j, C is a constant, $s(:, j)$ contains the similarity vector of instances to bag j, δ_{c,c',B_j} is 1 if bag B_j is positive for class c and negative for class c' and 0 otherwise, and $\mathcal{C}$ contains the set of equality constraints between instances.

We describe each region with a 32 dimensional feature vector by compressing its color and texture histograms using PCA. This representation is able to both describe objects with and without texture. The similarity between a region x_i and a bag B_j is computed based on the distance between x_i's feature vector and its closest neighbor among all regions in B_j as in Eq 1. We observe that taking region shape and sizes into account enhances the performance. Regions corresponding to different objects might have similar texture and color appearance. For instance, there are white regions corresponding to spoon, sugar and tea bag, but their sizes and shapes are different. To take region shapes and sizes into account, we fit an ellipse to each region and reweigh the computed distances based on the relative ratio of ellipse axis for matched regions. We then optimize the multi-class L1-SVM in Eq 2 to find a few positive instances for each object class.

The feature vectors of the regions of an image are built by assigning the pixels in that image to color and texture bins. Texture descriptors [177] are computed for each pixel and quantized into the 256 nearest k-means centers. Similarly, color descriptors for each pixel are quantized into 128 k-means bins. The color and texture histograms are built from the labels of the pixels in each region. The histograms are normalized to a total sum of one. The two histograms are concatenated to form the feature vector of a region. Then PCA is used to reduce the dimensionality of these vectors to 32.

6.2.2 Object Classification

Our goal is to automatically assign object labels to all foreground regions, while initially we have only a few labeled ones. To do so, we first propagate the labels using the region connectivities in video. For each activity sequence we build a pairwise adjacency matrix W by connecting its regions to their spatial and temporal neighbors. We set the class label of the regions which were initialized in previous step. To expand the label set, we minimize the following objective function

$$E(y) \quad = \quad \frac{1}{2}\sum_{i,j} w_{ij}\delta(y_i - y_j)^2$$

where y_i is the label of region i and w_{ij} is the similarity weight connecting regions i and j in computed adjacency matrix W. We estimate y for unlabeled regions by computing the harmonic function $f = argmin_{f|f_L}E(f)$ as described in Appendix A. Harmonic solution f is a $m \times r$ matrix where m is the number of regions and r is the number of labeled classes and can be computed in polynomial time by simple matrix operations . We fix the label of an unlabeled region i to c, if $f(i,c)$ is greater than $f(i,c')$ for $\forall c' = \{1, ..., r\}$, and further $f(i,c)$ is greater than a threshold.

After expanding the initial labels, we learn a classifier for each object class using Tranductive SVM (TSVM) [83]. To train a classifier for a particular object category, we set the label of its assigned regions to 1, the label of regions in foreground regions of negative bags to -1 and the label

Table 1: Our dataset consists of 7 activities and 16 objects.

Activities	Objects
Hotdog Sandwich	Hotdog, Bread, Mustard, Ketchup
Instant Coffee	Coffee, Water, Cup, Sugar, Spoon
Peanut-butter Sandwich	Peanut-butter, Spoon, Bread, Honey
Jam Sandwich	Jam, Chocolate Syrup, Peanut-butter, Bread
Sweet Tea	Cup, Water, Tea bag, Spoon
Cheese Sandwich	Bread, Cheese, Mayonnaise, Mustard
Coffee and Honey	Coffee, Cup, Water, Spoon, Honey

of regions assigned to other object classes to -1 as well. We set the label of unlabeled regions to 0. TSVM as described in [83], iteratively expands the positive and negative classes until convergence.

6.3 Dataset 1: GTEA

We collect a dataset of 7 daily activities from egocentric point of view performed by 4 subjects. We mount a light GoPro camera on a wearable cap, pointing forward to the area in front of the eyes. The camera is fixed and moves rigidly with the head. The camera captures and stores a high definition 1280 × 720, 30 frame per second 24-bit RGB image sequence. We extract frames with a 15 fps rate from the recorded videos. The total number of frames in the dataset are 31222.

Our dataset contains the following activities: Hotdog Sandwich, Instant Coffee, Peanut-butter Sandwich, Jam and Peanut Sandwich, Sweet Tea, Coffee and Honey, Cheese Sandwich. In Table 1 we have listed the activities and their corresponding objects. We use activities of subjects (2,3,4) as training data to learn object classifiers, and test on the activities of the subject 1. The set of objects appearing in each activity is known for training sequences, while for the test sequence they are unknown.

To validate our object recognition accuracy, we manually assign one object label to each frame of the test activities. In case of more than one foreground object, we assign the label to the object we think is the most salient. We later use these ground-truth annotations to measure our method's performance.

6.4 Experiment 1: Object Discovery

In this section we present results to demonstrate our method's capability in segmentation and labeling of daily activity videos into meaningful pieces.

Segmentation: We compare the accuracy of our foreground background segmentation approach to Ren and Gu [149]. Our method is completely unsupervised while Ren and Gu use an initial ground-truth segmentation set of images to learn priors on hand and object locations, optical flow

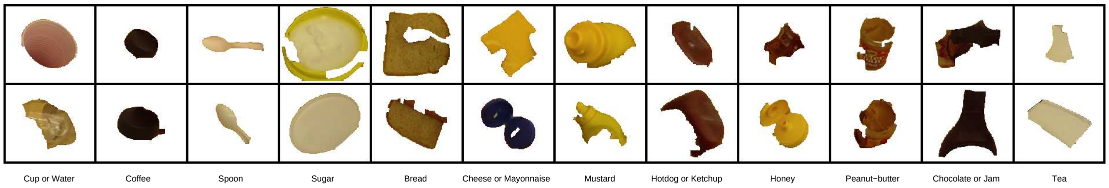


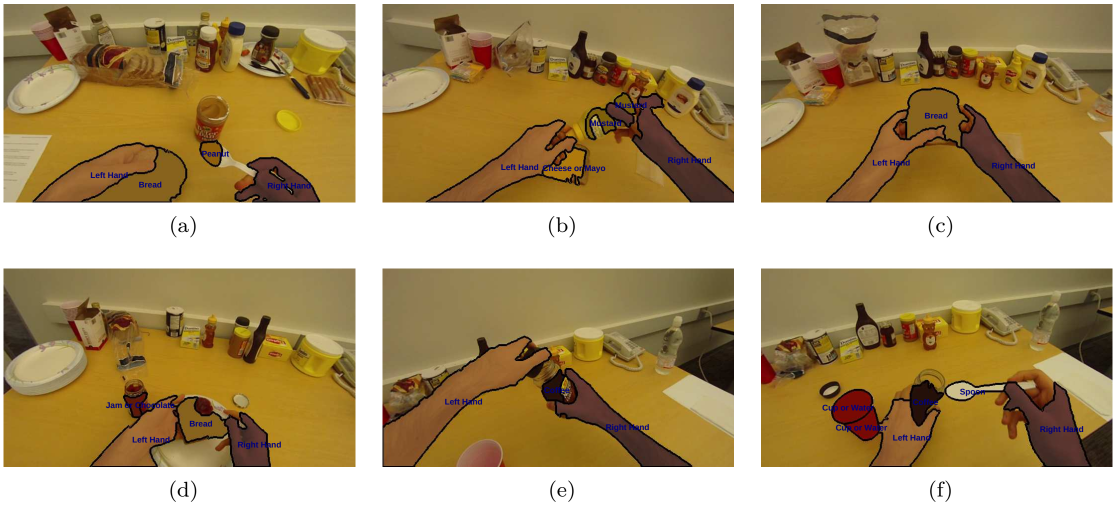
Figure 33: We first automatically initialize a few object regions corresponding to each class as described in Section 6.2.1. Two representative initialized regions are shown for each object category.

magnitude and other features. To compare our results, we manually annotated the foreground segmentations for 1000 frames in the first sequence of Intel dataset introduced in [149] using the interactive segmentation toolkit of Adobe After Effects. Our method achieves 48% segmentation error rate and outperforms their method which results in 67% error. We calculate the segmentation error as follows. We compute the area of the difference between the ground-truth and results. Then we divide this value by the area of the ground-truth. We do this for every image and then average the numbers over all the test images.

Object Recognition: Our method first initializes a few instances of each object using the approach described in Section 6.2.1. We show 2 representative examples for each object category in Fig 33. There are 4 pair of mutually co-occurring object instances. For example, “cup” and “water” always co-occur in our activities. This means that if the cup is used in an activity, water is also used, and if the cup is not used, water is also not used. As a result our method is not able to distinguish between them. We merge each pair into one class which reduces the number of object classes from 16 to 12. This reduction is not applied to the results in the next chapters. This is since in those chapters the action labels are also inferred which help to remove this issue.

We expand the initial object regions and learn a classifier for each object from the training sequences, as described in Section 6.2.2. We test our method on the test sequences as follows. We use our bottom-up segmentation method to automatically segment the active object area in test images. This area might contain more than one active object. For example, during “pouring milk in the cup”, both milk and cup exist in the active region. As a result, instead of assigning a single object label to the active region, we assign an object label to every super-pixel inside the active region. We classify each super-pixel in the active object area using our learned object appearance models. Examples are shown in Fig 34.

In Fig 35 we have shown a few interesting failures. In Fig 35(a), two mistakes are made. The plate that is sitting on the table is segmented as a part of the foreground region, and in addition, it is mislabeled as spoon. In Fig 35(b), a part of the hand is mislabeled as cheese. This is because the hand and the yellow cheese can have a very similar color. In Fig 35(c), a part of the bread



(a) (b) (c)

(d) (e) (f)

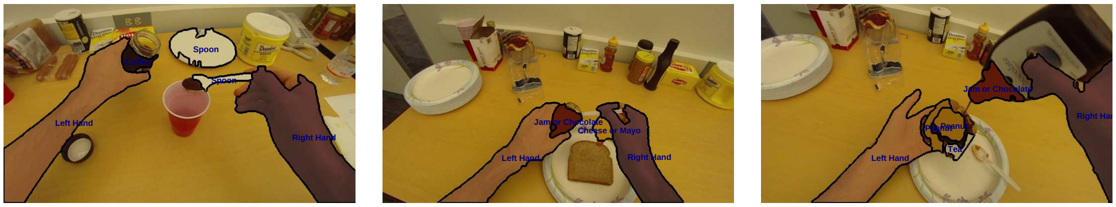
Figure 34: Our method extracts left hand, right hand and active objects at each frame. We learn a classifier for each object class and assign each non-hand region in foreground segment to the class with the highest response.

is mislabeled as tea. Some of these mistakes can be easily avoided by leveraging the context of the action being performed. For example, if the action is “spreading peanut-butter on bread”, it is very unlikely that there is a “tea-bag” in the foreground region. In Chapter 7 we will study these contextual relationships between activities, actions and objects and show that they improve the results.

We compare the labeling accuracy of our algorithm with the ground-truth object labels in Fig 36. The accuracy for predicting some of the objects is relatively high. These are the objects that have a unique color and texture, and have a very unique occurrence pattern in activities: coffee, cup/water, cheese/mayonnaise and peanut-butter. On the other hand, our algorithm results in poor performance on some object classes: spoon, sugar, and tea. Spoon and the tea-bag are small white objects, that can easily get confused with the white wall in the background. In addition, they are confused with other white objects such as mayonnaise, napkin and plate. Sugar also has a similar color to the table in the background which makes the Berkeley segmentation frequently fail in finding its boundaries.

In Fig 37, we show that our learning method is more capable in comparison to a general SVM-based MIL [6].

It is shown that activities can be categorized based on their object use patterns [186]. Segmenting the active object out of background is a crucial step, without which activity comparison based on all the objects appearing in video returns poor results. This is because there are often many objects



(a) (b) (c)

Figure 35: We show a few interesting failures of our method. In (a), the plate is mistakenly segmented as a part of active object regions. The spoon classifier fires on plate region as a result of their similar appearance. In (b), a small region belonging to hand is classified as cheese. In (c), a small part of bread is labeled as tea.

Table 2: We represent each activity based on either a vector that contains the frequency of objects in its foreground regions, or based on a vector that contains the frequency of objects in all areas. In the table, the query activities are shown in the first column. For each query activity, 3 nearest neighbors are found from the training set of activities. This is done once using the vectors built from the foreground regions, and once using the vectors built from all the areas.

Activity	KNN(Active Objects Histogram), k=1,2,3			KNN(All Objects Histogram), k=1,2,3		
Hotdog	**Hotdog**	**Hotdog**	Tea	JamNut	Peanut	**Hotdog**
Coffee	**Coffee**	CofHoney	**Coffee**	CofHoney	Tea	Tea
Peanut	**Peanut**	**Peanut**	JamNut	Cheese	CofHoney	Tea
JamNut	**JamNut**	Peanut	Peanut	Cheese	Cheese	Hotdog
Tea	Coffee	**Tea**	Coffee	**Tea**	Coffee	**Tea**
CofHoney	Coffee	**CofHoney**	Coffee	**CofHoney**	Tea	Tea
Cheese	**Cheese**	**Cheese**	**Cheese**	**Cheese**	Peanut	**Cheese**

sitting in the background that are completely irrelevant to the current task. These objects can confuse any algorithm that is aiming at recognizing the activity. To demonstrate this fact we run a simple experiment. We run our trained object detectors on all the frames of each activity. We represent an activity by a vector containing the frequency of objects detected in its frames. For each activity in the testing set, we use KNN on these vectors to find 3 activities from the training set that best matches it. We perform this experiment once by building the vectors using only the objects detected in the foreground region, and once by using the detected objects in all areas of the frames. In Table 2, we show that the approach which only uses the foreground objects, significantly outperforms the approach that uses all the objects.

6.5 Conclusion and Future Work

In this chapter, we showed that bottom-up foreground segmentation in egocentric vision is surprisingly useful and reliable. We showed that we can segment out the hands from the foreground region and also distinguish between the left and the right hands. Furthermore, we showed that we can discover objects with very weak supervision in egocentric videos. In the rest of this section, we discuss

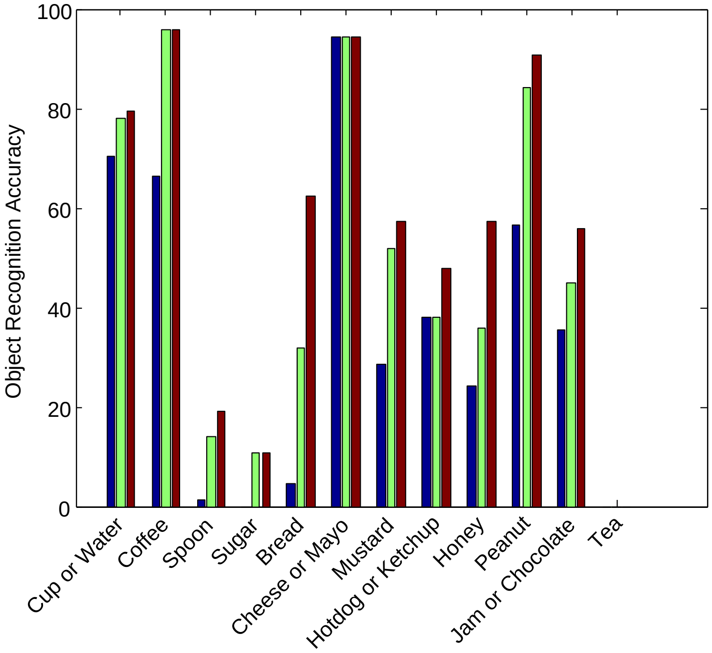


Figure 36: Object recognition accuracy. Random classification chance is 8.33%. Blue bars show how well the highest score detection in each frame matches the ground-truth object label. Green and red bars, depict these results for any of the 2 and 3 highest score detections. We provide these results since there might be more than one active object in a frame but the ground-truth provides only one label per frame.

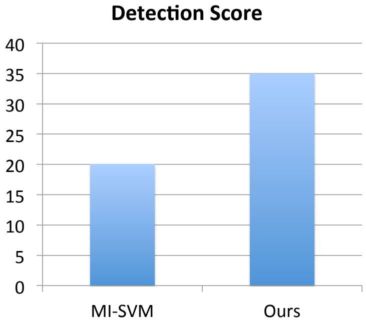


Figure 37: The object regions are sparse in the foreground and each object might contain regions with completely different appearance. As a result an algorithm such as MI-SVM [6] which doesn't take these considerations into account results in lower recognition accuracy. Random accuracy is 8.33%. We compare the results by matching the object labels with the highest detection scores to the ground-truth annotations.

some of the shortcomings and strengths of our method, and then we describe our future plan.

6.5.1 Benefits of Egocentric Vision

Here we try to answer the benefits of the fact that our videos were captured from egocentric view at different stages of our algorithm.

Overall: Our goal is to discover objects by watching first-person's interactions with the world. A person might interact with objects in the street, in the bus, at home and in the office. It is impossible to have enough static cameras that can capture all these moments. In addition, subject's body will frequently occlude the objects in static views.

Segmentation of hands and active regions: Our method assumes that the hands and the active objects move independently with respect to the static background. We use motion as the main feature to segment the hands and active regions. In case of a static camera, a simpler background subtraction technique can be used to distinguish first-person's body and the active objects from the background. It won't be able to distinguish subject's hands from the rest of the body.

In addition, the two cues that we used for separation of hands from active regions are not valid for the static setting. These two cues are as follows: (1) first-person's hands are the dominant part of the foreground and we can learn a color model for them; and (2) there is a location prior on where left hand, right hand and the active objects appear in egocentric video. In case of a static camera, we cannot assume that the first-person is the only subject who will appear in the video. As a result, the assumption that we can learn a color model for first-person's body is not valid anymore. Furthermore, there is no prior on the location at which first-person will appear in the static camera's view.

Weakly supervised object discovery: In our algorithm we leveraged the occurrence patterns of objects in activities to discover their labels. In case of static view, there is not guarantee that the camera will capture all the stages of an activity. Thus, this method will not be effective. For example, during the activity of "making coffee", the sugar might be in the other room. Then the static camera will not capture the usage of sugar. As a result, the MIL framework for finding the sugar will fail.

6.5.2 Scalability

Here we discuss an important question that is "how well the algorithm in this chapter scales to hundreds or thousands of objects?". The bottom-up segmentation approach is computationally expensive. However, we have implemented a version in OpenCV that uses GPU and runs only 10-20 times slower than real-time. The MIL stage is the part that might fail as the number of objects increase significantly. We are currently representing the regions with simple color and texture features, and still we are able to discriminate objects. This will most likely fail to generalize for thousands of different object instances. The solution is obviously to make the features more complicated. However, this might cause problems to the MIL framework.

6.5.3 Discussion

Here we discuss some of the characteristics of our algorithm.

Instance vs. Category: our algorithm is able to discover the instances of the objects, and not the categories. Discovering object categories with weak supervision is a much harder problem. However, since often individuals tend to use the same instance of an object type in their daily activities, our algorithm should still suffice to a large extent. Once object instances are discovered using our method from an individual's life, one can merge different instances of an object type from various people's data to learn models for object categories.

Things vs. Stuff: our object discovery method might discover regions that correspond to things (e.g. coffee jar, spoon, hands, etc.) and at the same time might discover regions that correspond to stuff (e.g. coffee powder in spoon, salt, milk, etc.). This can be both an advantage of our algorithm and at the same time a disadvantage. Being able to recognize stuff is very helpful for recognizing daily actions. For example, in order to recognize the action of "scooping coffee", it is crucial to be able to detect the region that correspond to coffee powder in spoon. On the other hand, this can be a disadvantage, since our algorithm cannot tell the difference between coffee jar and coffee powder.

The usage of regions instead of interest points or bounding boxes is very helpful in recognizing materials (stuff). Materials can from into any shape (e.g. milk can take the shape of its container). As a result, gradient-based features and bounding boxes are not effective for recognizing them. In Chapter 8, we will show that it is possible to discover regions that correspond to the state of the materials or objects from the actions given a very weak amount of supervision.

Features: We believe that color and texture representations for super-pixels are not discriminative enough for distinguishing hundreds or thousands of objects from each other. Different objects can have regions with very similar color and texture. However, on the other hand, this simple feature space makes it practical to discover objects with a weak amount of supervision. Our intuition is that weakly supervised methods often work only if nearest neighbor on the feature space leads to the instances of the same category. This does not necessarily hold in case of complicated gradient features with hundreds of dimensions.

In Chapter 9, we represent super-pixels with features that take the neighborhood context into account. For example, the lid of the peanut-butter jar can be red, and at the same time there can be a very similar red region on the cereal box. However, once put in context, each of these will have different regions surrounding them. The vectors corresponding to this feature representation become longer and more complex. However, since the training approach in Chapter 9 is fully supervised, it can learn the appropriate metrics for this more complex feature space.

6.5.4 Future Direction

We would like to extend this work in two directions in the future. First, we would like to discover objects using much less supervision. We do not think it is always realistic to have access to the label of objects that are used in an activity. As a result, our goal is to develop a completely unsupervised technique for discovering objects by watching an individual's daily activities. We would like to be able to discover objects only based on the fact that they might be grasped by first-person's hands, or based on the fact that their location in the scene might change. We think depth information is crucial for reaching this goal. Thus, our plan is to use a RGB-D wearable camera for learning the 3D models of objects by watching the interactions of the first-person with the environment. The 3D models of objects results in a better understanding of the functions and the usages of objects in daily activities.

Chapter VII

OBJECTS, ACTIONS AND ACTIVITIES: CLOSING THE LOOP

Having detected hands and active objects using our method in Chapter 6, in this chapter we introduce a method for recognizing daily object-manipulation activities based on the interaction between hands and objects (Fig 38). We believe that the egocentric paradigm is particularly beneficial for analyzing activities that involve object manipulation, for three reasons: First, occlusions of manipulated objects tend to be minimized as the workspace containing the objects is always visible to the camera. Second, objects tend to be presented at consistent viewing directions with respect to the egocentric camera, because the poses and displacements of manipulated objects are consistent in workspace coordinates. Third, actions and objects tend to appear in the center of the image and are usually in focus, resulting in high quality image measurements for the areas of interest.

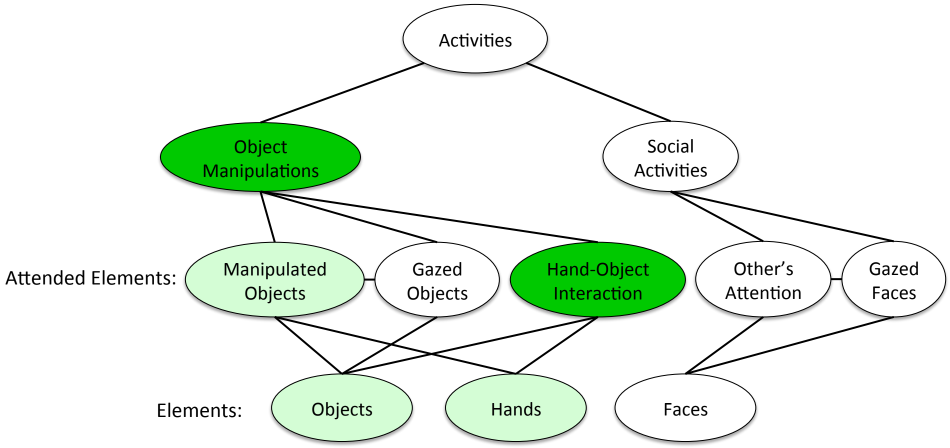


Figure 38: We introduce a method for recognizing daily object manipulation activities based on the interaction between hands and objects. The green nodes in the figure show the focus of this chapter. The light green nodes are detected using methods introduced in previous chapters, result of which are used in this chapter for object manipulation activity recognition.

Most day-to-day activities consist of actions that involve manipulating objects like pouring water into a cup, opening a peanut-butter jar, etc. Interactions between objects and hands contain important discriminative cues for action recognition. This suggests representing actions by objects and their interactions with hands. This approach is in contrast to traditional action recognition methods where body configurations and movements are the main features.

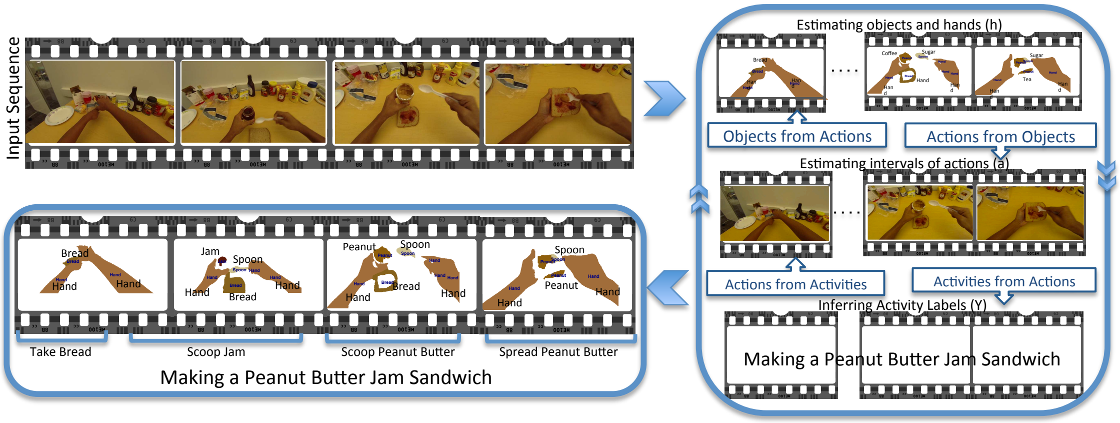


Figure 39: An overview of our approach.

A key aspect of our approach is the use of the semantic relationships between activities, actions, and objects to prune the search space arising in video interpretation. We show an overview of our framework in Fig 39. For example, the ability to detect objects being manipulated reduces the space of actions under consideration to those which are consistent with the object's affordances. Object identity also constrains the space of possible actions. For example, knowing that we are manipulating a cup rules out actions corresponding to making a sandwich and make actions corresponding to making coffee more probable. We are therefore interested in visual feature representations and classifiers which support the incorporation of such additional domain knowledge in the decision process.

In our approach, actions are represented as relations between objects and hands. Hand features, such as proximity and directionality of movement, are defined in an object-centric coordinate frame, thereby capturing the key properties of object manipulation. Activities are then modeled as a set of temporally-consistent actions. For example, an activity like making a peanut butter sandwich starts with taking a slice of bread and then opening a jar of peanut butter, followed by scooping peanut butter out of the jar and onto the bread. Our constraints on actions capture the fact that peanut butter cannot be scooped from the jar before it is opened. A key barrier to the use of semantic information in activity recognition is the need to hand-label object and actions across a large video corpus to support supervised learning. We address this issue by leveraging our method from Chapter 6 on unsupervised learning of object models from egocentric videos. We augment automatically-learned object models with weak annotations of actions to complete the training data for an activity model. We present a fully-automatic method for learning activity models from such weakly-labeled data.

Here we make three contributions: 1) We present a novel representation for egocentric actions based on hand-object interactions. 2) We develop a novel approach for automatically constructing a joint model of activities, actions and objects, in which the context provided by each element enhances the ability to recognize the others. 3) We provide experimental evaluations on an egocentric test bed and demonstrate benefits of joint modeling of actions, activities, and objects comparing to independent models.

We demonstrate the advantages of our semantic representations of actions and activities in comparison to state of the art feature based representation. The method in this chapter has appeared in the following ICCV 2011 paper: Fathi et al. [50].

7.1 A Model of Daily Activities

Our task is to analyze an image sequence of a person performing an activity like making a tuna sandwich. This entails inferring the activity label, segmenting the activity to a series of consecutive actions, and assigning object and hand labels to image regions in each frame. As a result, each input sequence contains a set of intervals $\mathbf{v} = \{u_1, ..., u_U\}$, where each interval u_i consists of F_i images $u_i = \{I_1, ..., I_{F_i}\}$, and each image I_j consists of m_j super-pixels. Throughout the chapter, we refer to super-pixels as regions. Each super-pixel is represented with a multi-channel feature vector x_i, that includes color, texture and shape.

Inference involves assigning an activity label y to each sequence $\mathbf{v} = \{u_1, ..., u_F\}$, an action label a_i to each interval u_i, and an object, hand or background label h_j to each super-pixel x_j. Each y is a member of a set of possible activity labels, for example, $\mathcal{Y} = \{$making a peanut-butter sandwich, making a cheese sandwich, making coffee, etc$\}$. Each a_i is a member of a set of possible actions $\mathcal{A} = \{$pour water into cup, spread peanut-butter on the bread, etc$\}$. Each action consists of a verb (e.g. pick, pour, open, etc) and a set of object names (e.g. water, cup, bread, peanut-butter, hand, background, etc). Finally each super-pixel is assigned an object or hand label from the set $\mathcal{H} = \{$hand, cup, coffee, bread, water-bottle, etc$\}$.

We believe that objects, actions and activities should interact. We model this interaction by the graphical model depicted in Fig 40. Action labels interact with activity, object and hand labels. During training we observe action and activity labels and have access to weak labels of objects. During inference we only observe features from superpixels and infer the object labels, action intervals and activity labels.

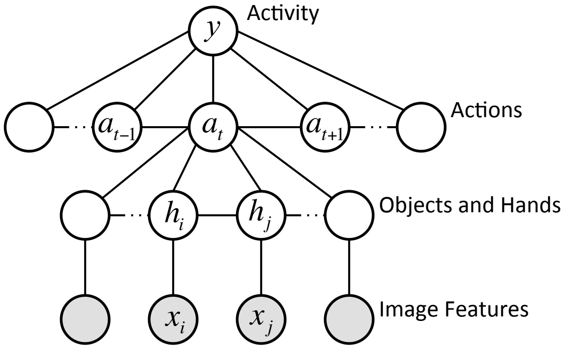


Figure 40: Our framework's model. During the testing phase, an activity label y is assigned to each given video. Action labels a_t are assigned to every frame t of a video. In each frame, an object label h_i is assigned to every super-pixel i given its feature vector of color, texture and shape x_i.

7.2 Learning and Inference

To setup the notation, given a set of training sequences $\mathbf{x}^{(n)}$ and labels $\{y^{(n)}, \mathbf{h}^{(n)}, \mathbf{a}^{(n)}\}$, our task is to build a model, that given a new sequence $\mathbf{x}^{(n)}$ produces the true set of activity, action and object labels $\{y^*, \mathbf{h}^*, \mathbf{a}^*\} = \{y^{(n)}, \mathbf{h}^{(n)}, \mathbf{a}^{(n)}\}$ as shown in Fig 24. Exact inference on such a complex graphical model is impractical. As an alternative, we propose to exploit the independence structure of the domain and factor this model into the following four interacting modules:

1. Learning to predict the intervals of actions based on hand-object interactions (a_t given $\{h_i, h_j, ...\}$).

2. Learning to classify activities using an action based representations of activities (y given $\{..., a_t, a_{t+1}, ...\}$).

3. Learning to modify intervals of actions given the activity labels, as well as hand and object interactions (a_t given y).

4. Learning to modify estimates of objects and hands given the action labels (h_i given a_t).

Our procedure is depicted in Fig 39. We start by initial estimates of appearance models for objects and hands in a weakly supervised setting. We use the multiple instance learning approach presented in Chapter 6 to provide initial object, hand and background models. We then obtain the action labels and use them to infer activity labels. Once we fixed the activity labels, we modify the actions accordingly. Having finalized action intervals we update our estimates of objects and hands. Our approach is similar to Expectation-Conditional Maximization [120] where the M step in the EM is replaced by conditional maximization steps. Below we describe each of these modules.

7.2.1 Learning Actions based on Object Interactions

This module estimates a discriminative score for assigning an action label a to an interval containing a vector of regions $\mathbf{x} = \{x_1, ..., x_m\}$, each of which are assigned an object or hand label h_i. This is a sub-problem of the original task which can be modeled by removing the top level of the graphical model (activities) as well as the connection between adjacent actions.

We want to learn a discriminative function $f_{h \to a}(a, \mathbf{h}, \mathbf{x})$, which returns a real number for any action assignment a to an interval consisting of image regions $\mathbf{x}$ and object labels $\mathbf{h}$. We initialize $\mathbf{h}$, which is the initial object label assignments to regions using the classifiers learned from the weakly supervised object recognition. We extract object and hand interaction features and learn $f_{h \to a}(a, \mathbf{h}, \mathbf{x})$ by training a discriminative classifier on those features for each action class a. We use Adaboost [153] for classification. In contrast to the popular interest point features used for action recognition, our features are capable of capturing the semantics of interactions in the scene. Here we describe the set of object and hand interaction features used in our system.

Object Frequency (f_1): contains the histogram of object labels (hand and background labels are included as well).

Object Optical Flow (f_2): we compute the average optical flow vector for each region. The vector of each region is discretized based on its orientation and magnitude.

Object Relative Location (f_3): we build an adjacency matrix for the regions. We quantize the relative location of the center of adjacent regions into bins. For every pair of object classes we compute the histogram of their relative location bins in the interval. We reduce the dimension using PCA.

Object Classification Score (f_4): sum of the classification scores for the regions assigned to each object type are concatenated to build this feature vector.

Object Pose (f_5): for each region we compute the pose based on its shape. We build a shape descriptor as a set of annular sections (similar to shape context), each of which can be thought of as a bin. We assign each bin's value to the total number of region pixels falling in that bin.

Hand Optical Flow (f_6): these features are similar to f_2. One is computed for left hand and one for right hand.

Hand Pose (f_7): similar to f_5, one for each hand.

Hand Location in Image (f_8): We split each image into multiple regions using horizontal and vertical cutting lines. We assign the number of left/right hand pixels falling in each region as its value.

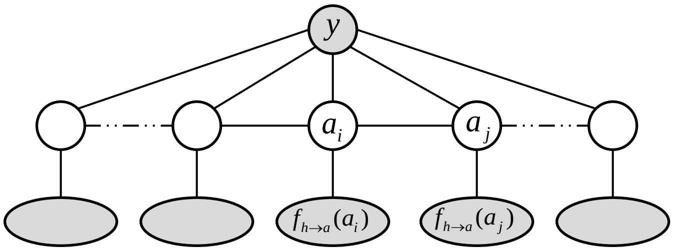


Figure 41: Model showing the decomposition of activities into actions. We refine the actions given the estimated activity label and action classification scores computed in the first stage of the algorithm.

Hand Size (f_9): The area of each hand in pixels.

Left/Right Hand Relative Location (f_{10}): for each image, if there are two hands in the image, we find their pair of closest points. We use their relative x and y distance as features. We concatenate these with the relative x and y location of the center of mass of the hands.

In Sec 7.3.1 we evaluate the performance of these features in the recognition of various action classes. Since there is a large number of actions (64) in our experiments, we break action recognition into two steps. We first estimate the action verbs (e.g. pour, dip, pick, etc) and then in the second step we estimate the object set. In the second step we use a probabilistic model on action verb label and object set classification scores. We learn classifiers for each object set using the same set of features mentioned above. We apply a probabilistic model to infer the object set given action verb label and object set scores.

7.2.2 Learning Activities from Actions

Given the set of action labels $\mathbf{a}$ we want to estimate the activity label y. We want to learn a discriminative function $f_{\mathbf{a}\to y}(\mathbf{a}, y)$ that receives the set of actions $\mathbf{a}$ in a sequence and an activity label y and returns a real number. We learn a classifier for each activity y given a histogram of action classes. We use Adaboost algorithm to learn the classifiers. During the test we build the feature vector from the action classes computed in previous stage. We assign the activity label with the highest classification score to the sequence.

7.2.3 Learning Actions from Activities

After fixing the activity label, we go back and enhance the action recognition results. Knowing the activity not only limits the set of possible actions, but also forces the appropriate ordering of actions. For example, pouring water is not an expected action in the making peanut-butter sandwich activity. Further, opening the peanut-butter is expected to happen before scooping peanut butter and spreading it on the bread. Given an activity label y assigned to a video and scores $f_{h\to a}$

computed in first stage, we want to assign action labels $\mathbf{a}$ to sub-intervals inside the sequence. A Conditional Random Field (CRF) chain [97] model (shown in Figure 41) is learned for every activity label y on action scores and the transition potentials between actions:

$$\begin{aligned}\arg\max_{\mathbf{a}} \quad & \sum_{(i,j)\in\mathcal{E}^{act}} \mathbf{w}_{y,a-a}^{\top}\upsilon(y, a_i, a_j) \\ & + \sum_{i\in\mathcal{V}^{act}} \mathbf{w}_{a,f}^{\top} f_{h\to a}(a_i, \mathbf{h}, \mathbf{x})\end{aligned}$$

where $\mathbf{w}$ are weight vectors, $\upsilon(y, a_i, a_j)$ models the transition between adjacent actions a_i and a_j given activity label y, and $f_{h\to a}(a_i, \mathbf{h}, \mathbf{x})$ is the classification score of action a_i based on hand and object interactions computed in Sec 7.2.1. The parameters of the model are optimized using the quasi-Newton algorithm. During inference, the Viterbi algorithm is used to assign action labels.

Our focus in this chapter is not on recognizing parallel actions and story telling [69, 20] or modeling temporal logical relations between intervals [5]. Instead, we focus on modeling the ordering between adjacent actions and the contextual relations between activities and actions.

7.2.4 Object Recognition using Action Context

We learn a probabilistic model for the objects given actions. The inputs to this model are object classification scores $\phi(x_i)$ of a region and the action label a. Output is the probability of each object label being assigned to the region x_i:

$$P(h_i|a, \phi(x_i)) \propto P(h_i|\phi(x_i))P(h_i|a)$$

We estimate $P(h_i|\phi(x_i))$ from learned classifiers on region appearance models and compute $P(h_i|a)$ from our training set.

Here we describe the method for computing object classification scores. The image annotations of the regions are unknown during both the training and testing phases. We are only provided with weak information on patterns of object-use in actions during training. For each action in training, we are given a verb and a set of nouns, corresponding to the objects used in that action. As a result, we know about the set of objects that are manipulated in action intervals. We use semi-supervised learning framework from Chapter 6 to build object classifiers given these weak informations.

In each interval, various objects might appear in the background. To only focus on objects being manipulated by hands, we segment the foreground from the background. Each foreground region contains hands, and might contain multiple regions corresponding to one or more objects.

For example, in the action "scooping coffee into cup using spoon", the objects "spoon", "coffee" and "cup" might appear in the foreground simultaneously. We use a multi-class MIL framework to initialize a few regions belonging to each object class, by using the actions as positive bags for the set of their manipulated objects and negative bags for other objects. We expand these regions using the semi-supervised learning technique described in Appendix A and learn object classifiers using transductive SVM [83].

7.3 Experiment 2: Object, Action and Activity Recognition

In the Experiment 1 in Chapter 6, we evaluated the performance of our weakly supervised object recognition framework. Here in Experiment 2, we demonstrate the effectiveness of joint inference of objects, actions and activities in egocentric videos. We present three sets of results to validate the performance of our method at its different stages: (1) object recognition, (2) action recognition and (3) activity recognition. We further analyze the performance of our semantic features for the task of action recognition.

We test our method on the GTEA (Georgia Tech Egocentric Activities) dataset. This dataset contains the ground-truth action labels for activities. Each action label consists of 4 variables: (1) a verb (e.g. take, open, pour); (2) a list of nouns corresponding to the objects that participate in the action (e.g. cup, spoon, bread); (3) a frame number at which the action starts; and (4) a frame number at which the action terminates. There are 64 action types (a unique set of verb and nouns) in the dataset, consisting of 11 unique verbs (Fig 42) and 16 object instances. This dataset contains 7 kinds of daily activities recorded from a head-mounted camera as they were performed by 4 subjects. The duration of each activity is about 1200 frames recorded at 15 fps. Here we use the activities performed by subjects 1, 3 and 4 for training, and use the activities performed by subject 2 for testing.

7.3.1 Action and Activity Recognition

We perform action recognition for every individual frame. We use features that capture the interaction of objects and hands as shown in Table 3. These features are described in detail in Sec 7.2.1. For each of the features, we learn multiple binary classifiers (one for each action verb class) using the Adaboost algorithm [153]. We tried SVM as well, but we achieved better results using Adaboost. During the test we return the action class with the highest score as the action label. We have compared the accuracy of our features for different classes in Table 3. Our main observation is that for each action class, there is a different feature that better captures its characteristics. For

Table 3: Classification accuracy of each feature on different action classes are shown (in percentages). The feature numbers are listed in the first column. The features are as follows: object frequency ($f1$), object optical flow (f_2), object relative location (f_3), object classification score (f_4), object pose (f_5), hand optical flow (f_6), hand pose (f_7), hand location in image (f_8), hand size (f_9), right/left hand relative location (f_{10}). Detailed description of these features can be found in Section 7.2.1. In the second column, the feature dimensions are shown. Third column contains the average accuracy of each feature over all the actions. The rest of the columns show the accuracy of the features on each of the actions. Tenth column shows the accuracy of the features on discriminating background class (non of the actions) from the other action classes.

Feature	Dim	Acc	Pick	Open	Scoop	Close	Pour	Stir	Bg	Spread	Put	Fold	Dip
f_1	18	27.4	34	15	14	11	40	1	38	31	3	**9**	1
f_2	18×8	31.7	25	15	18	**17**	**61**	5	41	**48**	3	3	3
f_3	$18 \times 18 \times 4$	25.2	30	11	11	9	34	3	41	23	2	4	3
f_4	18	25.2	38	12	**25**	9	37	8	35	9	4	2	3
f_5	18×4	26.9	31	11	10	11	39	1	45	21	6	1	0
f_6	2×8	30.8	20	25	0	10	54	0	**55**	25	0	0	0
f_7	2×8	**34.1**	**71**	29	6	12	51	5	28	24	4	0	25
f_8	$2 \times 3 \times 3$	**39.8**	45	38	23	**17**	8	2	44	22	1	0	**45**
f_9	2	25.8	33	**40**	0	7	17	23	43	3	0	0	14
f_{10}	4	26	4	39	7	3	32	**24**	50	19	**9**	0	2

example, during the action *pick*, the subjects always extend their hand to the end of table to take an object. As a result the hand shape discriminates this action class the best. Furthermore, we observe that the feature vector extracted from the hand location in image performs the best on average. This is a benefit of egocentric footage. The second best performing feature is the hand pose. If the camera was not mounted on the head we weren't able to acquire high resolution images of hands to extract the hand pose.

Since our features have semantic meaning, we can come up with interesting interpretations for how each feature should perform on each action class. In general, features based on the hand pose and hand location perform the best. While in traditional action recognition, the location of hands in the image is considered as a mis-leading feature, it performs the best in our domain because the Egocentric action is always recorded from the same vantage point.

For each frame, we concatenate the following features (f_2, f_6, f_7, f_8, f_9, f_{10}) to make our action representation feature vector (adding more features does not enhance the performance). We learn our action classifiers using 200 iterations of Adaboost algorithm. We compare the performance of our features with STIP and SIFT bag of words. In Fig 42 we show that our semantic representations of objects and hands provides a significant boost in recognition accuracy in comparison to widely used features like STIP and SIFT bag of words. Our features perform frame-based action recognition with 45% accuracy, while STIP performs with 14.4% and SIFT performs with 29.1% accuracy. It is interesting that SIFT bag of word features perform better than STIP. We believe this is because

(1) in daily activities objects play a discriminative role in recognizing actions, (2) the same action can produce a variety of different movement patterns (imagine all the different ways one can close a water-bottle, e.g. hold with left hand and close with right hand, do it only with right hand, etc). To build the bag of word for SIFT and STIP features, we cluster them using Affinity Propagation [61]. We tune the number of words to achieve the best result.

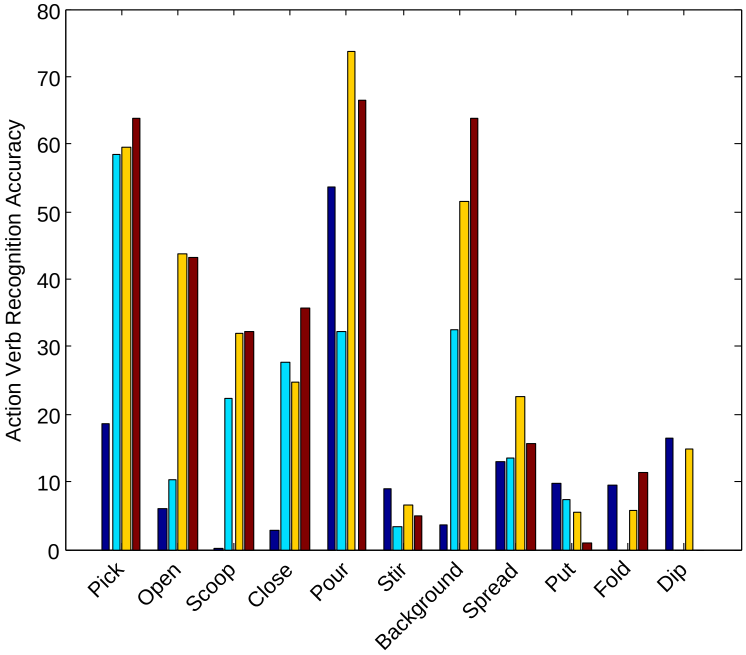


Figure 42: Action verb recognition results are compared between different methods: STIP [103] bag of words (blue), SIFT [114] bag of words (cyan), our features (yellow) and actions classification enhanced by our method after predicting the activity class (red). The total accuracy of different methods are as follows: STIP (14.4%); SIFT (29.1%); Adaboost classifier learned from the combination of our hand-object interaction features (45%); and further refinement using the context of the activity (47.7%). There are 11 action verb classes which means the random classification accuracy is 9.1%. An interesting observation is that, since our method classifies the *making tea* activity as *making coffee*, it fails to recognize the *dipping* the tea-bag action in that sequence.

To recognize the activities from the predicted actions, we learn a classifier on the histogram of action frequencies for each sequence. We learn multiple binary classifiers using Adaboost with 10 iterations. We trained a SVM classifier also, but the Adaboost classifier performed better. We can recognize 6 out of the 7 testing phase activities correctly. The only mistake is made by classifying *making tea* as *making coffee.* These two are very similar activities and contain very similar objects and actions.

We further compare our method which encodes interactions between activities, actions, and objects to the case of considering them independently. In our method, given the computed activity

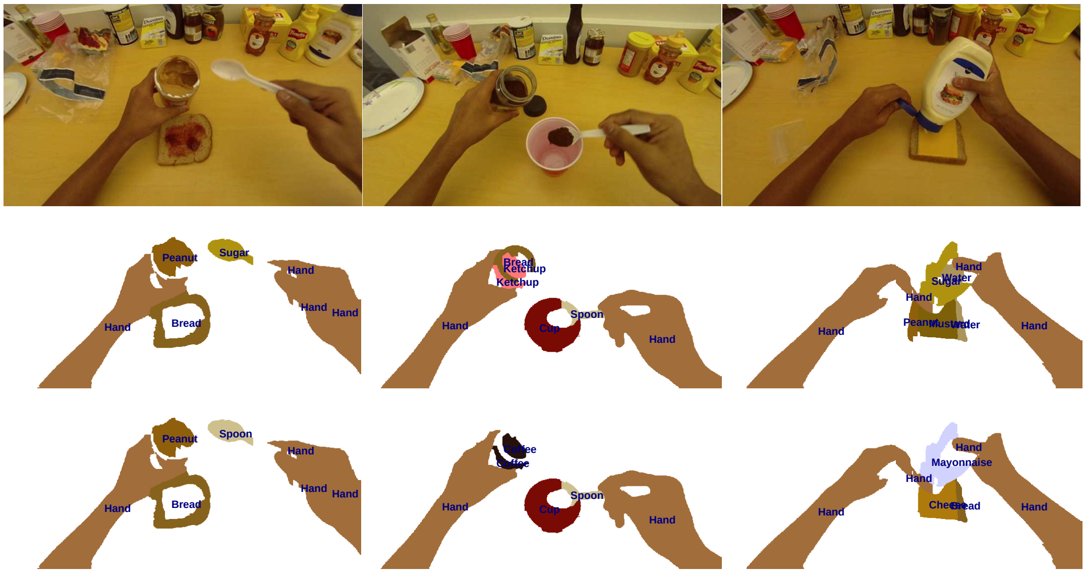


Figure 43: We show that object recognition accuracy is improved given the action as context. The images are shown in the first row. Object recognition results are shown in second row only based on object classification scores. In the third row, we show that knowing the action improves object recognition using our method in Sec 7.2.4.

label, the action classification scores are given to a CRF model built for that activity. We use Viterbi algorithm to infer the action classes. Even though we had mis-classified the *making tea* activity as *making coffee* (1 mistake out of 7 activities), the action recognition results are improved (47.7%) in comparison to using the hand and object features alone (45%) as shown in Fig 42.

Our final recognition accuracy for the 64 classes is 32.4% compared to 4.8% for STIP and 11.6% for SIFT bag of words. Note that we are classifying every frame and chance in a 64-class classification problem is 1.6%.

7.3.2 Object Recognition

Object recognition is improved if the action is known. For example, if the action is “pouring”, then “spoon” is not in the possible set of objects. We compare the object classification accuracies of classifiers learned using the method of Chapter 6 with our method that uses action context. We present both qualitative and quantitative results demonstrating that object recognition is enhanced in our new framework.

We do not have object labels during training and testing. As a result, to measure the object recognition accuracy quantitatively, we manually annotate the ground-truth object label corresponding to each super-pixel in the foreground region. We perform these annotations for a sub-sample of

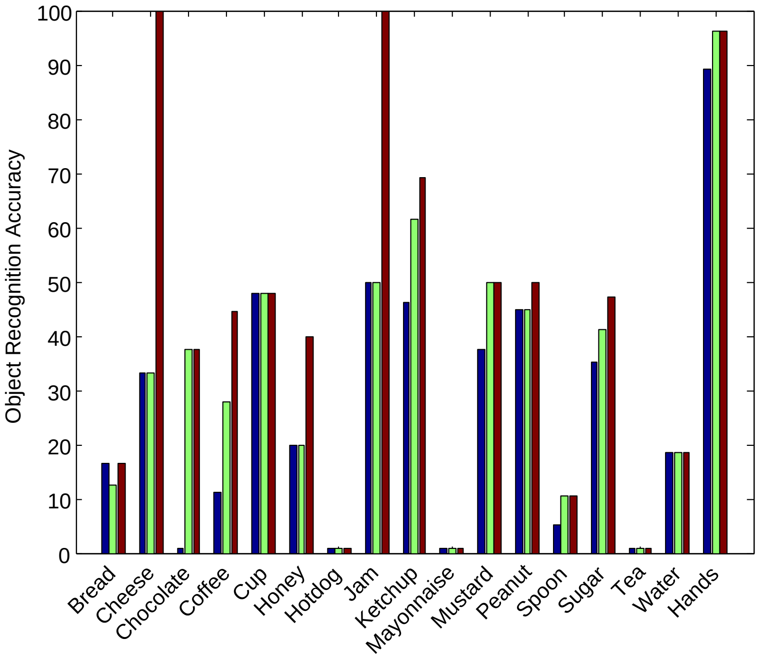


Figure 44: Object and hand recognition results are depicted. We compare the following three methods: object recognition results using the object classifiers only (blue, 27.3% average accuracy), object recognition results using our bottom-up top-down refinement algorithm (green, 33.1% average accuracy) and object recognition accuracy given the ground-truth action label (red, 43.4% average accuracy). As is shown, knowing the action improves object recognition accuracy. Our system is capable of classifying 96.3% of the regions corresponding to hands correctly.

frames (every 50 frames) of the test sequences. For each object class, we measure the the recognition accuracy. Recognition accuracy for an object class is the percentage of the super-pixels that are correctly classified by the algorithm and belong to that class. The results are shown in Fig 44. We demonstrate that object recognition accuracy improves when the ground-truth action labels are available. There is a significant boost in the case of jam, cheese, coffee, honey and chocolate. We show qualitative results of improved object recognition given action labels in Fig 43.

Our algorithm fails on recognizing the following object classes: tea-bag, hotdog and mayonnaise. In case of the tea-bag and the hotdog, the main issue is that the Berkeley segmentation which is used for getting super-pixels does not perform well on these objects. In case of hotdog, the reason that the Berkeley segmentation [8] fails is that it has a similar color and texture to hands, bread and the table and in case of the tea-bag, the reason is that it has a similar color and texture to the wall and the inside of the cup. In case of mayonnaise, Berkeley segmentation performs well, and the MIL framework is able to discover mayonnaise regions correctly. However, the final classifiers confuse mayonnaise with other objects that have similar color and texture such as spoon, napkin and plate.

In addition to the 16 object instances, in Fig 44, we show recognition accuracy for the hand regions. The bottom-up segmentation algorithm returns initial hand segments. Then we learn a classifier that discriminates regions that correspond to hands from background and other objects. This improves the recognition accuracy from 78% to 89% as shown in Fig 44. Furthermore, we use the action context to improve the hand recognition accuracy to 96.3%.

Note that the object detection results depicted in Fig 44 are not directly comparable to the results in Fig 36. The reason is that different criteria are used for measuring the accuracy in these two figures. In Fig 36, the ground-truth labels consist of one active object label per each frame. We count an active object in a frame as detected by our method if the region with highest classification score among all the regions in that frame is assigned an object label that matches the ground-truth label. However, in Fig 44, we assign ground-truth object labels to every region in the foreground. A region is correctly detected if the classification result of our algorithm matches the ground-truth label.

7.4 Conclusion and Future Work

We describe a novel approach to the analysis of activities in egocentric video. Our method constructs a description of an activity in terms of the objects and actions with which it is performed. We leverage the inherent coherence of views and appearance that arises from the egocentric context. We show that object and action models can be learned with very little supervision, by exploiting the joint properties of objects, hands, and actions. We propose a hierarchical inference architecture in which bottom-up propagation of evidence for objects and actions is used to predict the activity category, followed by top-down refinement of object and action descriptions based on the activity model. We demonstrate that our approach can produce superior results in comparison to standard bag-of-words type representations for activity categorization.

A potential issue with our current bottom-up top-down inference in the graphical model is that if a mistake is made at a level of the graph, it will propagate to other levels as well. For example, if an activity is labeled wrong, it will effect the label of the actions, and also consequently it will affect the label of the objects.

7.4.1 Benefits of Egocentric Vision

We believe that the joint inference of activities, actions and objects is essential for describing a scene. This fact is not limited to egocentric videos and indeed applies to any setting. However, there are two stages in our algorithm that we leverage the benefits of the egocentric vision: (1)

for weakly supervised leaning of objects; and (2) for recognizing actions. We listed the benefits of egocentric vision for weakly supervised learning of objects in Section 6.5.1. Here we further argue that egocentric vision helps to recognize daily actions. We showed in this chapter that the location of the hands in images and their poses are extremely useful features for action recognition in egocentric video. These features are not effective in allocentric video. In the view of a static camera, the location of the hand does not lead to any cues for the recognition of actions. In addition, the resolution of the hands in static view is not high enough for capturing the details of the hand pose during object manipulation.

Chapter VIII

LEARNING THE FUNCTION OF OBJECTS

In the last two chapters, we presented methods for recognizing objects and also recognizing activities in egocentric videos. Now, we want to go beyond surface level recognition, and propose a method that results in richer understanding of actions and objects. In this chapter, we will present a weakly supervised method for learning the state of the objects (e.g. open vs close, or empty vs. full) and materials. Then we will show that modeling actions through state changes results in superior performance in comparison to modeling actions based on motion and hand-object relationships. Finally, we will show that it is possible to temporally segment the long activity videos into action intervals by detecting the state of the objects in each frame.

What makes an action (e.g. "open the jar") identifiable? How can we tell if such an action is performed? Over the last two decades various cues have been used to model and understand actions in computer vision: holistic shape and motion description [21, 47], space-time interest points [104], feature tracks [145], object and hand interaction [50, 68, 190] and various other techniques. The common theme among all these works is that they model an action by encoding motion and appearance throughout the interval in which it is performed.

However, in order to fully understand actions we must understand their purpose [176]. Actions with similar motion patterns and hand-object relationships can have a different meaning because they accomplish a different goal. For example, "open coffee jar" and "closed coffee jar" are two different actions, in fact they are inverse. However, they produce similar motion patterns and involve the same object. The key to distinguishing these two actions is to be able to detect the **state** of the "coffee jar" (open vs. closed) and how it changes by these actions. For example, the opening action changes the state of an object from closed to open.

Based on this observation, we introduce a method for recognizing daily actions by recognizing the changes in the state of objects and materials. Most actions can be performed only if certain preconditions are met. Moreover, their execution causes some existing conditions to change. For instance, the action "spread jelly on bread using knife" requires jelly to be on the knife but not on the bread when it is applied. This action changes the state of the jelly from being on the knife to being spread on the bread. Or for example, "take cup" is an action before which the cup is not

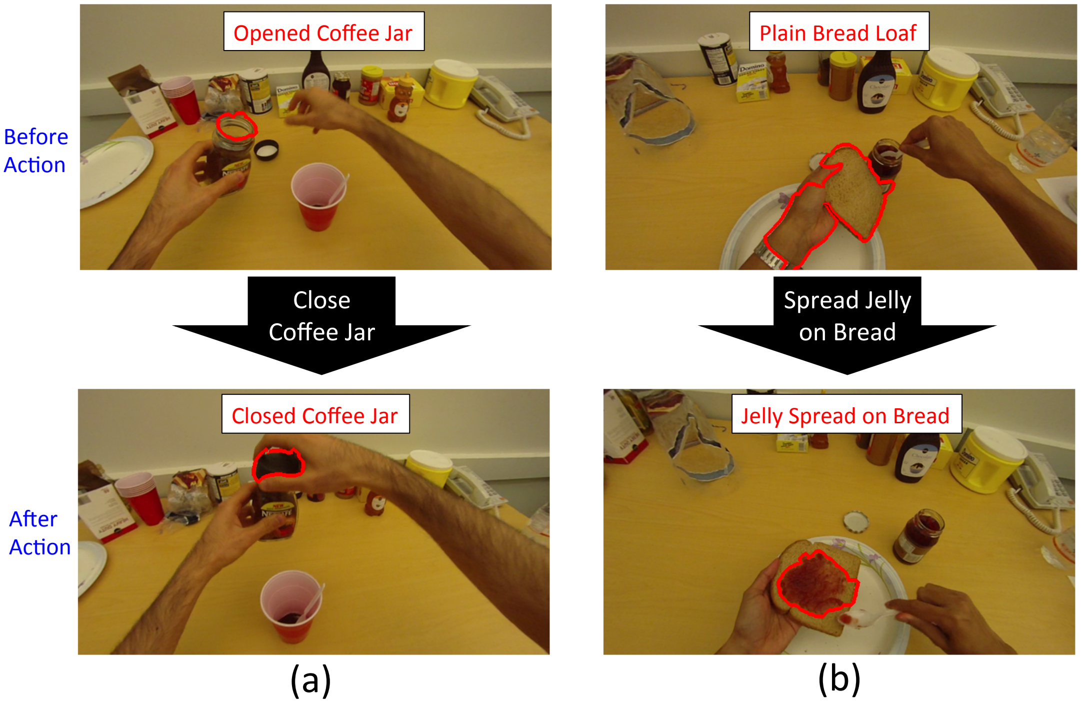


Figure 45: By comparing the initial and final frames of an action, and exploiting the action label, we can learn to detect the meaningful changes in the state of objects and materials produced by the action. In example (a), our method recognizes the action of “close coffee jar”, as a result of detecting regions corresponding to open and closed coffee jar. Similarly in example (b), the action of “spread jelly on bread” is recognized by detecting the regions corresponding to plain bread loaf and jelly spread on bread respectively in the initial and final frames of the action.

being held by the hand, but once it is performed the cup is grasped by the hand[1].

We are interested in two kinds of changes: 1) changes in the state of objects (e.g. coffee-jar becoming open or closed) and the transformation of stuff (e.g. coffee powder mixing with water, jelly getting spread on bread, egg getting scrambled). For example, the following actions take place during the activity of “making coffee”: (open coffee jar), (scoop coffee using spoon), (pour coffee into cup) and (put hot water into cup), (close coffee jar). Throughout these actions, the coffee jar changes states from closed to open and again to closed. Likewise, the coffee powder changes state from being in the coffee jar to being on the spoon, and then being in the cup, and finally dissolving into hot water.

Following our previous works [56, 50, 52] and like many other recent works [138, 165], we adopt egocentric paradigm for recognizing daily activities and actions. Analyzing the details of hand-object interaction is challenging in third-person view videos due to insufficient resolution of hands

[1] Note that this notion of actions as state changing processes holds for most cases, however, there are exceptions such as “dancing” that do not create any describable or observable changes in the environment. In this chapter our focus is on goal-oriented object-manipulation tasks which are often intended to accomplish a particular goal.

and objects. In contrast, the egocentric view puts the environment into the center of the action interpretation problem. In this view, the subject often naturally avoids occlusion which results in high resolution and detailed images of handled objects. We leverage the egocentric paradigm to build fine-grained representations of the object states and materials in order to describe object manipulation tasks.

In this chapter, we propose a weakly supervised method for learning the object and material states that are necessary for recognizing daily actions. Once these state detectors are learned, we run them at each frame of the videos and describe the environment at each moment in time based on the existence or absence of detected object and material states. We introduce methods that leverage the changes in the state of the environment to recognize actions and segment activities. Our results outperform state-of-the-art action recognition and activity segmentation results. Our contributions in this chapter are: 1) We present a model for actions based on the changes in the state of the environment, 2) we introduce a method for weakly supervised discovery of state-specific regions from action videos and 3) we provide an activity segmentation method by verifying the consistency of the environment state with the beginning and ending conditions of the actions. The method and the results of this chapter have appeared in the following CVPR 2013 paper: Fathi and Rehg [55].

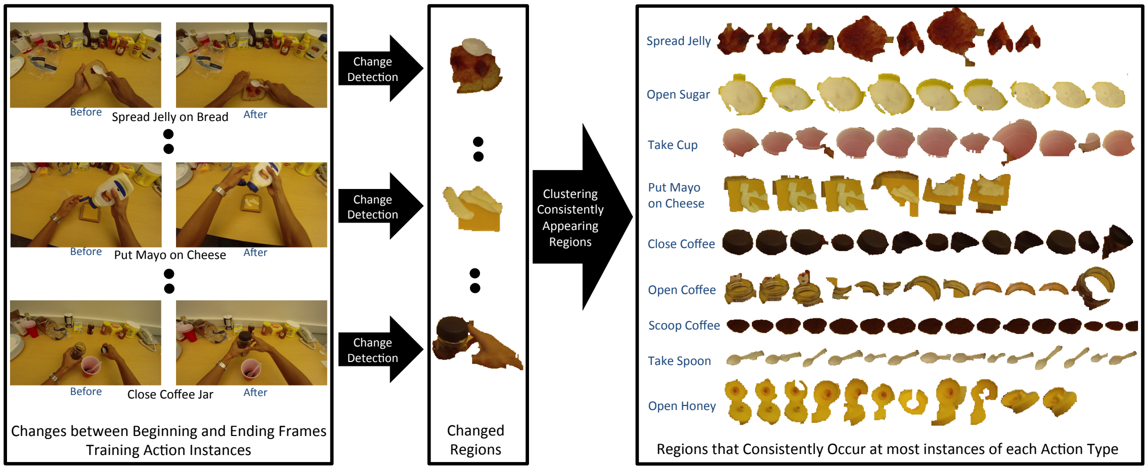


Figure 46: Stages of our state-specific region discovery framework are shown. This procedure only takes place during the training phase. First for each action instance, we compare its initial frames with its final frames to extract the regions that are changed. In the second stage we discard the changes that are not common over the examples of their corresponding action type. In the final stage, we learn a detector for each group of consistent regions. During the testing phase we apply the trained region detectors to describe actions and states.

8.1 Object States and Actions

Our task is to model daily actions via the changes they induce in the state of the environment. We formulate this problem as the discovery of **changed regions** that either correspond to a specific state of an object (e.g. open mouth of the bottle of water) or represent a particular material (stuff, e.g. coffee powder on spoon). We represent actions based on the changes they make in objects and materials. In order to find the regions that are changed as a result of an action, we compare their appearance before the action starts to their appearance after the action ends. The changed regions often correspond to either the state of the objects, or to the materials. Using our method, we show significant gains in action recognition and video segmentation performance.

During the training phase, we are given a set of activity videos. An activity like making peanut-butter and jelly sandwich, consists of a sequence of atomic actions (e.g. take bread, open peanut-butter jar, spread peanut-butter on bread using knife, etc.). For each training activity video, the actions are annotated. The annotation for each action contains its start frame, end frame, a verb (e.g. scoop), and a set of nouns (e.g. coffee, spoon). We emphasize that we are not provided with any object location or mask. In Sec 8.1.1, we introduce a method for discovering regions that correspond to object states and materials from video images. We further learn various state-specific region detectors from the set of discovered regions. In Sec 8.1.2 and 8.1.3, we propose a method for recognizing actions based on the change in the detected object states and materials. Finally, in Sec 8.1.4, we introduce a method for segmenting a new video into a sequence of actions by localizing their initial and final frames.

8.1.1 Discovering State-Specific Regions

The first step in our training phase identifies regions that are representative of the state of an object or existence of a material. In this stage, we make two assumptions: (1) an object state or material does not change unless an action is performed and (2) an object state or material change is associated with an action only if it consistently occurs at all instances of that action. Fig 46 illustrates our three stage approach to discovering the state-specific regions. In the first stage, we identify regions that either appear or disappear as a result of the execution of each action instance. For example, the region corresponding to the lid of the coffee jar will change as a result of performing the action of open coffee. However, there may be other irrelevant changes in addition. For example, a change in the appearance of hand as a result of its movement. In the second stage, we prune changes that are not consistently associated with an action. Finally, in the third stage we learn a detector for each group of discovered state-specific regions. We use these detectors to classify unknown regions during the testing phase.

Change Detection: In this stage, we find the regions that either appear or disappear throughout each action instance in the training set. Each action instance corresponds to a short interval which is a sub-part of a longer activity video. For each action instance, we sample a few frames from its beginning and a few frames from its end. We compare the beginning and ending frames to find their differences. For each pair of beginning and ending images, we match their pixels using large displacement optical flow [25]. Then for each pair of matched pixels, we compute change based on their color difference, similar to the method of Sand and Teller [152]. We calculate the significance of change for each region based on the average amount of change in its pixels. The regions that we use in our algorithm are acquired using the method in Arbelaez et al. [8].

These appearance and disappearance patterns often correspond to changes in object states or the creation of new materials. For example, pouring water into a cup containing coffee powder results in the appearance of a new dark brown liquid region in the cup. Of course there will be many other irrelevant changed regions due to occlusion, lighting effects, and other factors. To overcome such mistakes, we compare each beginning (ending) image to multiple ending (beginning) images. We set the amount of change to the minimum amount computed among all the comparisons. A few examples of the results of this stage are shown in the second column of Fig 46. After this pruning procedure, still there are often regions left that do not correspond to state changes and materials. The next step is to remove them.

Consistent Regions: In the previous stage, we extract regions that have changed between the initial and final frames of each action instance. Now in this stage, we only keep the subset of those regions that consistently occur across the instances of an action type. For example, a region that corresponds to coffee jar's lid consistently appears at the beginning of the "open coffee", but a spurious region would not. A region r consistently occurs at an action class $\mathcal{A}$, if there is a region $\hat{r}$ similar to r at each instance a of that action class ($a \in \mathcal{A}$). Here we suggest an algorithm that extracts the consistent regions, and further groups them based on their similarity. Inspired by the source constrained clustering method of Taralova et al. [171], we cluster the N extracted regions from instances of action class $\mathcal{A}$ into k sets $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}$ by enforcing the regions in each cluster to be drawn from the majority of action instances. We further add an additional constraint that each action instance can at most contribute one example to each cluster. This constraint prevents us from adding regions that correspond to non-relevant object parts but have similar appearance to the cluster. We do this by optimizing the following objective function:

$$\begin{aligned} \arg\min_{\mathcal{S}} \quad & \sum_{i=1}^{k} \sum_{x_j \in \mathcal{S}_i} \|x_j - \mu_i\|^2 \\ \mathit{subject\,to}: \quad & \sum_{a \in \mathcal{A}} \delta(S_i, a) \geq h \\ & \delta(S_i, a) \leq 1 \end{aligned} \tag{3}$$

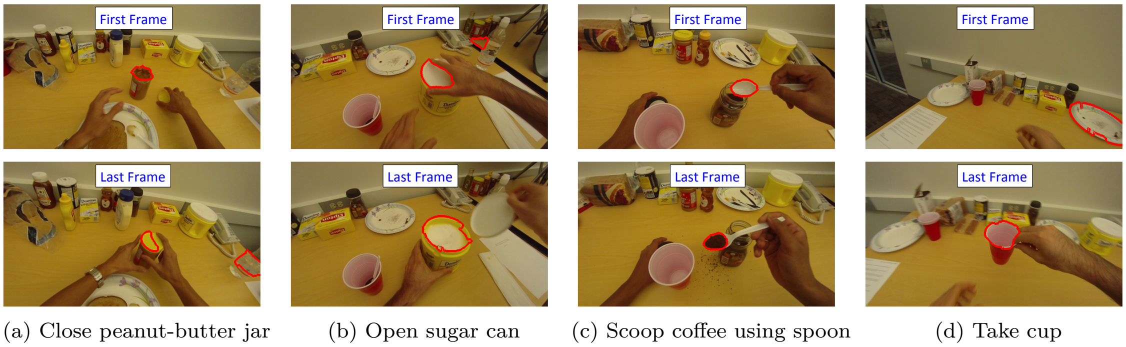
where x_j is a feature vector representing region j, μ_i is the mean of points in $\mathcal{S}_i$, a is an instance from the set of all instances of the action class $\mathcal{A}$, and $\delta(\mathcal{S}_i, a)$ is a function that returns the number of regions from action instance a in cluster S_i, and h is a scalar. This objective function is similar to the objective function of k-means with two additional constraints that enforce a cluster $\mathcal{S}_i$ to have samples from at least h action instances.

We approximately minimize the objective function in Eq 3 through a simple iterative approach. In each iteration, we pick the best set of h regions with minimum distance from each other and return them. We make sure each of these regions is picked from a different action instance. In the next iteration, we remove the previously returned regions from the set of remaining regions and repeat the procedure. We continue this until either k clusters are returned or there are less than h action instances with regions left in them. See Algorithm 8.1 for the details. In our experiments, we only use the first few clusters which have the highest self-similarity. We have shown examples of such clusters in the right-most column of Fig 46.

Algorithm 8.1 One iteration of selecting consistent regions

```
set of best $h$ regions $R = \{\}$
total intra-region distance $b = \infty$
a temporary set for keeping regions $\bar{R} = \{\}$
for (every region $r$ in every action instance $a \in \mathcal{A}$)
  for (each $\hat{a} \neq a, \hat{a} \in \mathcal{A}$)
    pick the closest region in $\hat{a}$ to $r$ and add it to $\bar{R}$
  end
  select a subset of $h$ regions in $\bar{R}$ with min total distance $d$
  if ($d < b$)
    set $R$ to the subset of $\bar{R}$
    set $b$ to $d$
  end
end
return $R$ as a cluster
remove $\bar{R}$ from the set of extracted regions
```

State-Specific Region Detectors: In this stage, we learn a detector for each of the region clusters. We train a linear SVM by using the regions belonging to the cluster as the positive set and all the regions in activities that do not contain the action as the negative set. We describe



(a) Close peanut-butter jar (b) Open sugar can (c) Scoop coffee using spoon (d) Take cup

Figure 47: In contrast to the features of conventional action recognition methods, our features are meaningful to humans. Regions that correspond to state-specific detectors with a classifier weight higher than a threshold are shown with a red boundary. For example in (a), in the first frame SVM puts a high weight on the open mouth of peanut-butter jar and in the last frame puts a high weight on peanut-butter jar's lid.

each region with color, texture and shape features. For each region, we build a 128 dimensional color histogram by quantizing the color values of pixels into clusters. In addition, we build a texture histogram by computing texture descriptors [177] for each pixel and quantizing them to 256 centers. We further compute a 16 dimensional shape feature vector for each region. Our shape features are similar to HOG [37] features, but instead of computing them on patches, we compute them on the whole region. We compute the gradient at all pixels inside the region and quantize them into 16 orientations. We count the occurrence of gradients in each orientation. We concatenate these three features together into a 400 dimensional feature vector that we use to represent regions.

8.1.2 States as Action Requirements

An action can be performed only if certain conditions are satisfied in the environment. For example, "clean the table" is an action that requires the table to be dirty before the action is performed, and clean afterwards. Thus, the key to recognizing a goal-oriented action is to be able to recognize the state of the environment both before and after that action.

We represent the environment state based on two criteria: (1) existence or absence of state-specific regions and (2) whether or not an object (region) is grasped and is being manipulated by the hands. To model the first criteria, we represent each frame of the test video by the response vector of the trained state-specific region detectors (Sec 8.1.1). For each detector, we run it on all the regions of the test frame and pick the highest classification score as its response. We set the responses that are higher than a threshold to 1, and the ones that are lower than a threshold to -1. We set the rest of the responses to 0. This quantization helps us to avoid overfitting. In order to model if the regions are being grasped by the hands or not, we use the foreground segmentation method in Chapter 6 which identifies if a region is being moved by the hands or not. We build a similar vector based on the responses of the detectors on the foreground regions, instead of applying them to all regions. We represent each frame by the concatenation of its response vectors.

8.1.3 Modeling Actions through State Changes

The majority of common action recognition approaches rely on analyzing the motion and appearance content of the action intervals. Movement patterns are crucial for recognition of many actions, in particular body movements such as running, walking, dancing, etc. However, most daily object-manipulation tasks are goal-oriented actions that are defined by the changes they cause to the state of the environment.

Given a test action interval, we build two response vectors. One is based on the response of the detectors on its beginning frames, and the other is based on the responses on its ending frames. We represent the interval by concatenation of these two vectors. We use linear SVM to train a classifier for each action type. Since we have concatenated the vectors of beginning and ending frames, linear SVM can model the change of an object state or material by putting weights on its corresponding responses. Linear SVM will put higher weights on state-specific regions that are consistently created either at the beginning or at the end of the action, and lower weights on the ones that do not relate to the action. We show visualizations of the classifier weight vectors for few action instances in Fig 47.

8.1.4 Activity Segmentation

Activity segmentation of a test video is the task of breaking a long activity video into a sequence of short actions. In order to do so, often one takes all the action detection scores as input and infers the frames that are assigned to each action in that video. In order to handle detection errors, a common strategy is to apply the action classifiers to every possible interval, and then use non-maximum suppression or dynamic programming [128, 50].

Here instead we leverage the capability of our framework for detecting environment states to segment activity videos. In state detection, different than action recognition, the problem is to assign a state label to each frame of the video. The possible set of states are: 1) before a particular action starts, during that action, after that action ends. Our method is as follows. For each action class (e.g. open coffee), we train two state detectors, one using its beginning frames and one using its ending frames. The state detectors are learned on top of the frame's responses to pre-trained state-specific region detectors (Sec 8.1.3).

The state detectors are trained using linear SVM by taking the action's beginning or ending frames as positive set and all the other training frames as negative set. Given a test activity video, we apply all the trained beginning and ending state detectors on its frames. This results in two $|\mathcal{A}| \times T$ matrices S_B and S_E respectively, where $|\mathcal{A}|$ is the number of action types, T is the number of frames in the test activity video, and $S_B[a,t]$ and $S_E[a,t]$ respectively contain the classification scores of detecting the initial and final frames of action a at frame t.

We segment an activity video into a sequence of intervals $\mathcal{I} = \{I_1, ..., I_{|\mathcal{I}|}\}$. An interval I_i has a few properties: I_i^a identifies the its action label, I_i^{st} identifies its initial frame number, and I_i^{en} identifies its final frame number. We segment the activity video by optimizing the following objective function:

$$\begin{aligned} \arg\max_{\mathcal{I}} \quad & \sum_{I_i \in \mathcal{I}} S_B[I_i^a, I_i^{st}] + S_E[I_i^a, I_i^{en}] \\ \mathit{subject\ to:} \quad & I_i^{st} < I_i^{en} \\ & (I_i^{st} - I_j^{st}).(I_i^{en} - I_j^{en}) > 0 \\ & M(I_i^a, I_{i+1}^a) > 0 \end{aligned} \tag{4}$$

where M is a binary transition matrix. The objective function aims at finding the best set of intervals where the total sum of scores is maximized. The score of interval $I_i : \{I_i^a, I_i^{st}, I_i^{en}\}$ is computed by adding the response of the detector corresponding to the initial frame of action I_i^a on frame I_i^{st} with the response of the detector corresponding to the final frame of action I_i^a on frame I_i^{en}. There are three constraints involved in the optimization. The first two constraints prevent action intervals from overlapping with each other. The third constraint limits the possible transitions between actions. For example, it is not possible to pour milk after close milk is performed. We train the matrix M based on observed action transitions in training activities.

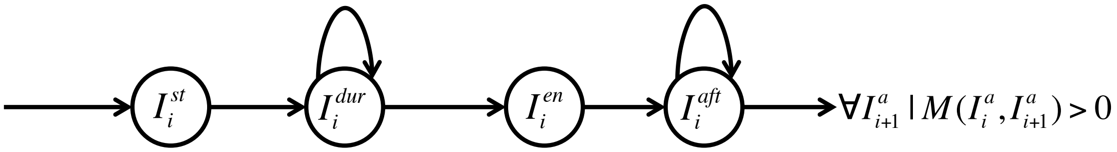


Figure 48: Possible transitions are shown for states of an interval I_i.

We can model this problem as a finite state sequential process and optimize it using dynamic programming. For this purpose, we have to assign a state to each frame of the video. In order to do this, in addition to the first frame of the interval I_i^{st} and its last frame I_i^{en}, we add two auxiliary states for it: during I_i^{dur} and after I_i^{aft}. The score of entering these states is zero, and they are only used to enforce the constraints of the Eq 4. For example, it is only possible to transition from the first frame of an interval to its during state, and then, either stay in its during state or move to its ending frame. The set of possible state transitions for an action are shown in Fig 48.

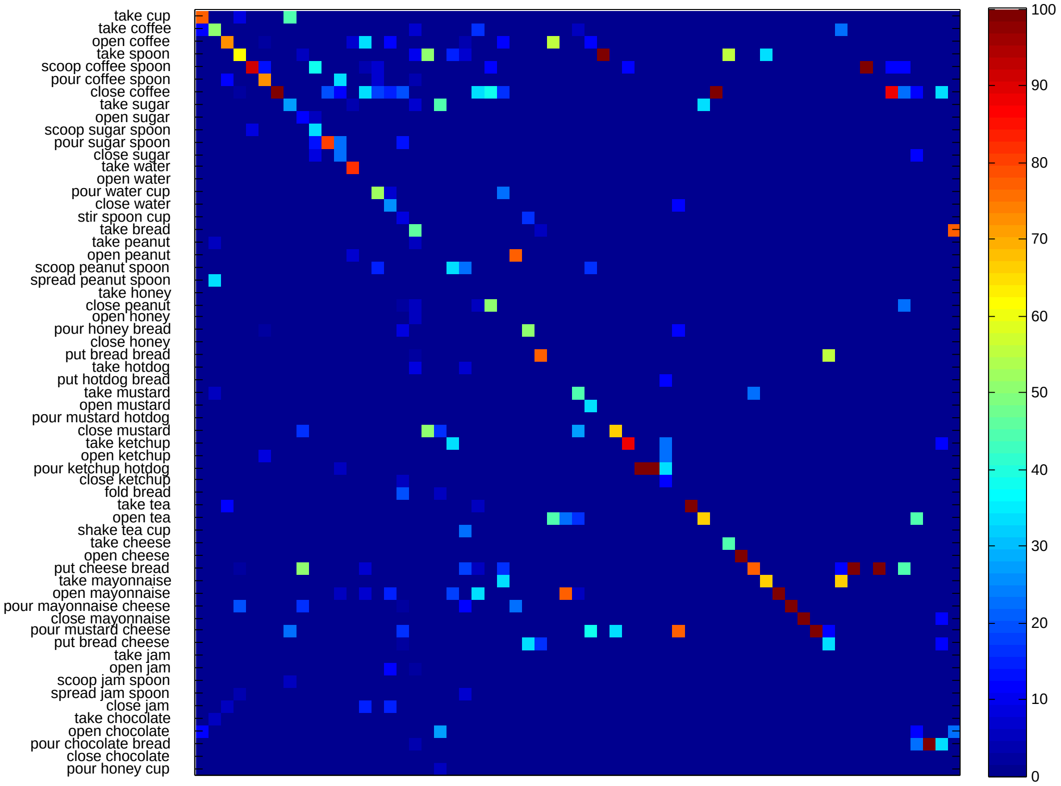


Figure 49: Confusion matrix for recognizing actions using our method is shown. The average accuracy is 39.7% on 61 classes of action which is significantly higher than the baseline (23%). Random classification chance is 1.6%.

8.2 Experiment 3: Object States and Action

To validate our model of actions based on state changes, we show extensive qualitative and quantitative results on two tasks: (a) action recognition: assigning an action to a given interval and (b) activity segmentation: decomposing an activity into a sequence of actions by detection and decoding. We compare our results to state-of-the-art performance and different baselines.

8.2.1 Action Recognition

We evaluate our method on GeorgiaTech Egocentric Activity (GTEA) dataset that was described in Section 6.3. This dataset consists of 7 types of activities, where each activity is performed by 4 subjects. There are 61 actions in this dataset, after omitting the background action classes and

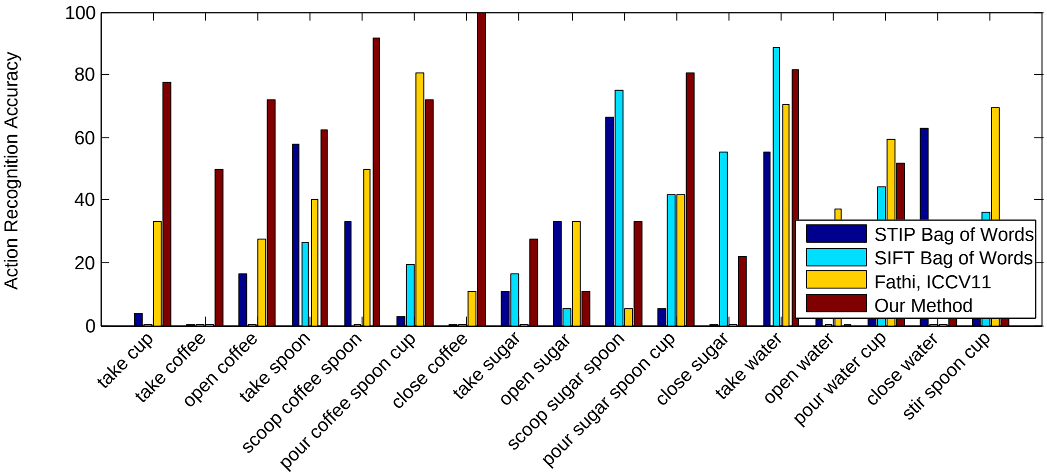


Figure 50: We compare the performance of our method with various baselines. STIP bag of words results in 11.6% accuracy, SIFT bag of words results in 19% accuracy, and our method in Chapter 7 results in 23% accuracy (shown in the figure as Fathi, ICCV11). Our state-based method (referred to as our method in the figure) significantly outperforms these baselines by achieving 39.7% accuracy on 61 classes, where the random chance is 1.6%. We show the comparison on the actions of the activity of making coffee.

fixing some of the mistakes in the original annotation. Training and testing sets are chosen as is done in Section 7.3.

Baselines: we compare our method to three baselines. The first baseline trains a linear SVM on concatenation of STIP [104] bag of words built from the first and second half of each interval. This STIP baseline performs poorly on this dataset, resulting in 11.6% accuracy in recognizing 61 classes. We believe the reason is that in an egocentric setting the camera is continuously moving, which makes space-time interest points fire at areas that do not relate to the action. The second baseline trains a linear SVM on two SIFT [114] bag of words built for each interval. SIFT features perform better than STIP, however, they still perform poorly, resulting in 19% accuracy. This is because many of the daily objects used in these activities are textureless and they produce no corners. Finally we compare our method to our method in Chapter 7 which uses hand motion, hand location, objects in foreground and hand pose to recognize actions. This method achieves 23% accuracy on this dataset for 61 classes[2].

Our State-Based Method: Given an action interval, we build two response vectors: one using its beginning frames and one using its ending frames, as described in Sec 8.1.3. We have 610 state-specific region detectors. We train 10 detectors from the consistent changes of each action

[2] These numbers are different than the ones reported in Chapter 7. The action recognition results of Chapter 7 are based on recognizing action verbs such as pouring, opening and closing. However, in this chapter, the action labels contain both verbs and object names. For example, pouring mayonnaise on the cheese, opening coffee jar and opening honey. There are 11 action verbs in the GTEA dataset, while the number of action labels is 61.

type. We describe a frame by applying each of these detectors on its regions and taking its highest response. We set the responses greater than 0.5 to 1, responses smaller than -0.5 to -1, and the responses between -0.5 and 0.5 to 0. We further build similar response vectors from foreground regions. Finally we represent each interval with a $610 \times 2 \times 2 = 2440$ dimensional feature vector, and train a linear SVM for each action type. Our method achieves 39.7% accuracy on 61 classes, where chance is 1.6%. This represents a significant improvement over previous baselines. We have shown the confusion matrix for our method in Fig 49. We have compared the accuracy of these methods in Fig 50. The state-based method results in low accuracy in recognizing the actions performed on some of the objects, for example sugar. The reason is that the algorithm has a hard time telling the difference between the different states of those objects. For example, an open sugar can and a closed sugar can look very similar. This is because the sugar can lid is white, and the sugar powder is also white, which makes it hard for the algorithm to tell if the can is open or closed.

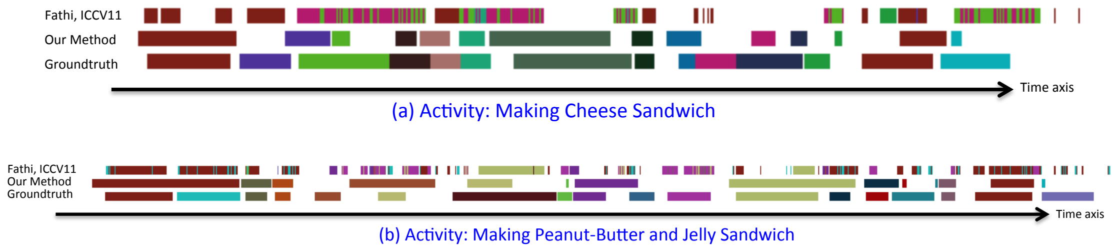
Classifier Visualization: Many conventional action recognition methods rely on features such as corners, point tracks, etc. that are not meaningful to humans. In contrast, our features are state-specific regions that correspond to object parts or materials, and they can be easily interpreted. The weights of the linear SVM classifier trained on examples of an action determines the state-specific parts and materials that should exist in its initial or final frames. We have shown regions that correspond to state-specific detectors with high classifier weights in Fig 47.

8.2.2 Activity Segmentation

Given a video, we use our activity segmentation method described in Sec 8.1.4 to segment it into different actions. We compare our method to the detection results from the method of Chapter 7. In the method of Chapter 7, we train a CRF for each activity and apply that on action scores to force transition constraints. That method enforces much harder constraints in comparison to our new approach. Using our new method we achieve 42% accuracy and outperform our previous results of 33% accuracy. The results are computed by counting the percentage of the frames that are correctly labeled in the test activities. We show segmentation results in Fig 51 on two test activity videos. The results are in general smoother in comparison to the results of Chapter 7.

8.3 Conclusion and Future Work

In this chapter, we present a model for actions based on the changes in the state of objects and materials. We show significant gains in both action recognition and detection results. We further introduce a simple temporal and logical encoding method for activity segmentation that outperforms



(a) Activity: Making Cheese Sandwich

(b) Activity: Making Peanut-Butter and Jelly Sandwich

Figure 51: Activity segmentation results are shown for two test activity videos. The horizontal axis represents time (frames). Different colors represent different action labels assigned to the frames. In each example, the bottom row shows the ground-truth results, the top row shows the results of our method in Chapter 7, and the middle row shows our current results. We correctly assign true labels to 42% of frames and outperform our previous results with 33% accuracy. In addition, our labels are smoother in comparison to our previous method.

the results of previous state-of-the-art methods. In addition, we introduce a method for discovering state-specific regions from the training action examples.

Limitations: our current approach for discovering states of the objects only works for object instances, and would fail if different instances of the same object are used. The other limitation of the current approach is that it cannot handle the scenario that the person performs the action while moving and the background changes significantly. Despite these limitations, we believe the high-level idea is promising and worth pursuing. We will work on these limitations in the future work.

Meal-Preparation Domain: We believe meal-preparation is a great domain for studying object and material states. There are tens of objects and materials used in making a simple meal like pizza. Objects transfer from one state to another and change. For example, a lettuce is initially a set of leaves stuck together, then the leaves become separate, then they get cut into pieces and finally they get merged with other vegetables or sauce. It is very hard to find any other domain that can compete with meal-preparation domain in the variety of objects and actions. We have annotated tens of actions in each of the activities of GTEA dataset.

Benefits of Egocentric Vision: We believe the reasons that our algorithm can discover the state of the objects with a very limited supervision are as follows: (1) consistent view point of the objects during activities in egocentric setting (e.g. the location of the "coffee jar" with respect to our eyes is often consistent during the action of "opening coffee jar"); (2) high resolution of the objects given the close distance of the hands to the eyes during daily activities; and (3) the usual absence of occlusion since an individual naturally tries to avoid occlusion during performing actions.

Future Work: We believe an interesting future direction for research involves building a taxonomy of possible states of objects and materials. Modeling these states would require richer features that capture shape, physical properties and affordances of things and stuff. A potential challenge would be modeling the changes that do not correspond to observable visual patterns. The benefit of our current weakly supervised approach is that it is sensitive to distinguishable visual changes in objects and materials, and implicitly ignores the ones which can't be observed in videos. Combining this benefit with stronger domain models could be valuable.

Chapter IX

LEVERAGING GAZE TO PREDICT ACTION AND INTENTION

In the previous chapters, we presented methods for recognizing daily actions in egocentric video. In Chapter 7, we introduced a method that models actions based on the interaction between hands and active objects, and in Chapter 8, we described an approach which models actions through object state changes. In this chapter, we study the value of first-person's gaze for recognizing daily actions. We will show that gaze measurements can improve the accuracy of action recognition results in egocentric video. However, at the same time we will demonstrate that we do not need to measure the gaze, but instead, we can predict the gazed points from contextual cues that are available in egocentric video.

Ever since the pioneering experiments of Yarbus [191], it is well known that human attention and gaze are directed in a top-down task-dependent and goal-oriented manner. This is summarized in the following quote from [191]: "Eye movement reflects the human thought processes; so the observer's thought may be followed to some extent from records of eye movement." Hayhoe and Ballard [73] note that the point of fixation in the scene may not be the location which is the most visually salient, but rather will correspond to the best location given the spatio-temporal demands of the task. However, in computer vision, research on visual attention has been primarily based on bottom-up approaches [82]. Research on attention based on top-down components such as scene content, actions and objects has been very limited [193, 59, 22].

A basic challenge in the top-down study of gaze is that there is not always a direct relationship between actions and fixations. For example, a person can easily carry an object in her hand and put it on the table without looking at it. To address this issue, in this chapter, we focus on object-manipulation tasks that require hand-eye coordination. These are actions that are hard to accomplish without using both hands and eyes in coordination. For example, when pouring a liquid into a bottle, subjects initially fixate on the mouth of the bottle, and then switch to monitoring the level of liquid in the container once they are past the half-way mark. In their classic study [100], Land and Hayhoe demonstrated that during object manipulation tasks a substantial percentage of fixations (around 80%) fall upon the task-relevant objects.

As an illustration of the close association between gaze and activities of daily living, Fig 52

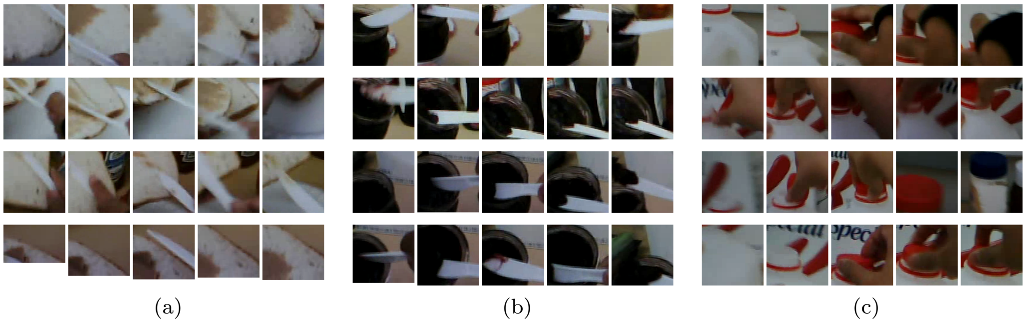


Figure 52: Humans often attend to the location that contains the spatio-temporal information of the task. While this might not be true in some cases such as covert gaze, but in general the region around the gaze location provides significant information about the action. In the figures above, in each row we show a sequence of bounding boxes extracted around the gaze point from a particular instance of the action. For each of the action types, we show four rows of boxes, each selected from one instance of the action. The actions are (a) spread peanut-butter on bread using knife, (b) scoop jam using knife and (c) close milk.

contains small windows of pixels which have been extracted from around the gaze location. Columns correspond to frames, sampled at every two seconds. Rows correspond to different instances of a particular action. We observe that the appearance of these small windows is very consistent among instances of the same action performed by different individuals. Moreover, window contents vary significantly between actions. This observation illustrates the close relationship between eye movement, action and objects in such tasks.

Previous investigations of eye movement have largely been based on studies of static scene viewing, using gaze tracking technology affixed to a monitor screen. However, in order to study gaze in the context of object manipulation tasks, a mobile system that captures human gaze in real-life setting is required. Recently, wearable gaze tracking systems, such as [41], Tobii[1] and SMI[2], have become available. These systems combine an outward-facing camera, which captures an ego-centric or first-person view of the scene, with inward-facing gaze sensing cameras that estimate the line of sight into the scene. Calibration of the multi-camera system makes it possible to continuously measure the point of gaze within the scene in front of the user. These systems create new opportunities to exploit gaze measurements in the context of real-world tasks and naturalistic settings. In this chapter, we address the question of how such gaze measurements could be useful for activity recognition in egocentric video.

This chapter addresses the following questions:

[1] http://www.tobii.com/

[2] http://www.eyetracking-glasses.com/

- How consistent are the fixation patterns of different individuals performing the same action?
- Does knowing the fixation location in images of a sequence help to better recognize actions?
- Can we leverage implicit cues that are provided by the body of the first-person to predict gaze?
- Can we develop a method that can learn where to look and how to recognize actions given egocentric video with gaze measurements?

We show that action and gaze behavior are highly coordinated in daily object manipulation tasks. We show that knowing gaze location significantly improves action recognition results, and knowing the action enables more accurate prediction of gaze location. We use these observations and findings in order to learn from humans where to look for and how to recognize the daily actions in egocentric videos. Portions of the method and results in this chapter have appeared in the following ECCV 2012 paper: Fathi et al. [52].

9.1 Dataset 2: GTEA Gaze

In this section and the next one, we present two new datasets which we believe are the first of their kind. These datasets contain gaze location information associated with egocentric videos of daily activities. Our datasets are recorded from the first-person point of view and contain the subjects' gaze location in each frame of the video and are publicly available[3].

We use the Tobii eye-tracking glasses to record the GTEA Gaze dataset. The Tobii system has an outward-facing camera that records at 30 fps rate and 480 × 640 pixel resolution. The glasses use an infrared inward-facing gaze sensing camera to output the 2D location of the eye gaze in each frame of the video. We setup a kitchen table with more than 30 different kinds of food and objects on it. Once each subject wore the eye-tracking glasses and the system was calibrated, we took the subject to the table, and asked them to make a meal for themselves that they can take and have if they like. We didn't put any constraints on their options. Based on the time of the day at which the subject was performing the meal preparation task and their personal preferences, they made different kinds of meal. The two most common meals made by the subjects were turkey sandwich and peanut-butter and jelly sandwich. A few snapshots from this dataset are shown in Figure 53.

We collected 17 sequences of meal preparation activities performed by 14 different subjects. Each sequence took about 4 minutes on average. In our experiments, we use 13 sequences for training and 4 sequences for testing. We make sure that none of the sequences in the test are performed by

[3] `http://cpl.cc.gatech.edu/projects/GTEA_Gaze/`

(a) (b)

Figure 53: Few images from the GTEA Gaze dataset are shown.

a subject from training sequences. We annotated all the actions existing in each sequence. Each sequence contains about 30 actions on average. Each action contains a verb (for example “pour”), a set of nouns (like “milk, cup”) and a starting and an ending frame number. There exists 94 unique actions (unique combination of verbs and nouns) in our dataset. However, many of these actions only take place one or two times through out all sequences. In our experiments we prune the rare actions and only focus on the 25 remaining ones that at least take place two times in training sequences and once in testing sequences. Our set of actions contain the following verbs: take, open, close, pour, sandwich, scoop, spread.

9.2 Dataset 3: GTEA Gaze+

We collected this dataset based on our experience in collecting the first one, in order to overcome some of its short comings. The video quality in this dataset is HD (1280 × 960), tasks are more organized, activities are performed in a natural setting, and the number of tasks and the number of objects used in each task are significantly bigger. The dataset is collected in Georgia Tech’s AwareHome, which is an instrumented house with a kitchen that contains all of the standard appliances and furnishings. A few images depicting our setup are shown in Figure 54. We used SMI eye-tracking glasses to record this dataset.

We have collected data from 6 subjects, each performing a set of 7 meal preparation activities. Activities are performed based on the following food recipes: American Breakfast, Turkey Sandwich, Cheese Burger, Greek Salad, Pizza, Pasta Salad, and Afternoon Snack. Each activity (sequence) takes around 10-15 minutes on average, resulting in more than one hour of data per subject. Gaze location at each frame is recorded. We have annotated the beginning and end of different actions

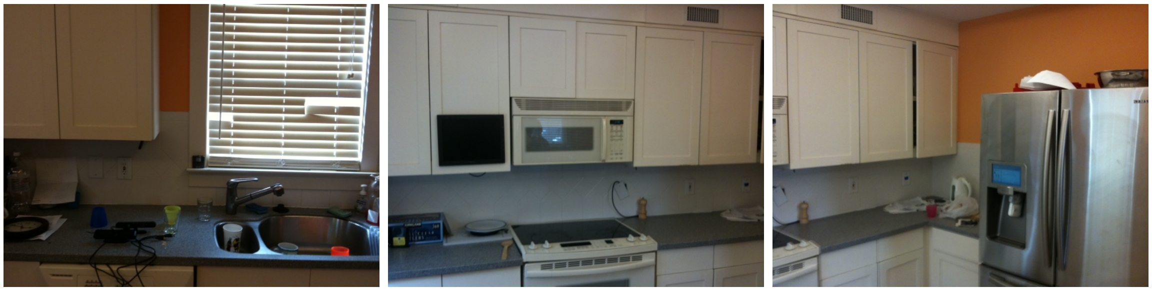


Figure 54: Aware Home at Georgia Tech.

in each activity. Each sequence contains around 100 different actions. Actions in this dataset are associated with the following verbs: taking, putting, pouring, cutting, opening, closing, mixing, transferring, turning on/off, washing, drying, flipping, dividing, spreading, compressing, cracking, peeling, squeezing, filling, reading, moving around, distributing, draining and reading.

Here are the recipes of the meals that are used in this dataset:

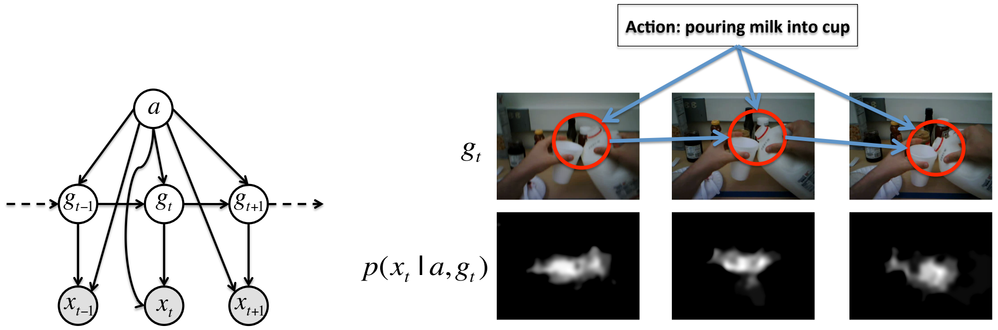
North American Breakfast: Heat a large non-stick frying pan to a setting just above medium. In large metal or glass mixing bowl, whisk the eggs with the milk and salt. Beat vigorously for a minute. Pour some oil in the frying pan and let it heat. Add the egg mixture. Start stirring the mixture after it starts getting hard. Take the orange juice and pour some into a cup. Heat oil in a pan and fry some bacon. Take the cream cheese and bagels, and put some cream cheese on the bagel.

Afternoon Snack: Combine peanut butter and honey in a small microwave-safe bowl. Microwave at high for 20 seconds. Mix the peanut-butter with strawberry jam in the bowl. Spread about 1/4 cup peanut butter mixture on each of 4 bread slices. Top with remaining bread slices. Put some water on the oven to boil in order to make tea. In the mean time, pour some cereal into a cup. Add milk and some honey or chocolate syrup. Pour the boiling water into a cup and put a lipton tea or an instant coffee in it. Add some sugar to the tea or coffee if you like.

Turkey Sandwich: Take a bread. Put some slices of turkey on the bread. Dice lettuce and tomato. Put cheese, lettuce and tomato on the bread. Put your choice of sauce including mustard, ketchup, mayonnaise. Top with another slice of bread.

Cheese Burger: Turn on the oven. Put a pan on the oven. Put a beef pad in the pan. Cook it until it is well done. Turn the beef every couple of minutes while cooking. Put cheese on the beef pad, and let it melt. While the meat is cooking, slice tomatoes and lettuce. Sandwich the cooked meat and cheese in bread, put lettuce, tomato and your choice of sauce.

Greek Salad: Slice the lettuce into a bowl. Peel the cucumber and slice it. Slice tomatoes.



(a)

(b)

Figure 55: In (a), we show the model for predicting the gaze location in images and action. We have visualized our model in context of a few frames in (b). The likelihood map of $p(x_t|g_t, a)$ is shown for action a set to "pour milk into cup". The brighter the pixels in images shown for $p(x_t|g_t, a)$ are, the higher the likelihood.

Slice onion and make rings. Put feta cheese. Add vinegar, lemon juice and olive oil.

Pasta Salad: Put water into a pan and get it to boil on the oven. Pour some macaroni into water until they become soft. Rinse the mac and wash it under cold water. Put the mac into a bowl. Chop tomatoes and green bell peppers into the bowl. Further add chopped cucumbers and onion rings and black olives. Slice some carrots into the salad. Add your choice of dressing.

Pizza: Take a pizza bread. Cut some sausage and put it on the bread. Put some green bell peppers. Fry some mushrooms and put it on the bread. Add mozzarella cheese and add ketchup. Set the oven to $400'$ and put the bread into the oven for 20 minutes.

9.3 Method I: Top-Down Gaze

Our algorithm estimates the action and the most likely sequence of gaze locations in an image sequence by leveraging the fact that human gaze is often focused at locations where the task is being performed. Usually the immediate surroundings of the gaze point contain most of the informative features, and other parts of image contain less relevant information.

9.3.1 Model

We use a generative model to describe the relationship between the egocentric action and the gaze location in each frame of an image sequence, as depicted in Fig 55(a). In this model, an action a can be inferred from the local image features x_t that are observed in the vicinity of the sequence of fixation points g_t. We have visually illustrated the concept of our model in Fig 55(b).

In our model, we have two conditional probabilities: likelihoods $p(x_t|a, g_t)$ and transitions $p(g_t|g_{t-1}, a)$. We model the probability of transition from a gaze location g_{t-1} in frame $t-1$

to gaze location g_t in frame t of an action a with a Gaussian on the distance of the two points in image coordinates. We learn a separate Gaussian model for each action.

$$p(g_t|g_{t-1}, a) \quad = \quad \frac{1}{\sigma_a\sqrt{2\pi}} exp(\frac{-(\parallel g_t - g_{t-1} \parallel - \mu_a)^2}{2\sigma_a^2})$$

The mean μ_a and the variance σ_a^2 of the Gaussian models are learned separately for each action a from training data. In the following we describe our features x_t, and in Sec 9.3.2 we describe the procedure for computing $p(x_t|g_t, a)$. Our method uses the image content in the neighborhood of the gaze location to infer the action.

Based on our observations and experiments, we use three sets of features for each pixel location in an image: (1) features representing the set of objects around that point, (2) appearance features, and (3) features capturing if the image location belongs to an object that will be manipulated by the hands in the near future.

Object-based Features: Objects play an important role in discriminating daily actions. In an action such as “spreading peanut-butter on the bread using knife”, usually it is possible to see parts of peanut, knife and bread in a local neighborhood of the gaze point. It is very uncommon to find the same pattern in an area of an image from another action. To build our object-based features, for each pixel in the image, we concatenate the maximum scores of different object classifiers in its local neighborhood to build a feature vector. We describe the details of our object detectors in Sec 9.3.3.

Appearance Features: Captures the appearance of the gaze location. This feature is used to determine the fixated part of the object. For example the appearance of a milk jar at its handle is different from its appearance at its mouth. In different actions, different parts of an object will be fixated. We compute the histogram of color and texture in a circular area around each pixel and use that as appearance feature.

Future Manipulation Features: This feature is based on the well known fact in the psychology literature that the gaze is usually ahead of the hands in the hand-eye coordinate system [100, 134]. Eyes usually lead to another task before the hands, in order to provide additional input for planning further movements. Land and Hayhoe [100] observed that the average lead time for the tea-making task was 0.56 s and for sandwich-making was 0.9 s. As a result, hand activity in a few frames ahead provides a strong cue for predicting the gaze location in the current frame. In order to build a feature that captures whether an object is manipulated by hands in the future, we first use the method in [149] to segment each frame of the video into foreground and background regions. The foreground regions contain the hands and the manipulated objects. To verify if a pixel in frame f belongs to foreground in t frames later in video, we transfer the computed foreground mask of frame $f+t$ to frame f using the chain of optical flow vectors between adjacent frames. An example is shown in Fig 56. We do this for multiple values of t, and build a $0-1$ feature vector for each pixel location that describes if it is part of the foreground in t frames later or not.

9.3.2 Inference

For each action we learn a SVM classifier that fires on the pixels that are more likely to correspond to the gaze location for that particular action, given the described set of features. To train the classifier, we select the positive features from the pixels surrounding the gaze locations in training sequences corresponding to a. We select the negative features from pixels far from the gaze point in training sequences corresponding to a and all the pixels in training sequences of other actions. A few representative results are shown in Fig 61. We learn the posterior for $p(a, g_t|x_t)$ by fitting a sigmoid function to the output of the SVM classifier learned for action a [139], similar to Lester et al. [107]. We can estimate the $p(x_t|a, g_t) \propto \frac{p(a,g_t|x_t)}{p(a,g_t)}$ from the output of SVM classifiers by assuming a uniform probability for $p(a, g_t)$.

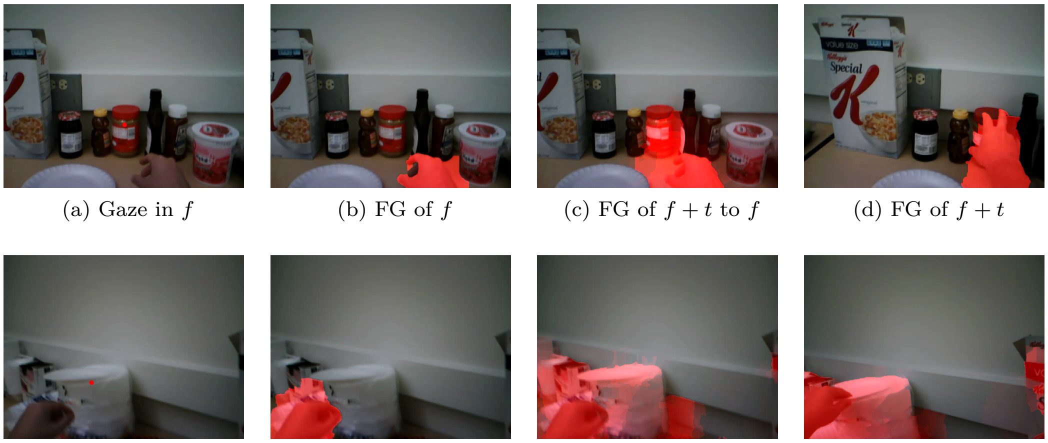
Our goal is to infer the action as well as the most likely sequence of gaze points in a test image sequence. The posterior probability of action a given the sequence of image features $X = \{x_1, ..., x_N\}$ is

$$p(a|X) \quad \propto \quad p(a, X) = \sum_G p(a, G, X) \approx p(a, G_a^*, X) \tag{5}$$

Since integration over all values of G is not practical, in Eq 5 we approximate $\sum_G p(a, G, X)$ with $p(a, G_a^*, X)$, where G_a^* is the most likely sequence of gaze locations given action a. If the action a is given, the graph in Fig 55(a) becomes an HMM in which the most likely sequence of gaze locations G_a^* can be computed using the max-product (Viterbi) algorithm. Given the computed most likely sequence of gaze locations for action a, $G_a^* = \{g_1^a, ..., g_N^a\}$, we have

$$p(a|X) \quad \propto \quad p(a) \prod_{t=1}^{N} p(x_t|a, g_t^a) \tag{6}$$

where we assume $p(a)$ to be a uniform distribution and $p(x_t|a, g_t^a)$ are estimated from the output of SVM classifier at location g_t^a as described above. Note that if the gaze locations were observed during the test, we could replace g_t^a in Eq 6 with observed gaze locations to compute $p(a|X)$.



(a) Gaze in f (b) FG of f (c) FG of $f + t$ to f (d) FG of $f + t$

(e) Gaze in f (f) FG of f (g) FG of $f + t$ to f (h) FG of $f + t$

Figure 56: This picture is best viewed in color. The gaze is usually a few frames ahead of hands. As a result, the foreground region a few frames later can provide a valuable cue for determining the gaze location in the current frame. We show two examples from initial frames of "take peanut-butter" and "take plate". The gaze falls on the object, while the hands have not reached to the object yet. In (a,e) the ground-truth gaze location in frame f of the action is shown. The computed foreground region in the frame f only contains the hand (b,f). However, when the foreground region from t frames later is transferred to this frame, it contains the gazed object (peanut-butter jar or plate) as well (c,g). The foreground region of frame $f + t$ is shown in (d,h).

9.3.3 Object Detection and Segmentation

Here we describe the details of the method we use for object detection and segmentation. Our framework is not dependent on the choice of object detector and can be applied to any possible object detection and segmentation method. However, to be clear about the details of our implementation here we describe the method used in this work.

We first use [8] to extract contours and use multiple thresholds to segment each frame into layers of regions. The lowest layer contains small super-pixels. Each super-pixel is included in bigger regions in the upper levels. In order to detect and segment the objects in each image, we learn a super-pixel classifier using SVM for each object type. For each super-pixel we concatenate the color and texture histogram of its containing regions, and the color and texture histogram of multiple circles with various radiuses around its center. We compute texture descriptors using the method of [177] and quantize them to 256 kmeans centers. We further extract color descriptors for each pixel and quantize them to 128 kmeans centers. We use a few manually segmented images from training set to learn a SVM super-pixel classifier for each object type. We learn 33 object classifiers in total, including a classifier for detecting the hands. As described in Sec 5.2.4, we use the learned object classifiers to build the object-based feature vector that captures the object context around a potential gaze point g_t. For each pixel in image, we concatenate the maximum scores of different object classifiers in its local neighborhood to build a feature vector.

9.4 Method II: Egocentric Gaze

In Section 9.3, we introduced a method for predicting gaze based on the close relationship between tasks and the gaze behavior. In particular, we showed that each task has a signature corresponding to the small circle around the gaze points during that action. In this section, we show that we can predict the gaze locations in egocentric video accurately, without the requirement of knowing the task that is being performed by the first-person. The work in this section is performed in collaboration with my colleague Yin Li.

Our main contribution in this section is leveraging the implicit cues that are provided by the first-person for predicting the gaze location in egocentric video. In particular, during object manipulation tasks, eyes, hands and the head of the first-person are in continual coordination within the context of current goal. We show that if we do not have access to the eye tracking measurements, still we can use head orientation, head movement and hand locations of the first-person to measure where her eyes are looking. For example, large head movement is almost always accompanied by a large gaze shift. Also, gaze point tends to fall on the objects that are held by first-person's hands. Since the camera is mounted on the subject's head, head orientation and movement are implicitly available from the video. In addition, hands can be segmented and tracked in video to help the estimation of gazed locations. These cues are very specific and unique to the egocentric vision setting.

We begin with the analysis of gaze tracking data from a wearable eye tracker and demonstrate that: (1) egocentric gaze is statistically different from on screen eye-tracking and (2) there exists a strong coordination between eye, head and hand movements in the object manipulation tasks. Moreover, we build a model for gaze prediction that accounts for eye-hand and eye-head coordinations, and combines them with temporal dynamics of gaze. The model requires no information of task or action, predicts gaze position at each frame and identifies fixations among them. Our gaze prediction results outperform several state-of-art bottom-up and top-down saliency detection algorithms by at least 4% on publicly available datasets.

9.4.1 Egocentric Cues for Gaze Prediction

The main contribution of this section is that different than the previous gaze prediction algorithms that rely on image features, we leverage the implicit cues that are provided by the first-person in egocentric setting to predict the locations of the gazed points. In egocentric setting, the camera is worn on the first-person's head, and as a result, the smallest head movement is captured in the video.

In addition, the images of the video are the best identifier of the first-person's head orientation in the world. Finally first-person's hands can be easily tracked in the video to identify where in the space the objects are being manipulated. In this section, we describe these implicit cues that are provided from first-person's body poses, and in section 9.4.2, we explain our method for predicting the locations of the gaze points based on these cues.

We apply our method to both of our publicly available gaze datasets, GTEA Gaze and GTEA Gaze+. Both of these datasets contain daily object manipulation activities in the context of meal preparation. These datasets contain egocentric videos overlaid by gaze measurements, captured using commercially available eye tracking glasses. In case of GTEA Gaze, we consider 17 egocentric video sequences from 14 subjects. In GTEA Gaze+, we use a subset of the dataset, including 15 egocentric video sequences from 5 subjects performing three different meal preparation activities. These two datasets cover a rich set of actions and contain over 3.5 hours of video in total.

Head Orientation: Several psychophysical experiments have indicated that eye gaze and head pose are closely coupled in daily activities [101, 102, 99, 73, 72], meaning that often gaze direction can be roughly approximated by head orientation. In the egocentric setting, the camera is mounted on the first-person's head, continuously capturing the scene in front of the first-person. Thus the center of each image in the video determines the location in the scene towards which the camera wearer's head is oriented. Now the question is how often the first-person's eyes will be fixating at what exist in the image center? Or in other words, how often an individual looks in the direction towards which her head is oriented?

Based on our analysis of the egocentric data, the gaze points are strongly biased towards the center of the images. Figure 57.(a) demonstrates the probability map of the gaze points in images of the GTEA Gaze and the GTEA Gaze+ datasets. Note that the nature of the center bias in egocentric data is different than that in on-screen gaze tracking. In egocentric data, the reason behind the center bias is that the subjects are more comfortable moving their head instead of moving their eyes, when they want to look at something. On the other hand, in on-screen gaze data, the head is almost always oriented towards the screen, and the reason behind the center bias is that often the stimuli appears in the center of the screen. We have compared the center bias in egocentric datasets with that in MIT's saliency dataset [85] in Figure 57.(a). Our observation is that the center bias is much stronger in egocentric data. Thus, head orientation provides a good approximation for the gaze direction. Note that the preference of the gaze point towards the bottom part of the image is a consequence of table-top object manipulation tasks.

Head Motion: We observe a strong correlation between the motion of the head and the location

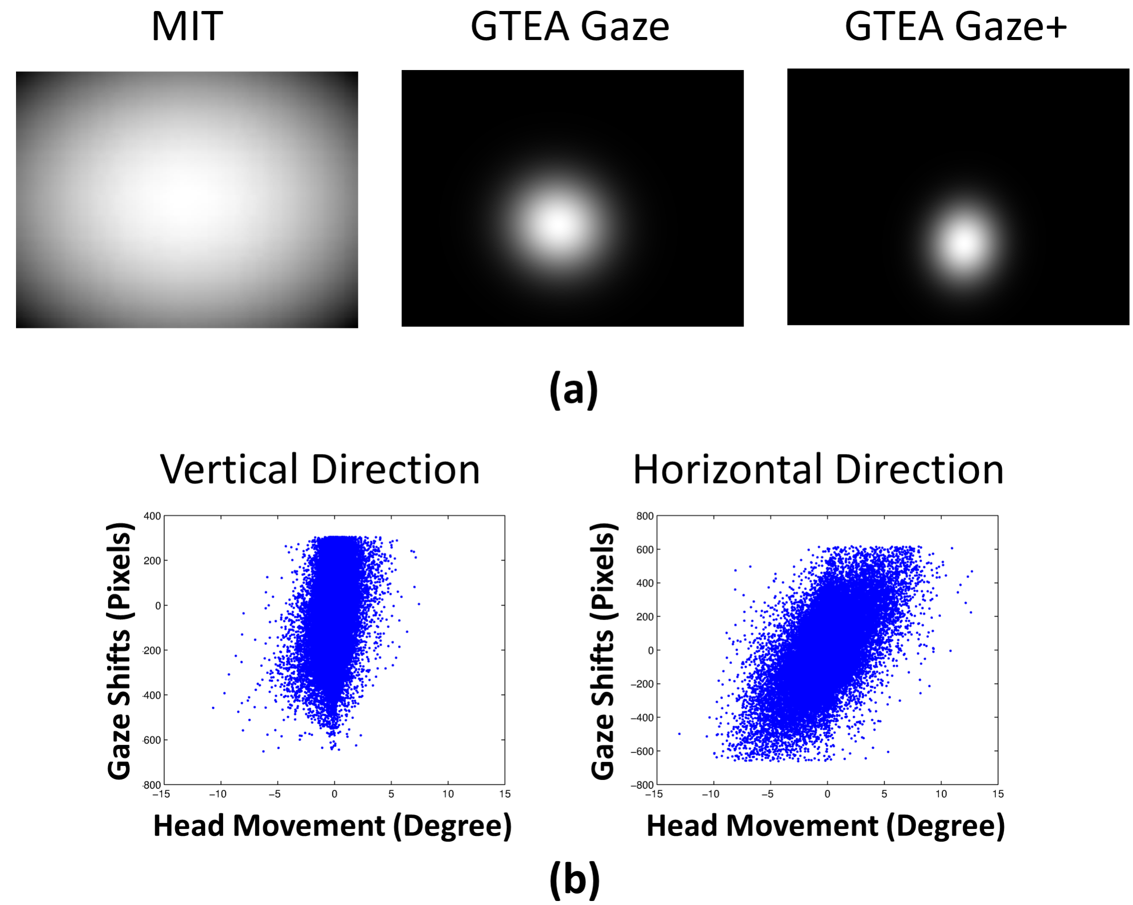


Figure 57: Individuals often look in the direction in which their head is oriented: (a) Center bias (from left to right) for MIT saliency dataset [85], GTEA Gaze dataset and GTEA Gaze+ dataset. Egocentric gaze has a much smaller variance in space. Thus, head orientation provides a good approximation for gaze direction in egocentric vision. Furthermore, head movement is associated with the movement of the gaze to the sides: (b) A scatter plot of head movement against gaze shift along vertical and horizontal direction in GTEA Gaze+ dataset. The plot suggest a linear correlation in the horizontal direction. This figure is generated by Yin Li.

of the gaze point in images. During the shift of attention to an object, the eyes make the move first and then the head starts following them. As a result, often when the head is rotating to the left (right), the location of the gaze point will also be moved to the left (right) in the image. The scatter plot of the gaze shift (from the center of the image) against the head motion for GTEA Gaze+ dataset is shown in Figure 57.(b). The plot suggests a linear correlation, especially for larger gaze shifts. As a result, we use the head movement as an additional feature for predicting gaze location in images.

We can estimate the head movement at each moment in time based on the background motion in the current frames of the video. To estimate the background motion of a frame, we first compute the large displacement optical flow (LDOF) [25] between that frame and the next ones. The reason

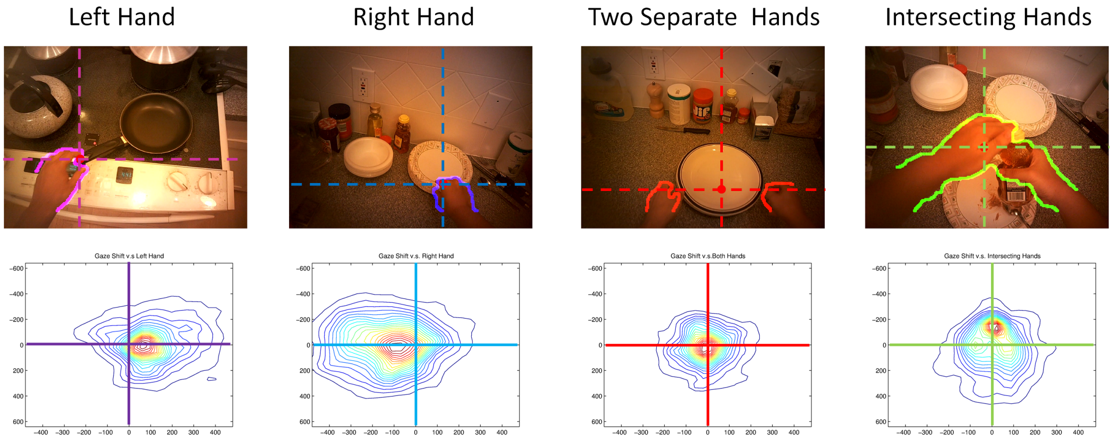


Figure 58: Top row: Hand segmentation and manipulation points (red dots). We present four different hand configurations and the correspondent manipulation points. The hands are colored by their configuration. Bottom row: Aligned gaze density map . We align the gaze points into the hand's coordinates by selecting the manipulation points as the origin, and projecting the gaze point into the new coordinate system every frame. We then plot the density map by averaging the aligned gaze points across all frames within the dataset. High density clusters can be found around the manipulation points, indicating spatial structures for eye-hand coordination. This figure is generated by Yin Li.

for using LDOF is that the head can move very fast sometimes in egocentric video, resulting in large displacements between adjacent images. We remove the pixels that correspond to the hand segments (described later), and use the flow in the remaining pixels to compute the mean 2D background motion vector. Head movement is then approximated by inverse tangent of the background motion vector divided by the camera focal length. The approximation, albeit simple, provides a reasonable estimate.

Hand Location: Eye-hand coordination is the key to successful object manipulation tasks. Eye gaze generally leads the movement of the hands to the target [100]. In turn, it has also been shown [148] that the proprioception of limbs may lead to gaze shift, where the hands are used to guide eye movements. We align gaze points with respect to the first-person's hands and discover clusters in the aligned gaze density map, suggesting a strong eye-hand coordination. In the rest of this section, we will first describe how to detect the left and the right hands, and then we will use the detected hands for predicting the locations of the gazed points in images.

Detecting the Hands: Our goal is to first segment the hands from the background and then discriminate between the regions corresponding to the left hand and the ones corresponding to the right hand. We extract dense texture features as described in the textonboost [159] framework for the pixels of an image. Then we train a dense CRF [95] on top of these features to segment hands in each image. Now that we have segmented the regions that correspond to the hands, in the next step we assign each hand region to either left hand, right hand, or both hands intersecting with each other. For each region we extract spatial and shape features including the location of the centroid, aspect ratio and the area of the hand masks. We then train multiple linear SVMs on top of these features to assign the hand region to one of the three mentioned choices. In addition, we force mutual exclusiveness between the hand region labels. For example, if there exist two hand regions in an image, only one of them can be left hand.

Gaze with respect to Hands: In order to estimate the location of gazed point given the hand regions, we first define a control point for each of the labeled hand regions. The control point corresponds to the location at which the hands will be most likely to manipulate the objects. For example, for the left hand, manipulation usually happens at the right tip of the hand region. For a region corresponding to both hands intersecting each other, the manipulation usually happens in the middle of the two hands. We select the control point as the centroid of all the extremas on the top part of the hand region boundary. A few examples are shown in Figure 58. The control point provides an anchor with respect to current hand pose, and allows us to align gaze point into the hand's coordinate.

Having set the manipulation (control) point as the origin, the density map of the aligned gaze points for four different hand configurations are plotted in Figure 58. We observe high density around the manipulation point in both GTEA Gaze and GTEA Gaze+ datasets. For example, given a region corresponding to the left hand, the gaze tends to fall on top right of the manipulation point, since that is where often the manipulated object appears. Similarly, for two separate hands, subjects are more likely to look in the middle, where the object often appears. For a region corresponding to left and right hand intersecting with each other, the gaze shifts towards the bottom, partly due to the opening/closing actions. We use the hand manipulation points as an additional feature for predicting the location of the gazed point.

9.4.2 Gaze Prediction

We model each of the cues corresponding to head orientation, head motion and hand location as a Gaussian. Given each image, we predict gaze as a mixture of these three Gaussians. Parameters are learned from the training data. We model the head orientation prior with a 2D Gaussian distribution $G_c = \mathcal{N}(\mu_c, \Sigma_c)$ where μ_c is the 2D center location and Σ_c is a 2×2 covariance matrix. We model the head motion prior with a 2D Gaussian distribution $G_m = \mathcal{N}(\mu_m, \Sigma_m)$ where we have

$$\begin{pmatrix} \mu_m^x \\ \mu_m^y \end{pmatrix} = \begin{pmatrix} a_1 & 0 & a_3 \\ 0 & a_2 & a_4 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix}$$

where μ_m^x and μ_m^y are the horizontal and vertical locations of the distribution mean in the image, v_x and v_y are the horizontal and vertical axis of the average optical flow vectors, and a_1, a_2, a_3, a_4 are parameters learned from the data. Finally, 2D Gaussians are learned for each of the four hand configurations as described in Section 9.4.1.

9.5 Experiment 4: Gaze and Action

In this section we present experimental results on our first dataset (GTEA Gaze). Results on the second dataset (GTEA Gaze+) can be found in the following url: `http://cpl.cc.gatech.edu/projects/GTEA_Gaze/`.

9.5.1 Action given Gaze

Recognition of daily actions has its own challenges that are different than those in traditional action recognition settings. Daily actions consist of a verb and one or more object names. As a result, object context plays an important role in discriminating different actions. This makes the recognition task easier since the action verb and objects can provide context for each other as shown in Chapter 7, but at the same time the task becomes harder since miss detection of an object can result in a wrong action label. Furthermore, detection of objects in the background as part of the foreground can lead to wrong action labels.

Another challenge in recognizing daily actions is that a simple action like “open peanut-butter jar” can be performed by completely different motion patterns. One might hold the jar by left hand and open it with right hand, one might leave the jar on the table and use one hand to open it, etc. Given all these variations in ways of performing an action, still the appearance of the area around the gaze point is usually consistent between different subjects performing the same action. Focusing at the neighborhood of the gaze location lets us get rid of those variations and leads to significant improvement in action recognition accuracy.

As described in Sec 9.3.2, for the case of observed gaze, we compute the probability of $p(a|X)$ using Eq 6 by replacing g_t^a with given gaze locations in frame t. Our method achieves 47% accuracy on action recognition compared to 27% accuracy of the method of Chapter 7. Random classification chance for 25 classes is 4%. We show the confusion matrix for recognition of different actions in Fig

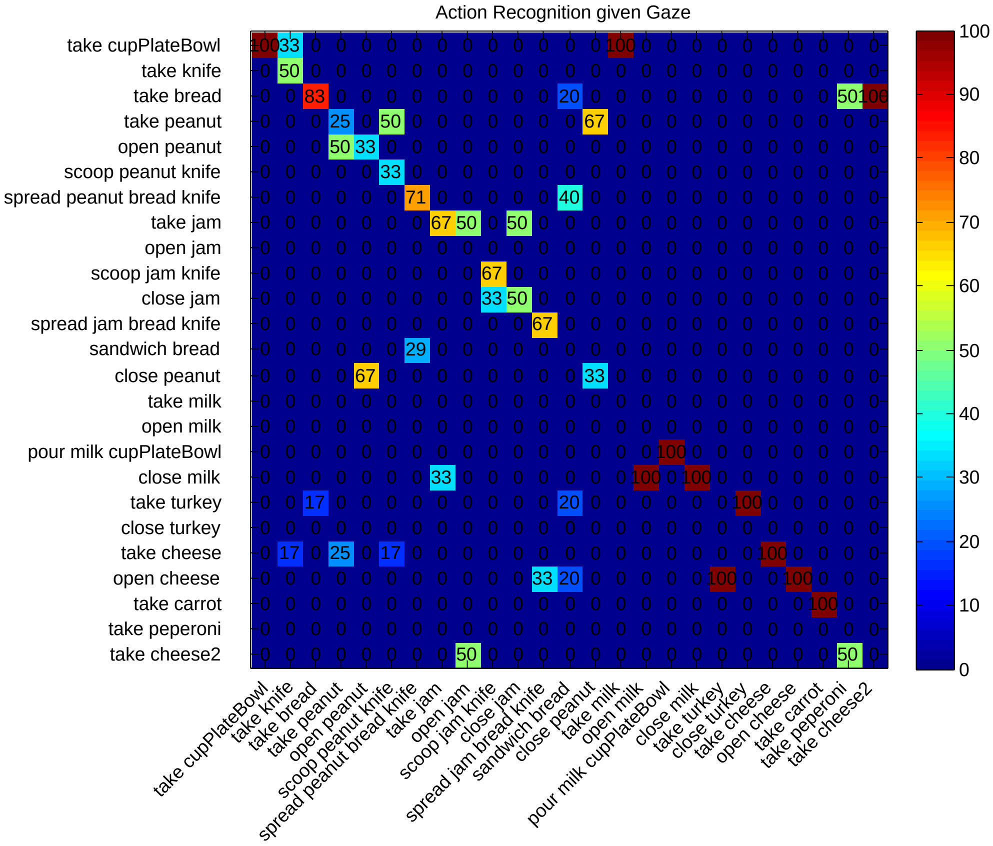


Figure 59: This figure is best viewed in color. Confusion matrix for recognizing actions given the gaze locations in each frame. Gaze information significantly improves action recognition. The average accuracy is 47% which is significantly higher than 27% accuracy achieved by the method in Chapter 7. Random classification chance is 4%.

59. We believe there are three main reason for the failure in recognizing some of the actions: (1) our algorithm does not take the temporal cues into account. For example, it is impossible to tell the difference between opening a jar from closing a jar given only one image. We use HMM to model the likelihood of the action in each frame and gaze transitions. However, HMM does not capture the change in the appearance of the circle at different stages of the action; (2) some objects have similar color and texture which confuses our method; and (3) it is not always the case the gaze point falls were the action is taking place.

We compare our results to that of Chapter 7 in Fig 60. The method of Chapter 7 first segment the foreground from background, then use a semi-supervised learning method to detect objects, and then extract features from hands and objects to perform action recognition. To make the comparison fair, since we use pre-learned object classifiers, we provide their method with our object classifiers as well.

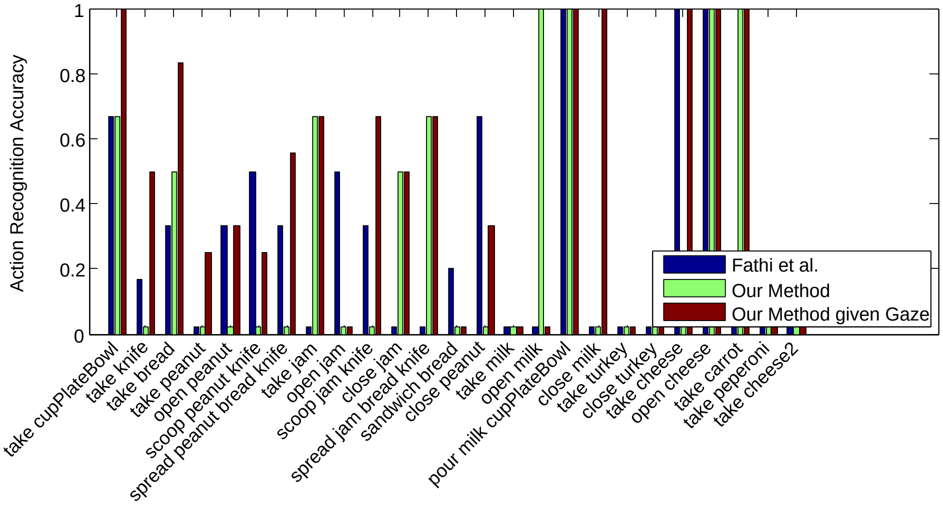
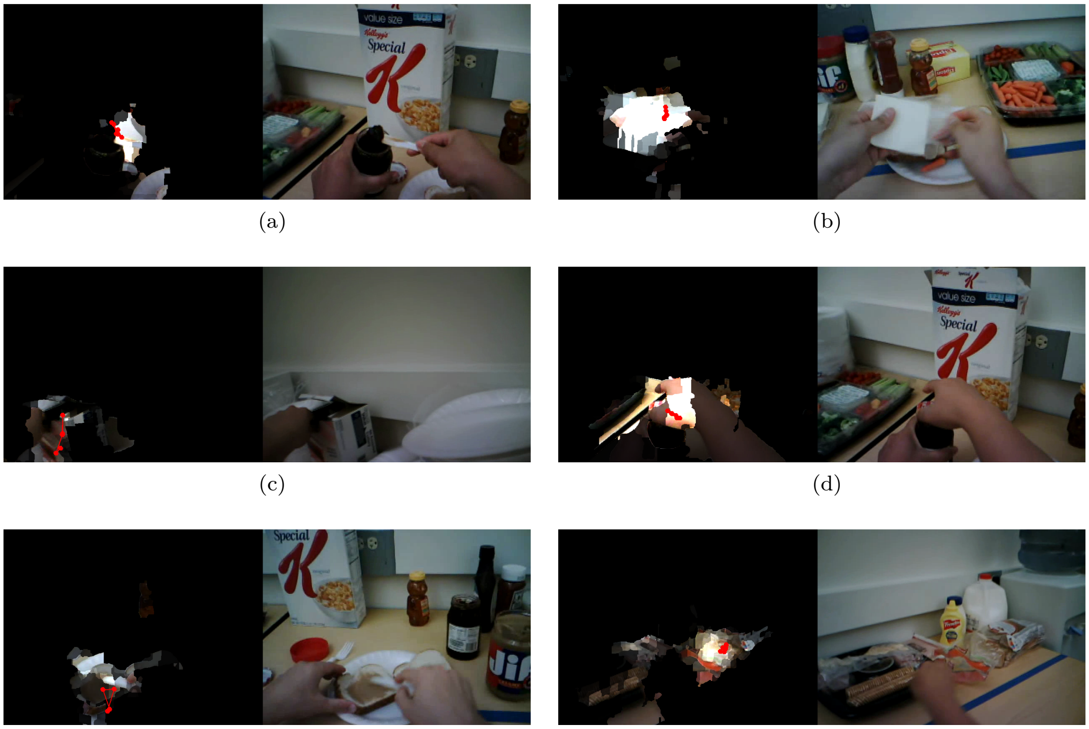


Figure 60: We compare our action recognition results with and without gaze observed during the test with our results in Chapter 7 which is marked by Fathi et al. [50] in the figure. Our method with observed gaze achieves 47% average accuracy. Our method that simultaneously infers gaze and action reaches 29% accuracy. The method of Chapter 7 gets 27% accuracy. The classification accuracy by chance is 4% for 25 classes.

9.5.2 Method I: Gaze given Action

The task provides a rich context for prediction of gaze location in images and video. Different subjects have a very consistent gaze pattern while performing the same action. We build a classifier that predicts human attention during performance of a particular action. We compute the likelihood of every pixel in image corresponding to gaze location by applying the classifier to feature vector extracted for that pixel location. In Fig 61 we show example outputs of our classifier. The pixels belonging to the action are scored higher than background pixels. In Fig 62, we show that our method significantly achieves better results in comparison to general saliency methods [82] that only use low-level image features. Note that we understand that this might not be a fair comparison since our results are generated by knowing the action label for the image. The main point of our results is that (1) if the action is known, the gaze can be predicted with a good precision and (2) we show an evidence that gaze and action are closely tied together, and use this finding to justify our framework.

Each gaze prediction method in Fig 62 outputs a saliency map, in which each pixel location in the image is assigned a score. We measure the accuracy of a method by computing the percentage of the pixel scores that are lower than the pixel score of ground-truth gaze location. For example, assume the ground-truth gaze location falls at a pixel with score 0.9. If 75% of the pixels in the image are assigned scores less than 0.9, then the accuracy of the gaze prediction method for that



(a) (b) (c) (d) (e) (f)

Figure 61: Our method predicts fixation locations in images for each particular action. The right hand side pictures show the frame and the left hand side images show our prediction results. The brighter the pixels are it means the higher the score returned by our algorithm is. The red dots show the ground-truth gaze locations from few adjacent frames. The actions are (a) scoop jam using knife, (b) open cheese, (c) take knife and (d) open jam, (e) spread peanut on bread using knife and (f) take bread.

frame is 75%. We average the accuracy over all frames belonging to the action and report them in Fig 62.

9.5.3 Method I: Simultaneous Inference

There are multiple reasons that motivate us to develop a method that works without having gaze data as well: (1) eye-tracking glasses are very expensive, need calibration , and still are not user friendly enough to be put on for more than a few minutes. We can learn parameters of our model from the data captured by eye-tracking glasses and then apply it to the data captured by cheap wearable cameras as well, (2) comparison of computed gaze locations with actual human data might lead to diagnosis of attention problems, measure the level of expertise and be used for human computer interaction and (3) simultaneous prediction of gaze and action demonstrates the close relationship between the two.

We use the inference method described in Sec 9.3.2 to recognize actions and estimate the gaze

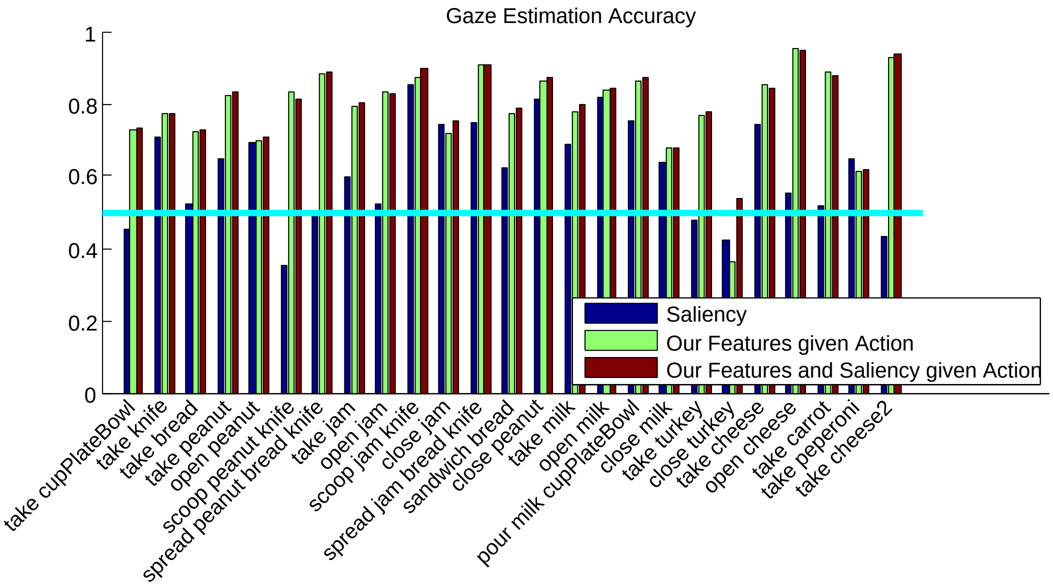


Figure 62: This figure is best viewed in color. The task plays an important role in predicting the gaze behavior. Saliency based methods which only use low-level features are not able to capture the task related attention. Knowing the action significantly improves the results of gaze prediction. The saliency [82] at gaze location is on average higher than the saliency at 60% of the other points in image. Our classifier's score at gaze location is on average better than 81.3% of the classification scores at other image locations. Combination of low-level features used by [82] with our features only slightly improves results to 81.9%, which means the higher level action knowledge plays a more important role for predicting where humans attend. Random chance is 50% shown by the cyan line.

location in each image sequence. We show our results in Fig 60. Our method achieves 29% accuracy compared to the method of Chapter 7 that achieves 27% accuracy. The accuracy of random classification by chance is 4%.

9.5.4 Method II: Gaze Prediction

We benchmark the gaze prediction results of method II in Section 9.4 on both GTEA Gaze and GTEA Gaze+ datasets. We show that our method significantly outperforms state-of-art bottom-up and top-down saliency detection algorithm by at least 5%. We first test our gaze prediction results. We compare our results with three bottom-up saliency detection algorithms (Itti and Koch [82], GBVS [71], Hou and Zhang [78]) and the top-down saliency results of Method I in Section 9.5.2. Itti and Koch [82] and GBVS [71] could handle both image and video as the input. We turn on the motion model for a fair comparison. Hou and Zhang [78] predict saliency given a single image. In addition, the Method I requires the action label as the input. In comparison, the Method II does not rely on low level features or action labels.

Our results are based on counting the average percentage of the pixel scores that are lower than

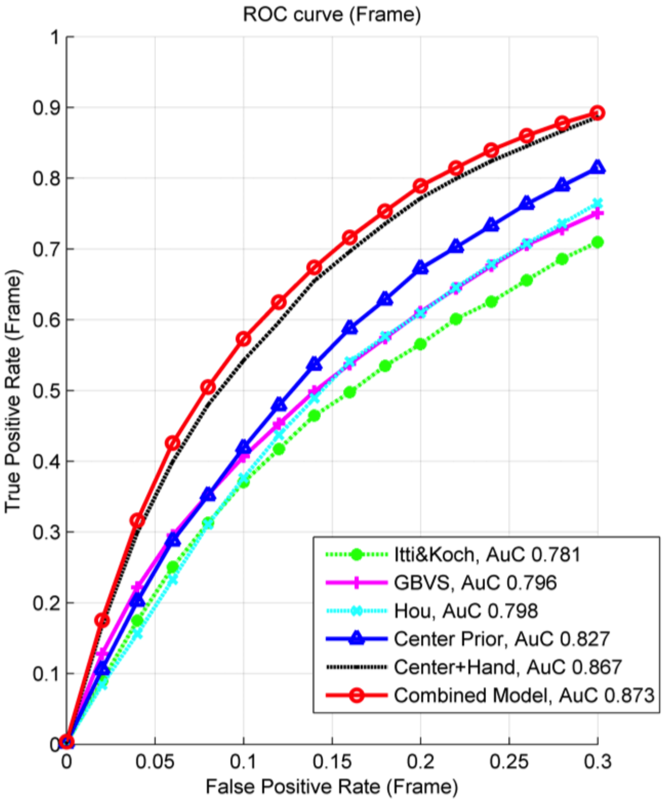


Figure 63: ROC curve for GTEA Gaze dataset. Our method with all cues achieves 87.3%. Center Prior performs surprisingly well (82.7%), beating all bottom-up methods. The results in this figure are produced by Yin Li.

the pixel score of ground-truth gaze location This is equal to a per image (Area under ROC Curve) AuC score. Overall, Method II achieved a score of 87.3% on GTEA Gaze, where Method I gives 81.9% by using the ground truth action labels and low-level features. Results are shown in Figure 63. Bottom-up saliency algorithms, by their nature, could not work well in egocentric setting. This is due to the fact that first person attention is heavily modulated by the current task.

We also tested Method II in GTEA+ dataset. 7 out of 15 videos are used for training our model. Since our model is simple, reducing the number of training sessions would have little influence over the performance. Method II achieved a score of 92.7% on GTEA Gaze+, outperforming all the bottom-up approaches. Results are showing in Figure 64.

9.5.5 Method II: Action Recognition

We extract features from circles around predicted gaze points to recognize actions. We achieve 41.5% accuracy on action recognition results which is close to the action recognition results given groundtruth gaze locations (47% accuracy). A comparison between Method II, action given gaze

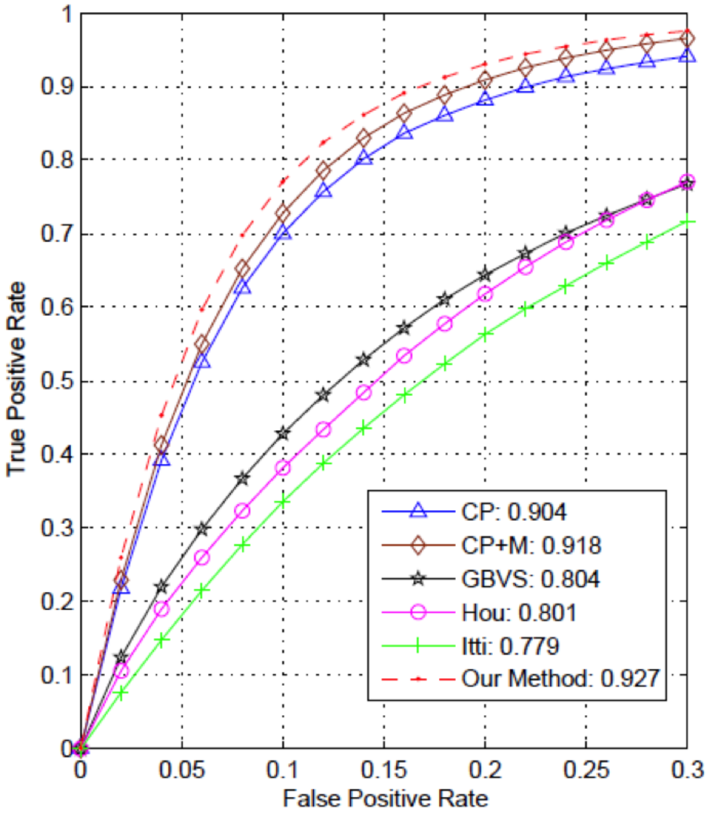


Figure 64: ROC curve for GTEA Gaze+ dataset. Our method with all cues achieves 92.7%. Center Prior (CP) performs surprisingly well (90.4%), beating all bottom-up methods. The results in this figure are produced by Yin Li.

and Method I is shown in Figure 65.

9.5.6 Action Recognition with Center Prior

To make sure our gaze prediction is improving the results, we also compare it to action recognition using a circle around center prior. Action recognition given a circle at the center of the image results in 21% accuracy. Furthermore, the accuracy given the circles at the center prior is 25%.

9.6 Conclusion and Future Work

In this chapter, we studied the relationship between gaze and action. Our results confirm the psychological findings of [100, 74] that the subjects often gaze at the objects that are relevant to the current task. We leveraged gaze measurements to improve the action recognition results. Furthermore, we proposed two methods for predicting gaze in absence of measurements. We showed that first-person's head movement and hand locations provide strong cues for predicting gaze location in egocentric vision. Currently, our results are limited to the domain of meal-preparation activities.

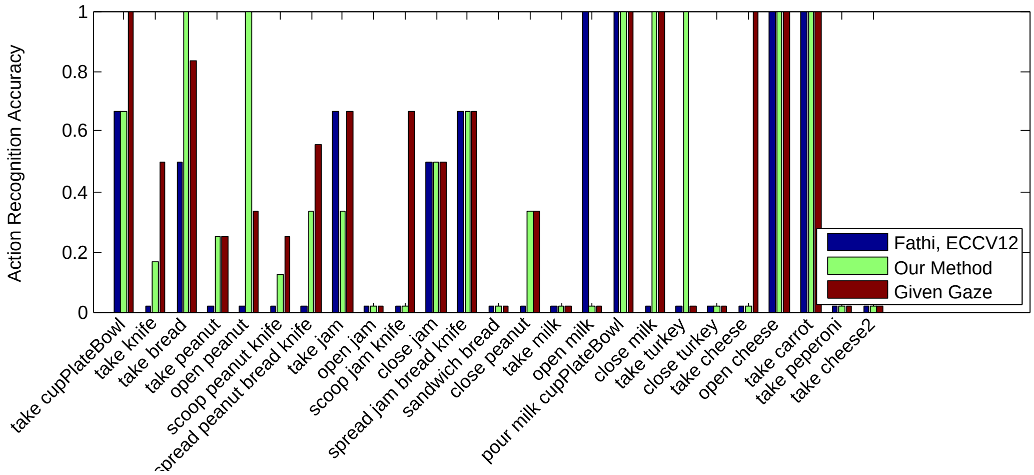


Figure 65: We compare action recognition results of Method II with that of Method I. In the figure, Method I is referred to as "Fathi, ECCV12" and Method II is referred to as "Our Method". For 7 out of 25 classes, Method II outperforms Method I, leading to a 12.5% improvement over the average accuracy. The overall accuracy for Method I is 41.5%.

However, we plan to expand our dataset to contain other kinds of daily activities such as laundry, grooming, hygiene and mending things.

A limitation of our current datasets is that they are collected in a single location. As a result, there is little variety in the appearance of the objects and the environment throughout the collected videos. This might be advantageous since it helps to design simple algorithms for recognizing object instances and instead focus more on understanding the value of the gaze in egocentric video.

In this chapter, we introduced two methods for gaze prediction in egocentric video. The first method used high-level cues such as objects and foreground region for predicting gaze location in the video. The second method leveraged cues from first-person's body like head orientation, head motion and hand location to predict the gaze. Our experiments were performed on meal-preparation activities. Theoretically, the first approach is limited to object-manipulation tasks. However, the second approach can theoretically generalize to other domains.

9.6.1 Benefits of the Egocentric Vision

Studying human gaze in the context of daily activities is only possible using egocentric vision. There are works that study human gaze in a static setting. However, gaze behavior during watching a movie is very different than the gaze behavior in the real life setting. In addition, we showed that it is possible to leverage the head orientation, head movement and hand locations to predict first-person's gaze direction in the egocentric setting.

Chapter X

CONCLUSION AND FUTURE WORK

The thesis characterizes the benefits of egocentric vision, which leverages wearable camera technology and provides a new line of attack on classical computer vision problems such as object categorization and activity recognition.

The dominant paradigm for object and activity recognition over the last decade has been based on using the web. In this paradigm, in order to learn a model for an object category like "coffee jar", various images of that object type are fetched from the web (e.g. through Google image search), features are extracted and then classifiers are learned. This paradigm has led to great advances in the field and has produced state-of-the-art results for object recognition. However, it has two main shortcomings: a) objects on the web appear in isolation and they miss the context of daily usage and b) web data does not represent what we see every day.

This thesis, demonstrates that egocentric vision can address these limitations as an alternative paradigm. Contextual cues and the actions of a user are exploited in an egocentric vision system to learn models of objects under very weak supervision. In addition, measurements of a subject's gaze during object manipulation tasks provide novel feature representations to support activity recognition. Moving beyond surface-level categorization, a method is designed for automatically discovering object state changes during actions, and an approach is proposed for building descriptive models of social interactions between groups of individuals. These new capabilities for egocentric video analysis will enable new applications in life logging, elder care, human-robot interaction, developmental screening, augmented reality and social media.

10.1 Contributions

Here we revisit the thesis statement and summarize the contributions of this work.

Thesis Statement: *Descriptive models* of objects and activities can be learned in a weakly supervised setting by leveraging *attentional cues* that are available in egocentric video.

The main contributions of this thesis are captured in two keywords: "descriptive models" and "attentional cues". Our goal is to go beyond recognizing activities by building richer descriptions based on their building blocks which are actions, hands, objects and faces. In order to achieve this goal, we leverage first-person's attentional cues to help our algorithms focus on the important areas of the images and videos.

10.1.1 Object Discovery in Egocentric Video

We used the attentional cues that are provided by first-person's hand movements to discover objects from everyday activities. The key to recognizing daily activities is being able to recognize the objects that are used in them. We realized that it is hard to learn models for all the existing objects in the world. On the other hand, we noticed that each person only uses a small subset of objects in daily life. Thus, we designed an adaptive method that learns daily objects from first-person's actions. We used the fact that handled objects move together with first-person's hands to carve them out of egocentric video and learn models of them. We further developed a weakly supervised learning method that labels the carved objects based on the patterns of object co-occurrence in activities. For example, "cup" is an object that often occurs in activity of "making coffee", but it rarely is used in activity of "making cheese sandwich". The same way that detecting objects is crucial for recognizing daily activities, activities also provide significant context for recognizing objects. We designed a graphical model that infers objects, actions and activities together and results in better performance in comparison to recognizing each independently.

Most of the previous work on weakly supervised object recognition target the image domain. Vijayanarasimhan and Grauman [180] use multiple instance learning to train object classifiers from the results of web search. Kuettel et al. [96] start with an initial set of object segmentations and propagate it to other images in ImageNet. In comparison to these works, we leverage a) the motion discontinuity cues which are available in video and b) location prior and contextual cues which are available in egocentric vision to segment the active region from the background. On the other hand, object labels given for a video contain much weaker supervision in comparison to object labels for images. The difference is that in the video, not only the location of the object in the frames is unknown, but also the time during which the object appears in the frames is also unknown. Our method can handle this by using the patterns of object co-occurrence in activities, and also by leveraging the capability of segmenting the active region.

10.1.2 Modeling Daily Actions through Object State Changes

We describe objects based on their states. In addition, we model actions based on how they cause the state of the objects to change. Over the last two decades various approaches have been taken to model and understand actions in computer vision. The common theme among all those works is that they model an action by encoding motion and appearance. However, actions with similar motion patterns and hand-object relationships can have different meanings because they accomplish different goals. For example, “open coffee jar” and “close coffee jar” are two different actions, in fact they are inverse. However, they produce very similar motion patterns and involve the same object. The key to distinguishing these two actions is to be able to detect the state of the “coffee jar” (open vs. close) and how it is changed by these actions. Based on this observation, we introduced a method for recognizing daily actions based on analyzing and modeling the state of objects and materials before and after the action is performed. We developed a weakly supervised method for learning the object and material states that are necessary for recognizing daily actions.

10.1.3 Using Gaze for Recognizing Daily Actions

We leveraged gaze as the finest-grained attentional cue to achieve higher performance in action recognition. We made progress towards answering the following questions regarding gaze and actions: Where do people look when they perform a particular action? Can we predict gaze in an egocentric setting? Can we learn from where people look to develop more efficient and more robust recognition algorithms?

We showed that since humans often look at the manipulated objects, knowing gaze can significantly improve action recognition results in egocentric setting. In comparison to Yi and Ballard [193], we treated gaze as a latent variable. When our model has access to gaze measurements, uses them, and when the gaze points are unknown, it predicts the gaze and action simultaneously.

We further questioned the need for gaze measurements in the context of daily activities. We showed that we can use two sets of high-level cues to predict gaze in egocentric setting: 1) first-person’s movements and 2) patterns of environment change. We showed that humans often look in the direction in which their head is oriented. Moreover, when the head is rotating to the left or right, the gaze starts moving to the left or right side of the image as well. In addition, during meal-preparation activities, humans often look at the objects that are being manipulated by their hands. We demonstrated that it is possible to recognize first-person’s actions based on features that are extracted from a small neighborhood of gaze points. Using these cues, we achieved accurate gaze prediction results. We further showed that we can recognize actions using the predicted gaze and get results close to if we had access to the ground-truth gaze measurements.

10.1.4 Modeling Social Interactions using Patterns of Attention

We were the first to propose the problem of recognizing social interactions in first-person videos. We developed a method that analyzes each frame of the video and estimates where the individuals are attending in the scene. We showed that the patterns of attention are strong cues for distinguishing the classes of social interaction.

Furthermore, we developed a first-person system that detects the moments of eye-contact between an adult wearing eye-tracking glasses and a toddler. We showed that detecting eye-contact in egocentric video is easier in comparison to videos recorded from static cameras. Our research is motivated by the fact that social behaviors such as making eye-contact and shift of attention provide significant cues for understanding developmental disorders like Autism.

10.2 Applications

Our research aims at enabling some of the applications that are described in this section. The current state of our algorithms is far from reaching these goals, but we believe that our current research direction is towards making these dreams into reality.

10.2.1 Activities of Daily Living

A system that can automatically interpret the daily routine of an individual will enable important applications including elder care, developmental screening, life logging and developmental robotics (Figure 66). In the U.S., persons aging 65 or older represent about 13% of the population [2]. Most elders would prefer to continue to live in their own homes. However, unfortunately, the majority of elderly people gradually lose functioning capabilities and require additional assistance in the home. A framework that can monitor and understand human daily activities is a first step towards an automated care giving system that can assist elder people at home. In addition, a system that can recognize foods and activities can automatically estimate the amount of calories, carbohydrate, and fat that a person eats throughout everyday.

In order to enable this application, we should be able to (1) recognize daily objects such as foods, cups and pills; (2) recognize daily activities of a person such as drinking, eating and cooking; and (3) predict the user's intentions and goals. Most of our work in previous chapters of this thesis are headed towards reaching these goals. However, there is still a large gap between the current state of our algorithms and achieving a robust performance on these tasks. We believe that the depth information can significantly improve the performance of our algorithms. We plan to use RGB-D cameras for this purpose in our future works.

10.2.2 Developmental Screening

There is a similar need for a daily activity recognition system that screens the development of children in their living space. In the U.S., 50% of children with developmental disabilities lose an important window for early treatment because their conditions is not identified until they start school [1]. Analyzing children's social behavior can help psychologists to recognize developmental disorders. The current screening methods are extremely labor intensive and are often based on a short interaction in the examiner's office. The main reason is that there are not enough experts for examining the large number of children who are under risk. A wearable egocentric system that can automatically observe and analyze the everyday behaviors of children can be extremely helpful for addressing these limitations.

Unfortunately, we have not yet found a device that children (in particular the ones on the spectrum) will tolerate wearing. In addition, such wearable device needs to be safe and robust to touch and unexpected movements. As a result, we have studied the scenario in which the eye-tracking device is worn by an examiner that interacts with the child. In Chapter 4, we showed that we can detect the moments of eye-contact between the examiner and the child. However, there are much more required for a thorough understanding of the social behavior than detecting eye-contact. As a future work, our plan is to recognize shift of attention to objects and faces. Furthermore, we are interested in better understanding the complexity of children's play with toys.

10.2.3 Interesting Moments

Another application of egocentric vision is finding interesting moments in a day-long video. For example, Jim Rehg was at Disney parks with his family, wearing an egocentric camera and pushing his son in the stroller. All of a sudden a stranger shows up out of blue and hands his son a balloon. Can we develop an automatic system that can find such interesting moments in a day-long video of a person hanging out at a park? Another application is developmental robotics. It is interesting if we can build a system that understands how a person solves a task and can transfer that knowledge to a robot that wants to solve the same task.

We are still very far from achieving this goal. In order to find interesting moments, not only we have to recognize the actions, people, objects and the scenes, but also we have to learn what types of moments are appealing to a person. A big challenge is being able to generate ground-truth for the degree to which a moment is interesting. We still do not have a solution for measuring the interestingness of a moment. However, we think such measurements need to be done both based on general opinions and also based on individual preferences.

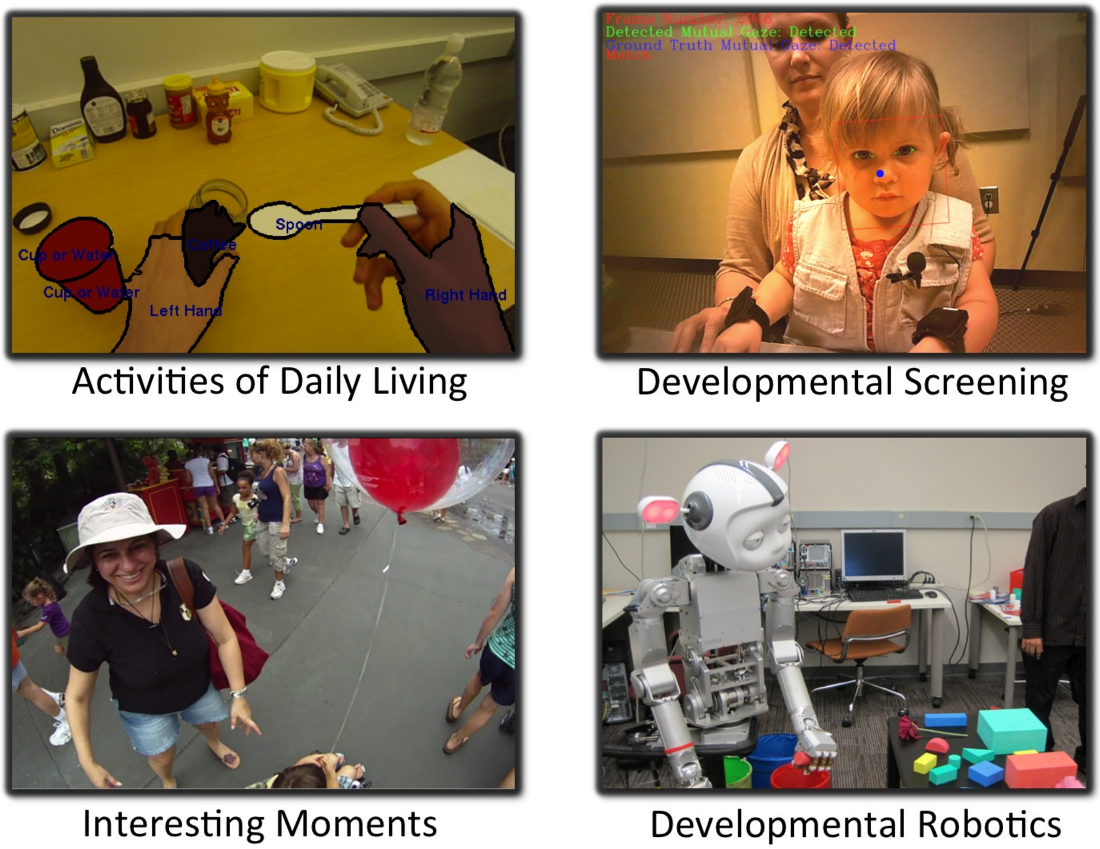


Figure 66: A system that can automatically understand what is happening around an individual at every moment can enable important applications.

10.2.4 Other Applications

Another possible application of egocentric vision is developmental robotics. The goal is to create an intelligent system that observes how a person solves a task and transfers this knowledge to a robot that wants to solve the same task. This application is not yet addressed in our works.

We believe that the state of computer vision in recognizing objects and activities is still far from the capability of current face detection and recognition algorithms. Thus, we think we are closer in enabling the applications that involve faces and their interaction in comparison to applications that deal with objects.

Applications of egocentric vision go beyond the ones mentioned above. We believe egocentric data can play an important role in advancing computer vision technology. First-person vision enables capturing thousands of hours of videos of objects that are used and activities that take place throughout people's daily lives. Although egocentric videos are not naturally associated with tags and meta data as images and videos on the web are, but instead, they are highly structured. First-person's fixations and hand movements provide strong cues that can be leveraged to carve the active objects and faces from egocentric videos. Furthermore, the videos are always being recorded in the context of the place and the time at which the first-person is located.

10.3 Discussion

In this section, we will describe our intuition and view on some of the key questions regarding the egocentric vision.

10.3.1 What is an Egocentric Camera?

A very important and key question is that "when a video can be called egocentric?". Is a video that is recorded by a handy-cam egocentric? Is a video that is recorded by a chest mounted camera egocentric? How does a chest mounted camera compare to a head mounted camera? There is another closely related question which is: what is the benefit of an egocentric video in comparison to a video that is recorded from a static camera? Here we describe our point of view on what an egocentric video is and what benefits it has.

We believe an egocentric vision system has three main characteristics:

1. An egocentric camera continuously captures the daily activities of the first-person. It rarely misses something that happens in the scene in front of the camera wearer. This is not true about a static camera that is mounted in a kitchen or in a park. Why is this important? We believe this is very important for (1) learning how actions cause changes to the state of the objects in the environment; and (2) learning the series of actions that need to be performed in order to complete an activity. In our GTEA Gaze+ dataset, subjects perform meal-preparation activities in a kitchen. We mounted 3 Asus RGBD cameras around the kitchen to capture all the actions of the subjects. Each of these cameras could capture a subset of these actions, but still there were actions that were missed by all the cameras. For example, when the person is taking something from the fridge, her body occludes the view of all of these cameras. The perception of these cameras from the world is that the subject goes to the fridge, it is not possible to observe what is going on, and all of a sudden there are some objects in subject's hands. As a result, it frequently happens that there is a change in the environment, but the action that caused that change is not observed by these cameras. This means that the activities are only partially observed, and there are missing points in between them. Now imagine a day-long activity of a person. Things happen at home, on the bus, in the street, in the park

and in the office. There is no way that the static cameras could capture all of the actions.

2. An egocentric camera is personal. It means that this camera always captures the experience of a particular person who is wearing it. This is not true for a static camera. The personal diary of the activities is an important source of context for studying the behaviors of an individual. In this thesis, we have not leveraged this fact, but we are very interested in exploring it in the future.

3. An egocentric vision system can potentially use the attentional cues of the camera wearer to improve object and action recognition results. For example, it can use eye-tracking technology to measure where the first-person is looking at each moment in time. Attentional cues not only can improve recognition and prediction accuracy, but also can create greater cognitive understandings about human vision.

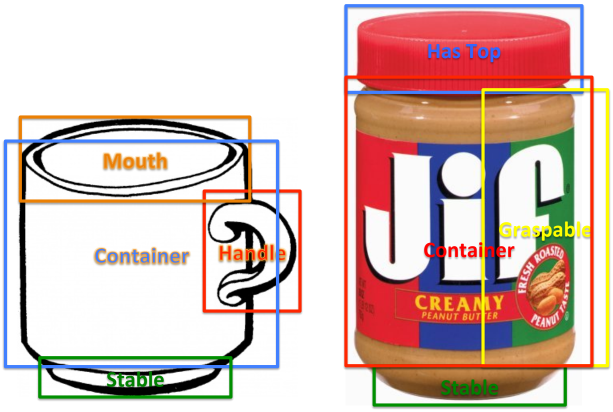
Given these benefits, we are convinced that egocentric vision is useful for (1) advancing the computer vision as a field; and (2) for enabling various applications of computer vision in our daily lives. Once we have agreed on this point, the question that “what tasks become easier in egocentric view in comparison to allocentric view?” becomes less relevant. Instead the question becomes “how can we extract the rich information that is embedded in egocentric video?”. This thesis’s main goal is to answer the latter question.

Regarding the question of “is a head mounted camera better or a chest mounted camera?”, we do not know the answer. This question is similar to asking “would it be better if we had our eyes on our chest instead of where they are?”. Maybe it was. These two settings are similar since they both can continuously capture the scene in front of the first-person. They are both personal solutions as well. However, the only difference is that the head mounted camera can be augmented with the eye-tracking technology and capture first-person’s attentional cues.

10.4 Open Questions and Future Work

10.4.1 Object Affordances

our goal is to build descriptive models of objects and actions from egocentric video. Throughout the thesis, we demonstrated the close relationship between actions and objects: 1) we modeled actions as the interactions between objects and hands and 2) we modeled actions as processes that change the state of the objects. As a result, we showed that in order to better understand the actions, we need to go beyond surface level recognition and build rich and descriptive models of objects. We demonstrated a method for learning the state of the objects. Yet, another descriptive property of objects that connects them to actions is the notion of affordance. Affordances of an object determine the actions that are possible to perform on that object. For example, a cup affords "pouring" which means it is possible to pour liquid into a cup.



(a) (b)

Figure 67: We believe an object can be modeled based on a deformable spatial relationship between its affordances. The affordances are not enough in order to completely recover the object label, but determine the high level category to which the object belongs.

Some affordances are possible to perceive from visual information. For example, a cup is "graspable", and that is obvious by looking at an image of a cup because it has a handle. As a result, often there are features, shapes or parts on an object that identify its affordances, as shown in Figure 67.

As a future work, we propose to detect the affordances of the objects from the videos of daily actions. Our goal is to develop a method that goes through videos of an action (e.g. pouring) and finds the features that are common between the objects that afford that action. We believe depth is a very important feature for recognizing object affordances. For example, depth can help to verify if an object has a container shape in order to know whether it affords pouring. As a result, we propose to collect a dataset of daily activities using a wearable RGB-D camera for this purpose.

10.4.2 Big Data from 1000s of Egocentric Vision Systems

Within a year, thousands of individuals will be wearing Google glass. This translates to tens of thousands of hours of video per day. The main question we want to answer is that how this data can help us approach human-level object and activity recognition? This data will contain the visual experience of thousands of people and will capture a large set of real world objects in the context of daily use. Obviously, annotation of this data is beyond human capability. Thus, there will be a need for unsupervised or weakly supervised learning techniques that can provide annotation for this data. We propose to investigate these methods for learning models for objects and activities in egocentric video with minimum supervision.

10.4.3 Visual Memory

An egocentric camera captures a large amount of visual data everyday. An interesting future direction is building a visual memex out of this data. We believe, we need to address three problems for building a visual memory:

- Discovering and indexing the important objects and faces
- Summarization and novelty detection
- Adapting to the user's preferences

In particular, we are very interested in the last bullet point. We believe memorableness or interestingness of a moment is a subjective matter. For example, a subject might find the moments on a roller coaster more interesting while another subject might find moments with family and friends more interesting. We propose to design learning algorithms that adapt to the preferences of a subject.

10.4.4 Health Related Applications

We plan to pursue health-related applications of egocentric vision. In particular, we are interested in designing applications that address elder care and developmental screening. For this purpose, we suggest to fuse other modalities such as sound with vision. For example, identifying the person talking in a social setting using voice analysis can result in a more reliable recognition of patterns of turn taking and can improve our previous work on recognizing social interactions. Other than sound, we would like to measure movement, depth, location, identity of objects, and first-person's internal state. For example, Q-sensors by Affectiva Inc. can provide first-person's electrodermal activity (skin conductance) which increases when the user is experiencing psychological or physiological arousal whether engaged, stressed or excited. We plan to use these measurements not only to make a more robust egocentric system, but also we like to leverage them to enhance current computer vision algorithms.

Appendix A

SEMI-SUPERVISED VIDEO SEGMENTATION

Video object segmentation is an important problem in video analysis, and it has many applications, including post-production, special effects, object recognition, object tracking, and video compression. In particular, the rapidly-growing numbers of videos which are available from the web represent an opportunity for new video applications and analysis methods.

A key motivation here is the observation that video-based object tracking and interactive video object segmentation are closely-related problems. In both cases, the goal is to segment an object from the video with a minimum number of annotations provided by the user. In tracking systems, the user identifies the object in the first frame (e.g. by drawing a contour around it) and it is tracked automatically in the remaining frames [33, 18]. However, these methods can fail in cases of occlusion or due to a severe change in object appearance. In contrast, recent systems for interactive video object segmentation provide very fine-grained control via a well-designed interface, making it possible for a user to get pixel-perfect segmentations [13, 141, 196].

Our goal is to address both of these problems within the same framework. In a tracking context, our approach makes it easy to fix up an existing solution by carving the object labeled in the first frame through out the video. Applications to biotracking are a motivating example, as there is a strong need for a general purpose tool for tracking a wide range of animals with different morphologies. In contrast to video post-production, in biotracking applications segmentations which are not pixel-perfect but which delineate the limbs of the animal (for example) can still be useful for animal behavior experiments.

Likewise, in the context of interactive video object cutout, our approach leverages constraints on video data and an active learning approach to minimize the number of annotations that must be supplied by the user. In case of biotracking, it is necessary to minimize the need for guidance by the user in order to have a useful tool for biologists. As a result our method aims at getting the most benefit from each user click. To achieve this goal, we develop an active learning method which chooses the most important frames to be labeled, and which guides the user in each frame about where to click.

Our framework is based on casting video segmentation as a semi-supervised learning problem with video-specific structures such as temporal coherence which makes the following contributions:

A framework for object cutout and interactive segmentation: In this work we present a framework which addresses video object cutout and has a natural extension to interactive segmentation. We use semi-supervised learning to propagate labels from known locations to unknown, and *uncertainty* is a key in a effective label propagation. Furthermore, uncertainty is incorporated into active learning to guide the user in annotating the video. Tsai et al [174] formalize tracking as a video object cutout where only the first frame was annotated. A limitation of their approach is that it is not clear how to make additional annotations effectively. We propose a new framework that addresses this problem.

Incremental self-training: We develop an iterative solution to semi-supervised video segmentation. At each step, we pick the least uncertain frame, fix all labels in the frame, and update system parameters (e.g. object appearance models). We show that incremental self-training is very effective in adapting to video content, outperforming standard semi-supervised learning and state-of-the-art tracking systems.

Intelligent user guidance: We develop a systematic way to provide intelligent guidance to users in an interactive setting. This is by selecting the most informative frames to be labeled by user, and guiding the user while labeling each frame.

The method described in this chapter was presented in our BMVC 2011 paper [49].

A.1 Previous Work

A.1.1 Semi-Supervised Learning

Zhu [202] provides a survey on semi-supervised learning methods in machine learning. Other examples of semi-supervised learning applied in computer vision include image classification [66] and tracking [198]. For segmentation, [181] showed a few examples of interactive segmentation, and [178] applied multiple instance learning to semantic segmentation. Furthermore Badrinarayanan and others [12, 26] use an EM based semi-supervised learning algorithm to propagate labels from the start and end of a video.

A.1.2 Active Learning

Active learning is another growing sub-field in learning (detailed survey provided by Settles [156]). Zhu et. al [203] present an active-learning approach based on risk minimization of unlabeled nodes using harmonic functions. In this work, we demonstrate that uncertainty leads to a better solution than risk minimization for video segmentation. Other examples of active learning include classifying video [188], object categorization [89] and image co-segmentation [15]. Kohli and Torr [94] introduced a method for estimating uncertainty for graph-cut through an expensive post-processing which can be used for active learning. In this chapter, we introduce an interactive segmentation method which we use for annotating training data. Our method use a continuous formulation using harmonic functions which computes soft labels and naturally provides uncertainty.

A.1.3 Object Tracking

Significant progress has been made recently on object tracking by various approaches where [36, 10, 18, 33] are a few representatives. Many of these trackers can produce object segmentations [150, 18, 33, 174]. Ren and Malik [150] showed that segmentation helped avoid drifting on long sports videos. Bibby and Reid [18] demonstrated the adaptation of target models and integration of multiple terms to track a wide range of targets. Chockalingam et. al [33] developed a level-set based system which tracks the target by dividing it into multiple regions combining spatially constrained appearance model and motion estimation. Tsai et al [174] developed a multi-label MRF model for offline tracking solved with Fast-PD. Our use of self-training in tracking is novel and yields promising results comparing to the state of the art [33, 174]. In addition we propose an interactive segmentation framework.

A.1.4 Interactive Segmentation

There is a huge literature on interactive image segmentation. The most relevant to our work is that of Grady's [65], which uses random walks for label prediction in images. For interactive video segmentation, a popular toolkit is LabelMe Video [196], where the user uses polygons to delineate objects. More complicated interactive trackers [183, 13] show segmentation results frame by frame, and let the user fix errors. These works are focused on pixel-perfect segmentations and user interface designs. In comparison, our goal is to intelligently guide the user to where to annotate and thus minimize user effort for each specific video. We propose a method that achieves a close to perfect segmentation after getting a few annotations from the user.

A.2 Video Segmentation Framework

The goal of this work is to address object tracking and interactive video segmentation in a unified framework. We want to enable robust extraction of objects from challenging videos (with large changes in shape, appearance and motion) using minimal user input. One application is biotracking: video segmentations desired in biotracking may not need to be pixel-perfect, but it has to handle a wide range of animals with very different morphologies, and it has to do with minimal effort from non-expert users.

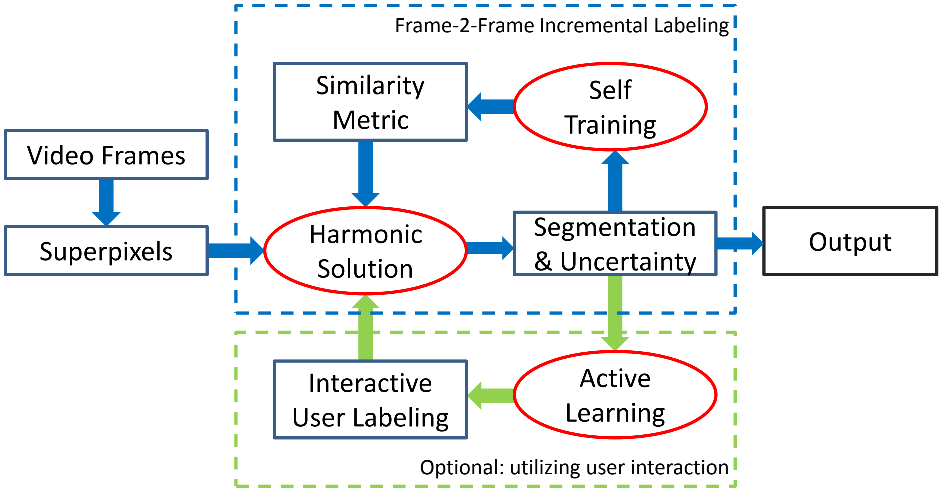

Frame-2-Frame Incremental Labeling
Similarity Metric
Self Training
Video Frames
Superpixels
Harmonic Solution
Segmentation & Uncertainty
Output
Interactive User Labeling
Active Learning
Optional: utilizing user interaction

Figure 68: A flow chart illustration of our approach.

Figure 68 gives the flow chart for our approach. We formulate video segmentation as a semi-supervised learning problem on a graph of super-pixels. Harmonic functions provide an efficient solution to the graph labeling problem and produce soft labels, which we use to measure labeling uncertainties at both the super-pixel and the frame level. We then iteratively choose the least uncertain (or most certain) frame in the video, discretize the soft labels, and apply self-training to update similarity metrics and the affinity graph. We will demonstrate that incremental self-training significantly improves segmentation accuracy in comparison to standard baselines and state of the art segmentation-based tracking methods.

Our approach has a natural extension to interactive segmentation. We can present the current segmentation to the user and ask for more input. To minimize user effort, we devise an effective scheme to intelligently suggest the user which frame and which super-pixel to label. Our scheme is empirically validated through simulation.

In the rest of this section we describe the semi-supervised learning formulation for video segmentation, its harmonic solution, and the underlying graph structures.

A.2.1 Semi-supervised learning and segmentation

Video object segmentation can be naturally viewed as a semi-supervised learning problem over a graph. In our case we treat super-pixels as data points, and our goal is to propagate the information from labeled ones to others. In a semi-supervised learning setting, there are $n = l + u$ data points x_i (l labeled and u unlabeled points, typically $u \gg l$), and the goal is to label the unlabeled points. One can formulate this problem on a graph $G = (V, E)$ with nodes $V = L + U$ and edges E, where

L are the labeled nodes and U the unlabeled ones. In our approach, we represent connectivities in this graph by a $n \times n$ symmetric matrix W, where w_{ij} represents the similarity or affinity between two nodes x_i and x_j:

$$w_{ij} = exp(-g_{ij}^T \Sigma g_{ij}) \tag{7}$$

where $g_{ij} = x_i - x_j$ and Σ is the inverse covariance matrix. We assume Σ is diagonal and contains hyper-parameters to scale the elements of g. In Section A.3.1, we present a method for learning the weights in Σ.

The labeling problem in G can be solved using min-cut [19]. Zhu et. al [204] propose a framework which relaxes the min-cut objective function and leads to a simple algorithm with interesting behaviors. They look for a harmonic function $f : V \rightarrow \mathbb{R}$ that is real-valued on the unlabeled data U, but constrained to be $f_L \in \{0, 1\}$ on the labeled data L. The intuition behind the harmonic solution is that it returns the probabilities of starting from unlabeled nodes and arriving at nodes with a particular label by randomly walking in the graph.

The harmonic solution f can be computed in polynomial time by simple matrix operations [204]. It is possible to represent the combinatorial laplacian $\Delta = D - W$ as four blocks separating labeled and unlabeled nodes:

$$\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix}$$

and similarly split f into labeled (f_l) and unlabeled (f_u) parts. The harmonic solution subject to $f|_L = f_l$ is

$$f_u = \Delta_{uu}^{-1} \Delta_{ul} f_l \tag{8}$$

The formulation above considers only pairwise edges. It is easy to incorporate priors or unary edges: add an auxiliary node for each class, and connect it to all the nodes using priors as edge weights.

The harmonic solution has several key advantages for video segmentation comparing to mincut. It provides a real-valued solution where uncertainty (and related quantities such as risk) can be easily computed and optimized, crucial to our approach. Furthermore, examples in [65] and our experiments show that it tends to be less sensitive to noise and weak boundaries.

A.2.2 Graph construction for video segmentation

The graph G we use for video object segmentation has three types of edges: (1) spatial, (2) temporal and (3) prior edges. We model spatial coherencies using appearance and boundary saliency, and model temporal coherency using appearance and optical flow.

Spatial edges. Spatial edges connect the adjacent super-pixels inside each frame. Super-pixels have different sizes and shapes. We represent each super-pixel by its color and texture histograms. Texture descriptors [177] are computed for each pixel and quantized into the 256 nearest k-means centers. Similarly, color descriptors for each pixel are quantized into 128 k-means bins. The distance g^s_{ij} between a pair of super-pixels is the difference between their color and texture histograms. We concatenate g^s_{ij} with the saliency of the shared contour between i and j as in [8], obtaining a 385 dimensional feature vector. We compress these vectors to 64 dimensions using PCA. We multiply the spatial weights $w^s_{ij} = exp(-(g^s_{ij})^T \Sigma^s g^s_{ij})$ by the length of the shared contour to compensate for size differences between super-pixels.

Temporal edges. We compute dense optical flow between adjacent frames using [24]. A super-pixel i is linked with a super-pixel j in the adjacent frame if optical flow connects some of their underlying pixels. Similar to the spatial edges, we use color and texture histograms to compute the distance g^t_{ij} (384 dimensional) between i and j and compress the acquired vectors to 64 dimensions using PCA. We observe that the appearance-based distance g^t_{ij} can compensate for errors in motion estimation. We multiply $w^t_{ij} = exp(-(g^t_{ij})^T \Sigma^t g^t_{ij})$ by the number of corresponding pixels between the super-pixels as given by the flow field. In addition to optical flow, we also compute sparse SIFT features[114] correspondence. If we find SIFT correspondences between two super-pixels (i.e. the number of inliers above a threshold), we increment w_{ij} with a large constant.

Prior edges. The spatial and temporal edges capture the similarity between adjacent super-pixels. To add priors for super-pixels belonging to each of the K classes, we add K labeled nodes, one for each label class, to the graph and connect all the unlabeled nodes to them. We learn the weights for these connections as $w^k_i = exp(-(g_i)^T \Sigma^k g_i)$, where g_i contains the color and texture histograms for super-pixel i.

We find that spatial and temporal edges are sufficient in many cases to produce good results. In some cases (e.g. the penguin sequence in SegTrack dataset [33, 174]), putting a significant weight on the object appearance model can hurt segmentation accuracy, because the object looks similar to the background. On the other hand, for some videos with inaccurate optical flow, it is necessary to have prior knowledge about object appearance.

The final edge weight W is a linear combination of the individual weights $W = W^s + \alpha W^t + \beta(W^0 + \cdots + W^{K-1})$. Our system learns and updates the similarity metrics Σ^s, Σ^t, $\Sigma^0, ..., \Sigma^{K-1}$ iteratively in the self-training process, thereby automatically adapting the features to the particular sequence.

A.3 Incremental Labeling using Self-Training

In the previous section, we presented a semi-supervised learning formulation of video segmentation and showed how to construct the graph over super-pixels in all video frames. Directly solving for the unlabeled nodes over this graph, through either mincut or the harmonic solution, does not work well (see Table 4). This is partly because it is impossible to find a set of parameters (in particular feature weights Σ) that work for all videos, and partly because objects change appearance over time and temporal links between frames can be noisy for large motions.

To address these issues, we develop an incremental approach where we iteratively label frames and apply self-training to adapt model parameters. At each step, we compute the harmonic solution, choose the least uncertain frame and fix (discretize) the labels in that frame. Afterwards, we use linear regression to update and balance the feature weights Σ.

A.3.1 Per-frame self-training

Self-training is a commonly used semi-supervised learning technique [202] which trains a classifier on the labeled data and applies it on the unlabeled data. In our studies, we find that self-training, when used in the context of spatial-temporal coherencies in a video graph, can lead to much better results than direct graph-based solution of mincut.

What is special about videos? Videos consist of a sequence of frames ordered in time. Video frames are closely coupled within themselves, and the associations between frames are "Markov". The object appearance in a frame is usually closest to its appearance in adjacent frames. As a result, conditioned on the labels in a frame, the labels in adjacent frames can be predicted with a high certainty.

We leverage the special structures in videos to assign labels iteratively. At each step, we compute the uncertainty of each unlabeled frame, and select the one with least uncertainty. This frame is often the one adjacent to the last labeled frame. We fix (discretize through thresholding) the labels in this new frame given the harmonic solution. After adding these nodes in the new frame to the set of labeled nodes L, we update the spatial, temporal and unary similarity metrics (Σ^s, Σ^t and $\Sigma^0, ..., \Sigma^{K-1}$) using the labels f_L. We use the new metrics to update the adjacency matrix W and

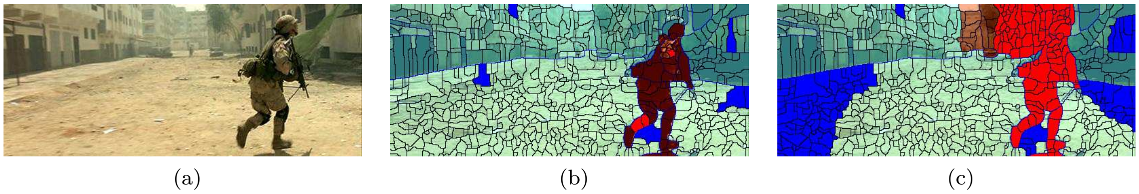


Figure 69: Self-training fails on a per-superpixel basis, showing the need for frame-level inference. (b) Segmentation based on harmonic solution (user labels in red and blue). (c) Superpixel based self-training fails after 100 iterations. Bright red and blue show labels fixed after self-training. Per-frame self-training works well on this sequence.

the Laplacian Δ, estimate the new soft labels f_u, fix the labels for the next least uncertain frame, and iterate. In each iteration the final labels for a new frame are fixed, and the algorithm iterates until all frames are labeled.

The frame uncertainty (entropy) H of a frame $\mathcal{F}$ is calculated by adding the uncertainty of all super-pixels belonging to $\mathcal{F}$ weighted by their size:

$$H(\mathcal{F}) \quad = \quad \sum_{i \in \mathcal{F}} H(f_i)\mathcal{S}(i) \tag{9}$$

where $\mathcal{S}(i)$ is the area of super-pixel i and $H(f_i)$ is its entropy:

$$H(f_i) \quad = \quad \sum_{k=0}^{K-1} -log(f(i,k))f(i,k) \tag{10}$$

A plausible alternative, which ignores the video structures, is to iteratively apply thresholding and self-training, but at the super-pixel level (i.e. fixing the label for the least uncertain super-pixel at each step) instead of the frame level. This strategy often results in poor segmentations; an example can be found in Fig 69.

A.3.2 Metric learning and weight balancing

The role of the inverse covariance matrices Σ is to assign weights to the features in g, such as color vs texture, to capture intra similarities and inter differences between object and background super-pixels. A perfect metric, both for spatial or temporal edges, has the following property

$$w_{ij} = e^{-g_{ij}^T \Sigma g_{ij}} = \begin{cases} 1 & if \ f_i = f_j \\ 0 & if \ f_i \neq f_j \end{cases} \tag{11}$$

(a)

(b)

(c)

Table 4: We compare our tracking results with [33, 174] using the average number of errors (pixels) per frame. Our results outperforms these methods in most of the sequences, and achieve marginal results to [174] in the rest. We further compare our method to graph-cut and graph-cut with self-training given the same set of parameters.

Sequence	GraphCut	GraphCut + Self Training	Ours	[33]	[174]	Average Object Size	Number of Frames
Parachute	254	253	251	502	**235**	3683	51
Girl	4121	1616	**1206**	1755	1304	8160	21
Monkey-dog	1312	3727	598	683	**563**	1440	71
Penguin	19569	19569	**1367**	6627	1705	20028	42
Bird-fall	454	766	342	454	**252**	495	30
Cheetah	1961	898	**711**	1217	1142	1584	29
Soldier	3344	2484	**1368**	2984	2228	6321	32
Monkey-water	1306	1266	**1009**	4142	2814	6011	31

where f_i,$f_j \in \{0, \cdots, K-1\}$ are the labels. It means we want the similarity weight between two super-pixels belonging to the same object to be high. Assuming that Σ is diagonal, it is more convenient to write $w_{ij} = exp(-\sum_{d=1}^{m} \sigma(d) g_{ij}(d)^2)$ where $\sigma(d)$ is the d-th diagonal element of Σ, and $g_{ij}(d)$ is the d-th element of the feature difference $g_{ij} = x_i - x_j$.

It is very hard, if not impossible, to find a set of weights $\sigma(d)$ that can work for a variety of videos. On the other hand, once the system has seen a specific video and acquired some labels, it can search for a custom metric Σ which satisfies the property of Eq 11 over the known labels. That is, we want $\Sigma_{d=1}^{m} \sigma(d) g_{ij}(d)^2$ to be 0 for the case of $w_{ij} = 1$ and ∞ for $w_{ij} = 0$. We use linear regression to estimate the $\sigma(d)$'s and find that regression works well with a limited amount of training data. Similarly, for unary metrics Σ^k, if super-pixel i in the labeled data belongs to class k, we force w_i^k to be 1 and 0 otherwise, and the metrics $\Sigma^0, \cdots, \Sigma^{K-1}$ are also learned through linear regression.

A.4 Assisted Interactive Segmentation

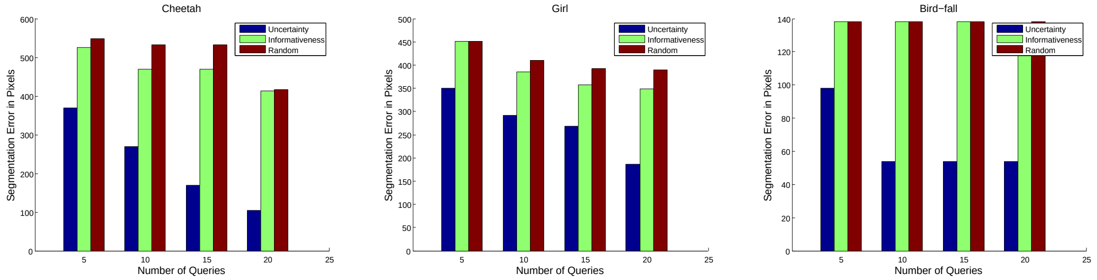
In this section we introduce an efficient interactive framework which achieves a close to perfect video segmentation with the minimum amount of user input. Our interactive segmentation method consists of two steps. In the first step our algorithm selects the most informative frame from the set of unlabeled frames and asks the user to label it. Then in the second step, our algorithm guides the user in annotating the frame by iteratively suggesting the next best super-pixel to be labeled.

A.4.1 Selecting the most informative frame

Previous interactive segmentation algorithms[13, 141] start from the first frame of the video and ask the user to annotate frames sequentially one after the other. This is necessary for acquiring a pixel-perfect segmentation. However, our goal is to reach a close to perfect video segmentation with as few user inputs as possible. As a result we intend to incorporate the user input for only a few

Table 5: The results of asking the user to annotate the most uncertain frames versus annotating the frames sequentially and randomly. Our method often achieve a better segmentation error given the same amount of user effort. In all cases we simulate annotating 5 frames. We compute the error based on the area of difference between ground-truth super-pixel labels and the labels computed by different schemes.

Method	Parachute	Girl	Monkey-dog	Penguin	Bird-fall	Cheetah	Soldier	Monkey
Most Uncertain	115	366	520	296	4	343	538	410
Random	122	412	344	329	4	428	769	513
Sequential	132	464	530	1171	45	445	1011	864
Initial Error	145	595	536	1291	45	497	1035	887



(a) (b) (c)

Figure 70: We show average number of erroneous pixels in a random frame, after t iterations of interactively labeling super-pixels for 3 representative sequences. We have compared the average segmentation error per frame given each method after $5, 10, 15, 20$ queries.

frames instead of getting the user to annotate every single frame. This means that we have to seek a smart way of selecting those few frames in order to acquire the most useful information from the user.

Here we propose an active learning strategy based on selecting the most uncertain frame $\mathcal{F}$ at each step and we show this method outperforms random or sequential frame selection approaches. Frame uncertainty is computed in the same way as in self-training (described in Eq 9). After the user is satisfied and finished with labeling a frame $\mathcal{F}$, the harmonic solution is updated to incorporate the new information, and the procedure is repeated. We compare the user effort required in this scheme against the approach of asking the user to annotate the frames sequentially or randomly.

A.4.2 Selecting the most informative super-pixel

When labeling each frame, our algorithm iteratively suggests the next superpixel to be labeled by the user. The user is asked to perform a click at each iteration: left click for object and right click for background. We explore different strategies for selecting the next super-pixel. We first evaluate a standard active learning technique based on risk minimization. Zhu et. al [203] compute the expected risk as $\sum_{k=1}^{K} f(i,k)\hat{\mathcal{R}}(f^{+x(x_i,k)})$, where $\hat{\mathcal{R}}(f^{+x(x_i,k)})$ is the risk after adding node i with

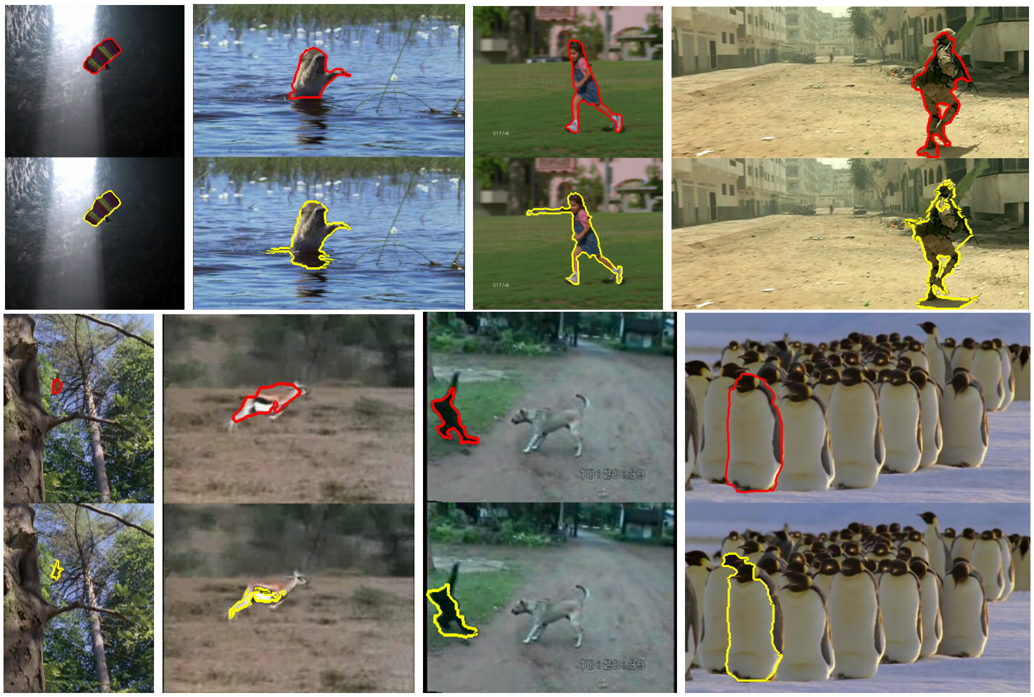


Figure 71: We qualitatively compare our results with [174]. Our results are shown with red contours, and the results from [174] with yellow contours. The sequences in the first row are *parachute*, *monkey-water*, *girl*, *soldier* and the ones in second row are *birdfall*, *cheetah*, *monkey-dog* and *penguin*. The contours produced by our algorithm are smoother, and the segmentations are usually more accurate.

label k. The risk is defined and estimated in [203] as the generalization error of Bayes classifier. We can suggest to the user the superpixel which minimizes the risk. Empirically, we find that risk minimization does not produce good suggestions, likely because the risk estimation is often inaccurate.

The second strategy we consider is querying the superpixel with the highest uncertainty. The most uncertain nodes are always located near the boundary between classes in the current labeling. We found this method more effective for our problem. The reason is that once the algorithm reaches a close to accurate segmentation, uncertainty based method queries super-pixels at the ambiguous areas on object boundary and refines the object mask. We compare these strategies with randomly selecting the next super-pixel.

A.5 Experiments

We provide results both on object tracking and on interactive video segmentation. We perform our experiments on 8 challenging sequences with ground-truth segmentation [33, 174]: parachute, girl, monkey-dog, penguin, bird-fall, cheetah, soldier and monkey-water. The first six are the videos used in SegTrack dataset [174] in order to fairly compare our work with previous methods. The length of these sequences are between 21 to 71 frames. In Table 4 we show that our approach produces better segmentations than [33], and also compares favorably to [174]. We further show qualitative comparisons in Figure 71.

As a baseline, in Table 4 we also compare to a standard graphcut solution and graph-cut combined with our self-training method using the same adjacency graph. In the results of standard graph-cut, the object mask either shrinks or expands over time. The self-training method is meant to solve this issue by iteratively segmenting frames one after the other. The combination of graph-cut and self-training produces better results than graph-cut alone, however still the shrinking bias in graph-cut is an issue. We show that the combination of self-training and harmonic solution produces the best result.

For interactive segmentation, we compare different strategies for frame selection in Table 5, and those for superpixel selection in Figure 70. To compare different frame selection strategies we simulate the user behavior using the ground-truth segmentation. In each iteration we select the next frame based on the scheme criteria (the most uncertain frame, random, sequential), use the ground truth to label it, and recalculate the segmentation. We empirically compare the three methods in Table 5.

We compare super-pixel selection strategies in Fig 70. We start with the first frame labeled, and simulate different strategies using the ground-truth labels instead of getting input from an actual user. Selecting the next super-pixel based on uncertainty outperforms both the risk minimization proposed in [203] and random selection. The reason is that given the first frame mask, the object is segmented throughout the video with a high accuracy, and the uncertainty based strategy queries super-pixels at the object boundary and can improve the result faster.

REFERENCES

[1] in *Center for Disease Control and Prevention, Developmental Screening, http://www.cdc.gov/ncbddd/child/devtool.htm.* 126

[2] "Aging statistics," in *U.S Department of Health and Human Services*, 2010. 125

[3] Aghazadeh, O., Sullivan, J., and Carlsson, S., "Novelty detection from an ego-centric perspective," in *CVPR*, 2011. 20

[4] Ahmad, S., "Visit: a neural model of covert attention," in *NIPS*, 1991. 19

[5] Allen, J. F., "Towards a general theory of action and time," in *Artificial Intelligence*, 1984. 78

[6] Andrews, S., Tsochantaridis, I., and Hofmann, T., "Support vector machines for multiple-instance learning," in *NIPS*, 2003. xii, 14, 66, 68

[7] Aner, A. and Kender, J., "Video summaries through mosaic-based shot and scene clustering," 18

[8] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J., "From contours to regions: an empirical evaluation," in *CVPR*, 2009. 59, 83, 90, 108, 137

[9] Aris, A., Gemmell, J., and Lueder, R., "Exploiting location and time for photo search and storytelling in mylifebits," in *Technical Report, MSR-TR-2004-102*, 2004. 20

[10] Avidan, S., "Ensemble tracking," in *CVPR*, 2005. 134

[11] Babaguchi, N., Ohara, K., and Ogura, T., "Learning personal preference from viewer's operations for browsing and its application to baseball video retrieval and summarization," in *IEEE Transaction on Multimedia*, 2007. 20

[12] Badrinarayanan, V., Galasso, F., and Cipolla, R., "Label propagation in video sequences," in *CVPR*, 2010. 133

[13] Bai, X., Wang, J., Simons, D., and Sapiro, G., "Video snapcut: robust video object cutout using localized classifiers," in *SIGGRAPH*, 2009. 132, 134, 140

[14] Basu, S., Bilenko, M., and Mooney, R. J., "A probabilistic framework for semi-supervised clustering," in *International Conference on Knowledge Discovery and Data Mining*, 2004. 59

[15] Batra, D., Kowdle, A., Parikh, D., Luo, J., and Chen, T., "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *CVPR*, 2010. 133

[16] Benfold, B. and Reid, I., "Guiding visual surveillance by tracking human attention," in *BMVC*, 2009. 15

[17] Berg, T. L. and Forsyth, D. A., "Animals on the web," in *CVPR*, 2006. 14, 55

[18] Bibby, C. and Reid, I., "Robust real-time visual tracking using pixel-wise posteriors," in *ECCV*, 2008. 132, 134

[19] Blum, A. and Chawla, S., "Learning from labeled and unlabeled data using graph mincuts," in *ICML*, 2001. 136

[20] Bobick, A. and Ivanov, Y., "Action recognition using probabilistic parsing," in *CVPR*, pp. 196–202, 1998. 78

[21] Bobick, A. F. and Davis, J., "The recognition of human movement using temporal templates," *PAMI*, vol. 23, no. 3, pp. 257–267, 2001. 12, 86

[22] Borji, A., Sihite, D. N., and Itti, L., "Probabilistic learning of task-specific visual attention," in *CVPR*, 2012. 100

[23] Breiman, L., "Random forests," in *Mach. Learn.*, 2001. 33, 34

[24] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J., "High accuracy optical flow estimation based on a theory for warping," in *ECCV*, 2004. 46, 137

[25] Brox, T., Bregler, C., and Malik, J., "Large displacement optical flow," in *CVPR*, 2009. 90, 111

[26] Budvytis, I., Badrinarayanan, V., and Cipolla, R., "Label propagation in complex video sequences using semi-supervised learning," in *BMVC*, 2010. 133

[27] Buehler, P., Everingham, M., and Zisserman, A., "Learning sign language by watching tv (using weakly aligned subtitles)," in *CVPR*, 2009. 14, 55

[28] Bulling, A. and Roggen, D., "Recognition of visual memory recall processes using eye movement analysis," in *UbiComp*, 2011. 16

[29] Bulling, A., Ward, J. A., Gellersen, H., and Troster, G., "Robust recognition of reading activity in transit using wearable electrooculography," in *Pervasive Computing*, 2008. 16

[30] Caspi, Y., Axelrod, A., Matsushita, Y., and Gamliel, A., "Dynamic stills and clip trailers," in *The Visual Computer*, 2006. 18

[31] Chawarska, K. and Shic, F., "Looking but not seeing: Atypical visual scanning and recognition of faces in 2 and 4-year-old children with autism spectrum disorder," in *Journal of Autism and Developmental Disorders*, 2009. 17

[32] Chen, Y., Bi, J., and Wang, J. Z., "Miles: multiple-instance learning via embedded instance selection," in *PAMI*, 2006. 14, 61

[33] Chockalingam, P., Pradeep, N., and Birchfield, S. T., "Adaptive fragments-based tracking of non-rigid objects using level sets," in *ICCV*, 2009. vii, 132, 134, 137, 140, 142, 143

[34] Choi, W., Shahid, K., and Savarese, S., "Learning context for collective activity recognition," in *CVPR*, 2011. 12, 15, 21, 45

[35] Choudhury, T., "Sensing and modeling human networks," in *Doctoral Thesis, MIT*, 2004. 15

[36] Collins, R., Liu, Y., and Leordeanu, M., "On-line selection of discriminative tracking features," in *PAMI*, 2005. 134

[37] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in *CVPR*, pp. 886–893, 2005. 92

[38] Dawson, G., Toth, K., Abbott, R., Osterling, J., Munson, J., Estes, A., and Liaw, J., "Early social attention impairments in autism: Social orienting, joint attention, and attention to distress," in *Developmental Psychology*, 2004. 17

[39] DeMenthon, D., Kobla, V., and Doermann, D., "Video summarization by curve simplification," in *Proceedings of ACM Multimedia*, 1998. 18

[40] Deng, J., Berg, A., Li, K., and Fei-Fei, L., "What does classifying more than 10,000 image categories tell us?," in *ECCV*, 2010. 55

[41] Devyver, M., Tsukada, A., and Kanade, T., "A wearable device for first person vision," in *RESNA 2012 Annual Conference*, 2011. 101

[42] Dhar, S., Ordonez, V., and berg, T. L., "High level describable attributes for predicting aesthetics and interestingness," in *CVPR*, 2011. 21

[43] Ding, L. and Yilmaz, A., "Learning relations among movie characters: a social network perspective," in *ECCV*, 2010. 15, 21

[44] Doherty, A. R. and Smeaton, A. F., "Automatically segmenting lifelog data into events," in *International Workshop on Image Analysis for Multimedia Interactive Services*, 2008. 20

[45] Doherty, A. R. and Smeaton, A. F., "Automatically augmenting lifelog events using pervasively generated content from millions of people," in *Sensors*, 2010. 21

[46] Dufaux, F., "Key-frame selection to represent a video," in *IEEE International Conference on Multimedia and Expo*, 2000. 18

[47] Efros, A. A., Berg, A. C., Mori, G., and Malik, J., "Recognizing action at a distance," in *ICCV*, 2003. 12, 86

[48] Einhauser, W., Spain, M., and Perona, P., "Objects predict fixations better than early saliency," in *Journal of Vision*, 2008. 16

[49] Fathi, A., Balcan, M. F., Ren, X., and Rehg, J. M., "Combining self training and active learning for video segmentation," in *BMVC*, 2011. 133

[50] Fathi, A., Farhadi, A., and Rehg, J. M., "Understanding egocentric activities," in *ICCV*, 2011. xvi, 74, 86, 87, 93, 116

[51] Fathi, A., Hodgins, J. K., and Rehg, J. M., "Social interactions: A first-person perspective," in *CVPR*, 2012. viii, 9, 23, 40

[52] Fathi, A., Li, Y., and Rehg, J. M., "Learning to recognize daily actions using gaze," in *ECCV*, 2012. 9, 24, 25, 87, 102

[53] Fathi, A. and Mori, G., "Human pose estimation using motion exemplars," in *ICCV*, 2007. 7

[54] Fathi, A. and Mori, G., "Action recognition by learning mid-level motion features," in *CVPR*, 2008. 7, 12

[55] Fathi, A. and Rehg, J. M., "Modeling actions through state changes," in *CVPR*, 2013. 88

[56] Fathi, A., Ren, X., and Rehg, J. M., "Learning to recognize objects in egocentric activities," in *CVPR*, 2011. viii, 9, 23, 57, 87

[57] Fergus, R., Bernal, H., Weiss, Y., and Torralba, A., "Semantic label sharing for learning with many categories," in *ECCV*, 2010. 14

[58] Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A., "Learning object categories from google's image search.," in *ICCV*, 2005. 14

[59] Findlay, J. and Gilchrist, I., *Active Vision:The Psychology of Looking and Seeing.* Oxford Psychology Series, Oxford University Press, 2003. 100

[60] Franchak, J. M., Kretch, K. S., Soska, K. C., Babcock, J. S., and Adolph, K. E., "Head-mounted eye-tracking of infants' natural interactions: a new method," in *Symposium on Eye-Tracking Research and Applications, ETRA*, 2010. 17

[61] Frey, B. and Dueck, D., "Clustering by passing messages between data points," in *Science*, 2007. 81

[62] Gemmell, J., Williams, L., Wood, K., Lueder, R., and Bell, G., "Passive capture and ensuing issues for a personal lifetime store," in *ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 2004. 20, 38

[63] Gibson, D. R., "Taking turns and talking ties: networks and conversational interaction," in *American Journal of Sociology*, 2005. 21

[64] Girgensohn, A. and Boreczky, J., "Time-constrained key-frame selection technique," in *IEEE International Conference on Multimedia Computing and Systems*, 1999. 18

[65] Grady, L., "Random walks for image segmentation," in *PAMI*, 2006. 134, 136

[66] Guillaumin, M., Verbeek, J., and Schmid, C., "Multimodal semi-supervised learning for image classification," in *CVPR*, pp. 902–909, IEEE, 2010. 133

[67] Guo, J. and Feng, G., "How eye gaze feedback changes parent-child joint attention in shared storybook reading? an eye-tracking intervention study," in *2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction*, 2011. 17

[68] Gupta, A., Kembhavi, A., and Davis, L. S., "Observing human-object interactions: using spatial and functional compatibility for recognition," in *PAMI*, 2009. 12, 13, 86

[69] Gupta, A., Srinivasan, P., Shi, J., and Davis, L. S., "Understanding videos, constructing plots: learning a visually grounded storyline model from annotated videos," in *CVPR*, 2009. 78

[70] Hansen, D. W. and Ji, Q., "In the eye of the beholder: a survey of models for eyes and gaze," in *PAMI*, 2010. 30, 31

[71] Harel, J., Koch, C., and Perona, P., "Graph-based visual saliency," in *NIPS*, 2006. 118

[72] Hayhoe, M., "Vision using routines: a functional account of vision," *Visual Cognition*, vol. 7, pp. 43–64, 2000. 110

[73] Hayhoe, M. and Ballard, D., "Eye movements in natural behavior," in *TRENDS in Congnitive Sciences*, 2005. 100, 110

[74] Hayhoe, M. and Ballard, D., "Eye movements in natural behavior," *Trends in Cognitive Science*, vol. 9, April 2005. 120

[75] Heidemann, G., "Focus-of-attention from local color symmetries," in *PAMI*, 2004. 19

[76] Heitz, G. and Koller, D., "Learning spatial context: using stuff to find things," in *ECCV*, 2008. 21

[77] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., and Wood, K., "Sensecam: a retrospective memory aid," in *UbiComp*, 2006. 20

[78] Hou, X. and Zhang, L., "Dynamic visual attention: searching for coding length increments," in *NIPS*, 2008. 118

[79] Ikizler-Cinbis, N. and Sclaroff, S., "Object, scene and actions: combining multiple features for human action recognition," in *ECCV*, 2010. 14

[80] Irani, M., Anandan, P., Bergenand, J., Kumar, R., and Hsu, S., "Efficient representation of video sequences and their applications," in *Signal processing: Image Communication*, 1996. 18

[81] Itti, L. and Baldi, P., "A principled approach to detecting surprising events in video," in *CVPR*, pp. 631–637, 2005. 16, 19

[82] Itti, L., Koch, C., and Niebur, E., "A model of saliency-based visual attention for rapid scene analysis," in *PAMI*, 1998. xvi, 16, 19, 100, 116, 118

[83] Joachims, T., "Transductive inference for text classification using support vector machines," in *ICML*, 1999. 63, 64, 79

[84] Jones, W., Carr, K., and Klin, A., "Absence of preferential looking to the eyes of approaching adults predicts level of social disability in 2-year-old toddlers with autism spectrum disorder," in *Archives of General Psychiatry*, 2008. 17

[85] Judd, T., Ehinger, K., Durand, F., and Torralba, A., "Learning to predict where humans look," in *ICCV*, 2009. xv, 16, 110, 111

[86] Kanade, T. and Hebert, M., "First-person vision," in *Proceedings of the IEEE*, 2012. 1

[87] Kanantani, K., "Motion segmentation by subspace seperation and model selection," in *ICCV*, 2001. 14

[88] Kang, H.-W., Matsushita, Y., Tang, X., and Chen, X.-Q., "Space-time video montage," in *CVPR*, 2006. 19

[89] Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T., "Active learning with gaussian processes for object categorization," in *ICCV*, 2007. 133

[90] Kemp, C. C., "A wearable system that learns a kinematic model and finds structure in everyday manipulation by using absolute orientation sensors and camera," in *Ph.D. Thesis, MIT*, 2005. 13

[91] Kim, C. and Hwang, J., "An integrated scheme for object-based video abstraction," in *Proceedings of ACM Multimedia*, 2001. 18

[92] Kitani, K. M., Okabe, T., Sato, Y., and Sugimoto, A., "Fast unsupervised ego-action learning for first-person sports videos," in *CVPR*, 2011. 7, 13

[93] Klin, A., Jones, W., Schultz, R., Volkmar, F., and Cohen, D., "Visual fixation patterns during viewing of naturalistic social situations as predictors of social competence in individuals with autism," in *Archives of General Psychiatry*, 2002. 17

[94] Kohli, P. and Torr, P., "Measuring uncertainty in graph cut solutions," in *ECCV*, 2006. 133

[95] Krahenbuhl, P. and Koltun, V., "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011. 112

[96] Kuettel, D., Guillaumin, M., and Ferrari, V., "Segmentation propagation in imagenet," in *ECCV*, 2012. 123

[97] Lafferty, J., McCallum, A., and Pereira, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001. 78

[98] Lan, T., Wang, Y., Yang, W., and Mori, G., "Beyond actions: discriminative models for contextual group activities," in *NIPS*, 2010. 12, 15, 21

[99] Land, M., Mennie, N., and Rusted, J., "The roles of vision and eye movements in the control of activities of daily living," *Perception*, vol. 28, no. 11, pp. 1311–1328, 1999. 110

[100] Land, M. F. and Hayhoe, M., "In what ways do eye movements contribute to everyday activities?," *Vision Research*, vol. 41, pp. 3559–3565, 2001. 5, 16, 100, 106, 112, 120

[101] Land, M., “Predictable eye-head coordination during driving,” *Nature*, vol. 359, pp. 318–320, September 1992. 110

[102] Land, M., “Motion and vision: why animals move their eyes,” *J. Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 185, October 1999. 110

[103] Laptev, I., “On space-time interest points,” in *IJCV*, 2005. xii, 12, 81

[104] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B., “Learning realistic human actions from movies,” in *CVPR*, 2008. 12, 55, 86, 96

[105] Lee, Y. J., Ghosh, J., and Grauman, K., “Discovering important people and objects for egocentric video summarization,” in *CVPR*, 2012. 20, 26

[106] Leekam, S. R., Lopez, B., and Moore, C., “Attention and joint attention in preschool children with autism,” in *Developmental Psychology*, 2000. 17

[107] Lester, J., Choudhury, T., Kern, N., Borriello, G., and Hannaford, B., “A hybrid discriminative/generative approach for modeling human activities,” in *IJCAI*, 2005. 107

[108] Li, B. and Sezan, I., “Event detection and summarization in american football broadcast video,” in *SPIE, Storage ad Retrieval for Media Databases*, 2002. 19

[109] Li, C. and Kitani, K., “Pixel-level hand detection for ego-centric videos,” in *CVPR*, 2013. 14

[110] Li, L., Wang, G., and Fei-Fei, L., “Optimol: automatic online picture collection via incremental model learning,” in *CVPR*, 2007. 14

[111] Li, L. J. and Fei-Fei, L., “What, where and who? classifying event by scene and object recognition,” in *CVPR*, 2007. 13

[112] Liu, A. and Salvucci, D., “Modeling and prediction of human driver behavior,” in *HCI*, 2001. 16

[113] Liu, D., Hua, G., and Chen, T., “Videocut: removing irrelevant frames by discovering the object of interest,” in *ECCV*, 2008. 19

[114] Lowe, D., “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004. xii, 59, 81, 96, 137

[115] Lu, Z. and Grauman, K., “Story-driven summarization for egocentric video,” in *CVPR*, 2013. 20

[116] Ma, Y.-F., Hua, X.-S., Lu, L., and Zhang, H.-J., “A generic framework of user attention model and its application in video summarization,” in *IEEE Transaction on Multimedia*, 2005. 19

[117] Mann, R., Jepson, A., and Siskind, J. M., “Computational perception of scene dynamics,” in *ECCV*, 1996. 13

[118] Marin-Jimenez, M. J., Zisserman, A., and Ferrari, V., “"here's looking at you, kid." detecting people looking at each other in videos,” in *BMVC*, 2011. 15, 21

[119] Marszalek, M., Laptev, I., and Schmid, C., “Actions in context,” in *CVPR*, 2009. 13, 55

[120] Meng, X. L. and Rubin, D. B., “Maximum likelihood estimation via the ecm algorithm: a general framework,” in *Biometrika Trust*, 1993. 75

[121] Messing, R., Pal, C., and Kautz, H., “Activity recognition using the velocity histories of tracked keypoints,” in *ICCV*, 2009. 12

[122] Mishra, A. K. and Aloimonos, Y., "Visual segmentation of simple objects for robots," in *RSS*, 2011. 16

[123] Model, D. and Eizenman, M., "A probabilistic approach for the estimation of angle kappa in infants," in *Symposium on Eye Tracking Research and Applications, ETRA*, 2012. 17

[124] Moeslund, T. B., Hilton, A., and Kruger, V., "A survey of advances in vision-based human motion capture and analysis," in *CVIU*, 2006. 12

[125] Moore, D., Essa, I., and Hayes, M., "Exploiting human actions and object context for recognition tasks," in *ICCV*, 1999. 12, 13

[126] Morariu, V. I. and Davis, L. S., "Multi-agent event recognition in structured scenarios," in *CVPR*, 2011. 15, 21

[127] Ngo, C., Zhang, H., and Pong, T., "Recent advances in content-based video analysis," in *International Journal of Image and Graphics*, 2001. 18

[128] Nguyen, M. H., Lan, Z., and la Torre, F. D., "Joint segmentation and classification of human actions in video," in *CVPR*, 2011. 93

[129] Ni, B., Yan, S., and Kassim, A., "Recognizing human group activities with localized causalities," in *CVPR*, 2009. 15

[130] Niebles, J. C., Wang, H., and Fei-Fei, L., "Unsupervised learning of human action categories using spatial-temporal words," in *BMVC*, 2006. 12

[131] Noris, B., Barker, M., Nadel, J., Hentsch, F., Ansermet, F., and Billard, A., "Measuring gaze of children with autism spectrum disorders in naturalistic interactions," in *Engineering in Medicine and Biology Society, EMBC*, 2011. 17

[132] Park, H. S., Jain, E., and Sheikh, Y., "3d social saliency from head-mounted cameras," in *NIPS*, 2012. 7

[133] Patron-Perez, A., Marszalek, M., Zisserman, A., and Reid, I. D., "High five: recognizing human interactions in tv shows," in *BMVC*, 2010. 15

[134] Pelz, J. B. and Consa, R., "Oculomotor behavior and perceptual strategies in complex tasks," in *Vision Research*, 2001. 16, 106

[135] Petrovic, N., Jojic, N., and Huang, T., "Adaptive video fast forward," in *Multimedia Tools and Applications*, 2005. 19

[136] Pfeifer, S., Lienhart, R., Fischer, S., and Effelsberg, W., "Abstracting digital movies automatically," in *Journal of Visual Communication and Image Representation*, 1996. 18

[137] Piccardi, M. and Jan, T., "Mean-shift background image modeling," in *Intl. Conf. on Image Processing (ICIP)*, pp. 3399–3402, October 2004. 17

[138] Pirsiavash, H. and Ramanan, D., "Detecting activities of daily living in first-person camera views," in *CVPR*, 2012. 13, 24, 87

[139] Platt, J., "Probabilities for sv machines," in *Advanced in Large Margin Classifiers, MIT Press*, 1999. 107

[140] Prest, A., Leistner, C., Civera, J., Schmid, C., and Ferrari, V., "Learning object class detectors from weakly annotated video," in *CVPR*, 2012. 14

[141] Price, B. L., Morse, B., and Cohen, S., "Livecut: learning-based interactive video segmentation by evaluation of multiple propagated cues," in *CVPR*, 2010. 132, 140

[142] Pritch, Y., Rav-Acha, A., Gutman, A., and Peleg, S., "Webcam synopsis: peeking around the world," in *ICCV*, 2007. 19

[143] Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T., "Hidden-state conditional random fields," in *PAMI*, 2007. 47

[144] Ramanathan, V., Yao, B., and Fei-Fei, L., "Social role discovery in human events," in *CVPR*, 2013. 15

[145] Raptis, M. and Soatto, S., "Tracklet descriptors for action modeling and video analysis," in *ECCV*, 2010. 12, 86

[146] Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J., "Collecting image annotations using amazon's mechanical turk," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010. 14

[147] Reinagel, P. and Zador, A. M., "Natural scene statistics at the centre of gaze," in *Network-Computation in Neural Systems*, 1999. 16

[148] Ren, L. and Crawford, J., "Coordinate transformations for hand-guided saccades," in *Experimental brain research*, 2009. 112

[149] Ren, X. and Gu, C., "Figure-ground segmentation improves handled object recognition in egocentric video," in *CVPR*, 2010. xi, 13, 14, 58, 64, 65, 106

[150] Ren, X. and Malik, J., "Tracking as repeated figure-ground segmentation," in *CVPR*, 2007. 134

[151] Ryoo, M. and Aggarwal, J., "Hierarchical recognition of human activities interacting with objects," in *CVPR*, 2007. 13

[152] Sand, P. and Teller, S., "Video matching," in *SIGGRAPH*, 2004. 90

[153] Schapire, R. and Singer, Y., "Improved boosting algorithms using confidence-rated predictions," in *COLT*, 1998. 76, 79

[154] Schleicher, R., Galley, N., Briest, S., , and Galley, L., "Blinks and saccades are indicators of fatigue in sleepiness warnings: looking tired?," in *Ergonomics*, 2008. 16

[155] Senju, A. and Johnson, M. H., "Atypical eye contact in autism: models, mechanisms and development," in *Neuroscience and Biobehavioral Reviews*. 17

[156] Settles, B., "Active learning literature survey," in *CS Tech Report 1648, University of Wisconsin-Madison*, 2009. 133

[157] Shamma, D. A., Shaw, R., Shafton, P. L., and Liu, Y., "What what i watch," in *ACM MIR*, 2007. 20

[158] Shotton, J., Fittzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A., "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011. 7

[159] Shotton, J., Winn, J., Rother, C., and Criminisi, A., "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *ECCV*, 2006. 112

[160] Simakov, D., Caspi, Y., Schechtman, E., and Irani, M., "Summarizing visual data using bidirectional similarity," in *CVPR*, 2008. 19

[161] Singletary, B. and Starner, T., "Learning visual models of social engagement," in *Intl. Work. on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems*, 2001. 20

[162] Sivic, J., Russell, B., Efros, A. A., Zisserman, A., and Freeman, B., "Discovering objects and their location in images," in *ICCV*, 2005. 14

[163] Skotte, J., Nojgaard, J., Jorgensen, L., Christensen, K., and Sjogaard, G., "Eye blink frequency during different computer tasks quantified by electrooculography," in *European Journal of Applied Physiology*, 2007. 16

[164] Smith, M. A. and Kanade, T., "Video skimming and characterization through the combination of image and language understanding techniques," in *ICCV*, 1997. 19

[165] Spriggs, E. H., Torre, F. D. L., and Hebert, M., "Temporal segmentation and activity classification from first-person sensing," in *Egovision Workshop*, 2009. 13, 87

[166] Starner, T., Weaver, J., and Pentland, A., "Real-time american sign language recognition using desk and wearable computer based video," in *PAMI*, pp. 1371–1375, 1998. 13, 22

[167] Stauffer, C. and Grimson, W., "Adaptive background mixture models for real-time tracking," in *CVPR*, vol. 2, pp. 246–252, 1999. 14

[168] Stuyven, E., der Goten, K. V., Vandierendonck, A., Claeys, K., and Crevits, L., "The effect of cognitive load on saccadic eye movements," in *Acta Psychologica*, 2000. 16

[169] Surie, D., Pederson, T., Lagriffoul, F., Janlert, L.-E., and Sjolie, D., "Activity recognition using an egocentric perspective of everyday objects," in *International Conference on Ubiquitous and Intelligent Computing*, 2007. 4

[170] Syeda-Mahmood, T. and Ponceleon, D., "Learning video browsing behavior and its application in the generation of video previews," in *ACM International Conference on Multimedia*, 2001. 20

[171] Taralova, E., la Torre, F. D., and Hebert, M., "Source constrained clustering," in *ICCV*, 2011. 90

[172] Tatler, B. W., Hayhoe, M. M., Land, M. F., and Ballard, D. H., "Eye guidance in natural vision: reinterpreting salience," in *Journal of Vision*, 2011. 16

[173] Torralba, A., Oliva, A., Castelhano, M., and Henderson, J., "Contextual guidance of eye movements and attention in real-world scenes: the role of global features on object search," in *Psychological Review*, 2006. 16

[174] Tsai, D., Flagg, M., and Rehg, J. M., "Motion coherent tracking with multi-label mrf optimization," in *BMVC*, 2010. vii, xvii, 133, 134, 137, 140, 142, 143

[175] Turaga, P., Chellapa, R., Subrahmanian, V. S., and Udrea, O., "Machine recognition of human activities: a survey," in *IEEE Transaction on Circuits and Systems for Video Technology*, 2008. 12

[176] Vaina, L. and Jaulent, M., "Object structure and action requirements: a compatibility model for function recognition," in *Int'l J. Intelligent Systems*, 1991. 86

[177] Varma, M. and Zisserman, A., "A statistical approach to texture classification from single images," in *IJCV*, 2005. 59, 63, 92, 108, 137

[178] Vezhnevets, A. and Buhmann, J. M., "Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning," in *CVPR*, 2010. 133

[179] Vidal, R., Tron, R., and Hartely, R., “Multiframe motion segmentation with missing data using powerfactorization and gpca,” in *IJCV*, 2008. 14

[180] Vijayanarasimhan, S. and Grauman, K., “Keywords to visual categories: multiple-instance learning for weakly supervised object categorization,” in *CVPR*, 2008. 14, 123

[181] Wang, F., Zhang, C., Shen, H. C., and Wang, J., “Semi-supervised classification using linear neighborhood propagation,” in *CVPR*, 2006. 133

[182] Wang, G., Gallagher, A., Luo, J., and Forsyth, D., “Seeing people in social context: recognizing people and social relationships,” in *ECCV*, 2010. 21

[183] Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F., “Interactive video cutout,” in *ACM Trans. Graph.*, 2005. 134

[184] Wetherby, A. M., Watt, N., Morgan, L., and Shumway, S., “Social communication proiles of children with autism spectrum disorders late in the second year of life,” in *Journal of Autism and Developmental Disorders*, 2007. 17

[185] Wolf, W., “Key frame selection by motion analysis,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996. 18

[186] Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J. M., “A scalable approach to activity recognition based on object use,” in *CVPR*, 2007. 12, 13, 66

[187] Yan, J. and Pollefeys, M., “A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate,” in *ECCV*, 2006. 14

[188] Yan, R., Yang, J., and Hauptmann, A., “Automatically labeling video data using multi-class active learning,” in *ICCV*, 2003. 133

[189] Yang, W., Wang, Y., and Mori, G., “Recognizing human actions from still images with latent poses,” in *CVPR*, 2010. 13

[190] Yao, B. and Fei-Fei, L., “Modeling mutual context of object and human pose in human-object interaction activities,” in *CVPR*, 2010. 13, 86

[191] Yarbus, A., *Eye Movements and Vision.* Plenum Press, 1967. 16, 100

[192] Ye, Z., Li, Y., Fathi, A., Han, Y., Rozga, A., Abowd, G. D., and Rehg, J. M., “Detecting eye contact using wearable eye-tracking glasses,” in *2nd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction (PETMEI) in conjunction with UbiComp.* 30

[193] Yi, W. and Ballard, D., “Recognizing behavior in hand-eye coordination patterns,” in *International Journal of Humanoid Robots*, 2009. 16, 100, 124

[194] Yu, B., Ma, W.-Y., Nahrstedt, K., and Zhang, H.-J., “Video summarization based on user log enhanced link analysis,” in *ACM International Conference on Multimedia*, 2003. 20

[195] Yu, T., Lim, S.-N., Patwardhan, K., and Krahnstoever, N., “Monitoring, recognizing and discovering social networks,” in *CVPR*, 2009. 15, 21

[196] Yuen, J., Russell, B., Liu, C., and Torralba, A., “Labelme video: building a video database with human annotations,” in *ICCV*, 2009. 132, 134

[197] Zacks, J. M., Speer, N. K., vettel, J. M., and Jacoby, L. L., “Event understanding and memory in healthy aging and dementia of the alzheimer type,” in *Psychol. Aging*, 2006. 21

[198] Zeisl, B., Leistner, C., Saffari, A., and Bischof, H., "On-line semi-supervised multiple-instance boosting," in *CVPR*, 2010. 133

[199] Zen, G. and Ricci, E., "Earth mover's prototypes: a convex learning approach for discovering activity patterns in dynamic scenes," in *CVPR*, 2011. 21

[200] Zhang, H. J., "An integrated system for content-based video retrieval and browsing," in *Pattern Recognition*, 1997. 18

[201] Zhao, L., Qi, W., Li, S., Yang, S., and Zhang, H., "Key-frame extraction and shot retrieval using nearest feature line(nfl)," in *Proceedings of ACM Multimedia Workshop*, 2000. 18

[202] Zhu, X., "Semi-supervised learning literature survey," *University of Wisconsin-Madison*, 2008. 133, 138

[203] Zhu, X., Ghahramani, Z., and Lafferty, J., "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003. 133, 141, 142, 143

[204] Zhu, X., Ghahramani, Z., and Lafferty, J., "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003. 136

[205] Zwaigenbaum, L., Bryson, S. E., Rogers, T., Roberts, W., Brian, J., and Szatmari, P., "Behavioral manifestations of autism in the first year of life," in *International Journal of Developmental Neuroscience*, 2005. 17