Master Thesis

Finding Objects of Common Interests from Multiple First-Person Videos

(複数の一人称視点動画からの注目対象の抽出)

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

48-146461 Nattawan TANTIRUJANANONT
Advisor Prof. Yoichi SATO

February 2016

Abstract

Recent development of head-mounted cameras allows us to easily record first-person videos. First-person videos convey attentions, intentions, and interactions of camera wearers. Analyzing multiple first-person videos recorded by a group of people helps us to understand social group activities such as group discussions.

In this thesis, we introduce a novel technique for discovering objects of common interests from multiple first-person videos recorded in a social event. Objects of common interests are defined as the objects being looked at by a group of people jointly for a period of time. While such objects are salient cues to understand group activities, it is still a challenge to detect them from multiple first-person videos. The difficulties arise from the necessity to locate unknown objects with unrestricted appearances, categories and sizes in an unsupervised manner. The problem is even more challenging when many objects are presented in a scene.

Our method makes use of a glass-type eyes tracking device to incorporate points of gaze into an object co-segmentation framework. Video co-segmentation techniques provide an important cue for objects of common interest since they can unsupervisedly segments common objects from background by using mutual information of the common objects (e.g., color/shape similarities) across videos. However, video object co-segmentation cannot pinpoint an object of common interests in a cluttered scene, where there are many common objects exist together. By utilizing eye tracking data, we can limit the location of the objects of common interests. Thus, our proposed method can identify precisely one object of common interests per video frame.

First, our method segments video frames into over-segmented supervoxels. Then, candidates of objects are generated by combining supervoxel segments according to gaze events, i.e., fixation events. A hierarchy of fixations is used for identifying a set of segments that corresponds to each object candidate. Finally, the candidates are clustered by a graph-based object co-segmentation framework, and the objects of common interests are detected. The proposed method can work under several challenging situations, such as absence of common objects and existence of multiple of objects in a cluttered scene.

Acknowledgements

I would like to extend my sincere thanks to all those who provided me the possibility to complete this thesis. It would not have been possible without the kind support and help of many individuals who provided expertise that greatly assisted the research.

First, I would like to express my deepest appreciation to my research advisor, Prof. Yoichi Sato, for the continuous support of my Master study. His kind insight and expertise greatly assisted the research. I could not have imagined having a better advisor and mentor for my Master study.

I am highly indebted to Dr. Ryo Yonetani for his patience and constant supervision. His directions and insightful ideas greatly help me in completion of this research. He has also taught me a lot about novel computer vision techniques and assisted me in many experiments.

I would like to express my special thanks of gratitude to my colleague, Ms. Rie Kamikubo, who spend many hours helping me to record a new dataset for evaluating my proposed method. I am also grateful to Mr. Wiennat Mongkulmann for his encouragements and useful advices on my research, my entrance examination to The University of Tokyo, and my life in general.

I would like to give credits to CREST, JST for financial support of the project. I am also grateful to all of the department faculty members who gave access to the laboratory and research facilities, and help me in various processes, especially to Sato Laboratory secretaries, Ms. Sakie Suzuki, Ms. Yoko Imagawa, Ms. Chio Usui. Without their precious support, it would not be possible to conduct this research.

Finally, I would like to express my love and gratitude to my family: my parents and to my sister for the unceasing support and attention throughout the time I study in Japan.

December 7$^{\text{th}}$, 2015
Nattawan Tantirujananont

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Due to the emergence of head-mounted cameras, it has become possible to unobtrusively record videos of human activities from first-person perspective. First-person videos not only show camera wearers' actions, but also convey their attentions and interactions with other people or objects. More importantly, wearable cameras can be used to record first-person videos of each participant who joins a social event. Those videos provide important cues to recognize social interactions between participants. Recently, many researches in a field of egocentric vision focus on analyzing such recordings in order to achieve a better understanding of group behaviors, for example, social relationships [4], social interactions [14], and social attention [**?**][27].

In this work, we deal with a novel problem of detecting and localizing group attentions. We record first-person videos and eye tracking data of many people who participate in a group activity by using a glass-type eye tracker (Figure 1.1). Our objective is to detect and localize objects that are looked at jointly by many people for some period of time, i.e., the objects of common interests (Figure 1.2). We believe that discovering such objects is beneficial for social behavioral studies. It can also assist in detecting important objects in an application like multiple videos summarization [5] [20].

1.2 Challenges and Contributions

There are several challenges in detecting the objects of common interests. The main difficulty is the need to locate unknown objects with unrestricted appearances, types, sizes, and viewpoints in an unsupervised manner. Also, the objects can move-in/-out of the field of views of the participants due to unconstrained head movements.

Figure 1.1: Pupil eye tracker [1].

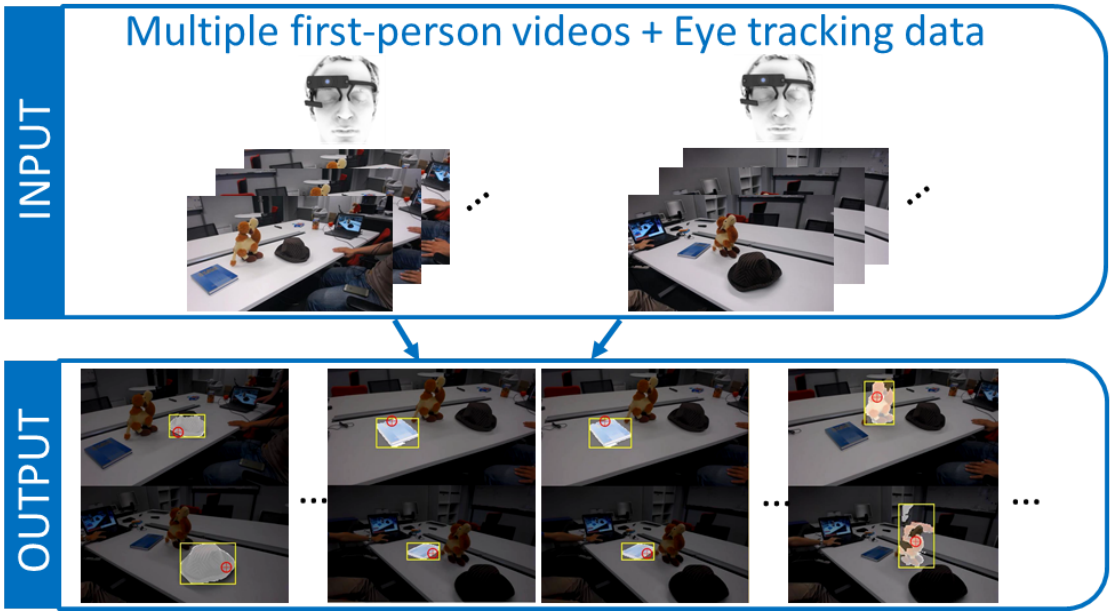

Figure 1.2: The goal of our work. The objective is to detect objects of common interests from multiple first-person videos and their accompanying eye tracking data.

To overcome the aforementioned difficulties, we propose a method to discover the objects of common interests based on an object co-segmentation framework. Recent video object co-segmentation techniques segment common objects from multiple videos in unsupervised [11][40] or weakly-supervised [36] manner. They use mutual information of the target object (e.g., color/shape similarity) that appears in many videos to make up for the lack of supervision. However, it is still difficult to pinpoint the location of the objects of common interests, especially when many objects exist in the scene.

Our key insight is that the points of gaze recorded together with first-person videos can be utilized to limit the locations of object candidates. With gaze information, we can discover exactly one object of interest per video frame, even if there are multiple objects present in a cluttered scene. We also extend the state-of-the-art clustering approach [40] to identify if objects of common interests are present or absent from the participants' field of views.

Another contribution is a new first-person videos dataset that we record. The dataset is used for evaluating our proposed method.

In summary, our contributions are as follows:

1. We introduce a novel problem of detecting objects of common interests, given first-person videos and eye tracking data of multiple participants.

2. We proposed an efficient method to solve the problem. The proposed method utilizes eye tracking data to assist finding objects which are similar appearances across multiple videos and are looked at by multiple people.

3. We record a new dataset for evaluating our method. The dataset contains 18 video sets recorded in three different scenes. Each set consists of 2 first-person videos showing joint attentions on five objects (∼1 min. each).

1.3 Thesis Outline

This thesis is organized as follows. The first part of Chapter 2 includes recent researches on gaze event detection, specifically saccade and fixation detection algorithms. The second part of Chapter 2 introduces many works on video object co-segmentation methods, including the state-of-the-art approach [40]. The third part of Chapter 2 is a brief introduction to supervoxel segmentation methods. Then, Chapter 3 describes our proposed method. And, Chapter 4 presents the performance evaluation on our dataset and a detailed analysis of our proposed method. Finally, Chapter 5 outlines ideas for future research direction and concludes the thesis.

Chapter 2

Related Works

In this chapter, we introduce many topics that are related to our proposed method. The objective is to detect objects of common interests from first-person videos. Our problem is slightly different from the current group attention localization problem [**?**][27]. First, we also use points of gaze as a cue for attentions. Second, we want to detect the objects of common interests, and not just predicting the points of joint attentions. This provides a new challenge, that is, the need to locate the boundary of the objects of common interests. To solve this challenge, we incorporate three related works introduced in this chapter, that is, *gaze event detection*, *supervoxel segmentation*, and *video object co-segmentation*.

In the first step, the proposed method generates the proposals of the objects of common interests. A boundary of each object candidate is determined by a combination of supervoxels guided by gaze events. We present a brief description on gaze event detection and supervoxel segmentation in the first and the second parts of this chapter, respectively. In the second step, the proposed method groups the proposals to find the objects of common interests by using a video object co-segmentation framework. The third part of this chapter provides a detail on the video object co-segmentation approaches.

2.1 Gaze event detection

Eye tracking data can be noisy or loss because of blinking, rapid head motion, or tracking error. Moreover, some points of gaze might not convey camera wearer's intention, for example, a point of gaze that was collected when camera wearer unintentionally glanced at some location in a quick motion. In both cases, we need to process the raw gaze data in order to remove the noisy data, and identify the gaze portion that express camera wearer's attention. The process is called gaze event detection.

The objective of gaze event detection is to segment the eye tracking data into periods of fixation, saccade, smooth pursuit, and blink. In our work, we detect two types of gaze event: saccade and fixation. Saccade is a rapid movement of eyes in a short period of time. Fixation occurs when camera wearer fixates at something for a while (Figure 2.1). Since fixations normally express camera wearer's interest, we want to detect all fixation periods within our videos.

In the egocentric scheme, head motions obstruct the gaze event detection process, specifically, a vestibulo-ocular reflex (Figure 2.2). The vestibulo-ocular reflex

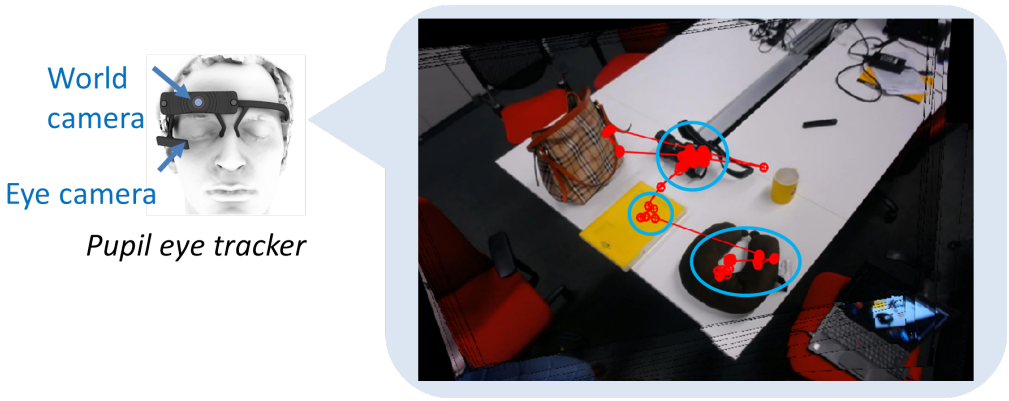

Figure 2.1: An illustration of gaze event detection. Several video frames are concatenated into a panoramic image. Red marks show a sequence of gaze points, Blue circles denotes the clusters of gaze that represent fixation events.

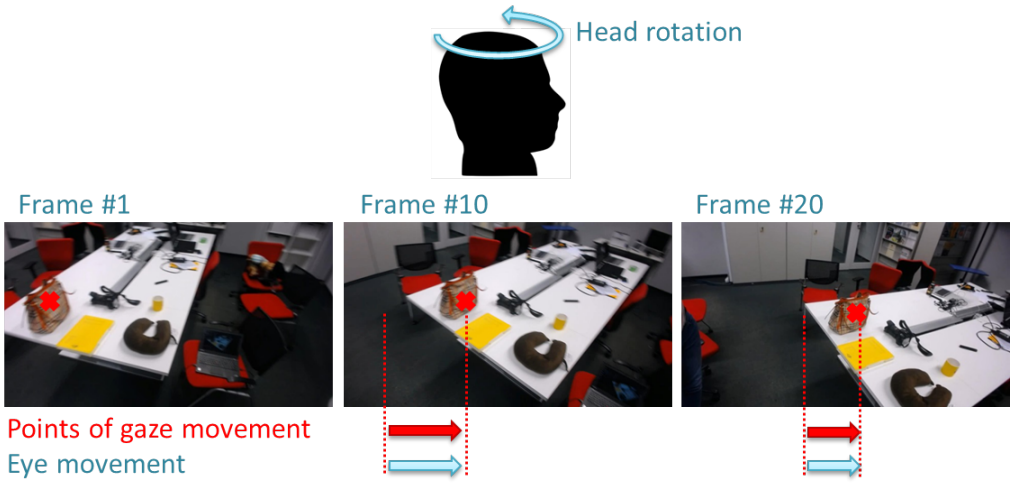

Figure 2.2: Eye movement compensates for head rotation.

is a reflex eye movement that is produced when a human fixates at one location and then rotates his/her head. This reflex will produce an eye movement in an opposite direction in order to compensate for the head motion, thus preserving the original location on the visual field. As a result, points of gaze in this period will be incorrectly classified as saccade instead of fixation because of the eyes' motion.

To solve this problem, recent literature [24] proposed a method to compensate for the head motion in order to obtain the correct event detection results. To negate the head motion, the relative motions of two consecutive frames are first estimated to transform the points of gaze into the same coordinate. Specifically, a video frame is used as a template to match with the next frame. Then, the relative motion can be computed from the maximum of the cross correlation using phase correlation. Finally, a normal gaze event detection method is used on the transformed points of gaze.

Normally, fixation detection algorithms classify the events based on dispersion, velocity, acceleration, or the combination of them; see [33]. Dispersion-based algorithms identify a portion of eye tracking data as a fixation period if the points of gaze are located within a spatially limited region for more than a predefined duration.

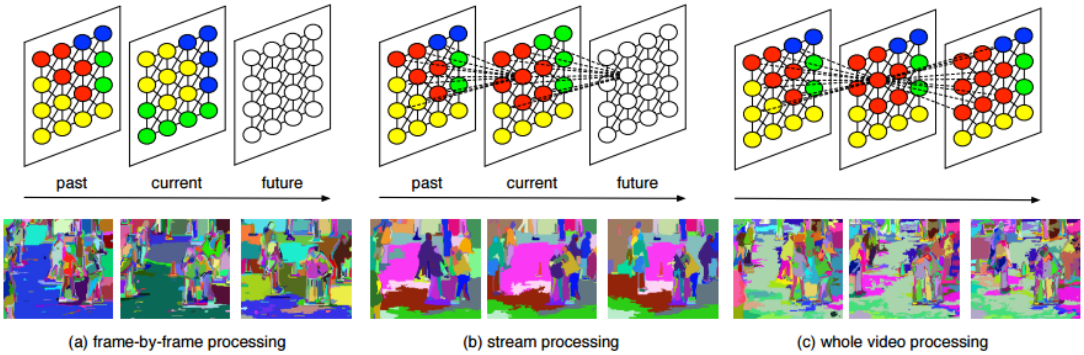

Figure 2.3: A comparison between three different processing paradigms for video segmentation [38]. (a) Frame-by-frame processing which process each frame individually. The result is bad due to low temporal coherency. (b) Stream processing [38]. And, (c) 3D volume processing which process a whole video. The result is the best, but not suitable for long video due to high complexity.

Such method has been used in a commercial eye tracker like Tobii Technology.

Velocity-based algorithms simply compute the velocity of gaze movements, and threshold them into saccade and fixation periods. In other words, if gaze velocity exceeds a threshold, the gaze data will be identified as a saccade event. Fixation period is defined as a period between saccade. The velocity threshold can be predefined by user, or automatically computed based on the input gaze data. Recent research [26] iteratively estimates the velocity threshold based on mean and standard deviation of the velocity that are under the threshold.

Similarly, acceleration-based algorithms specify an acceleration threshold to categorize eye tracking data into saccades and fixations. Acceleration criteria ordinarily acts as a complimentary of the velocity-based methods.

2.2 Supervoxel segmentation

Supervoxel segmentation is a method to segment video into several small volumes. Each supervoxel (volume) should have consistent characteristics, both spatially and temporally. Recent works in this domain include [17][38] and [39].

The work of [17] segments a video into several layers of supervoxels. The lower levels are more finely segmented volume, while the higher levels are more coarsely segmented regions. Specifically, they oversegments a video into a small space-time regions grouped by appearance. Then, they construct an initial 3-D graph, where each node is a space-time region and edge represents similarity between regions. The segmentation method is apply on the graph again to obtain a new graph, which consists of super-regions, i.e., regions composed from smaller regions. The super-regions in-turn forms a graph that can be segmented again. By iteratively repeat this process, they represent a hierarchy of regions in tree structure.

The state-of-the-art method of [38] implements a graph-based hierarchical segmentation method [17] into a streaming framework. The concept is motivated by data stream algorithms (Figure 2.3), where each video frame is processed only once and does not change the segmentation of previous frames. An example result is shown in Figure 2.4.

Figure 2.4: Example output from a streaming hierarchical supervoxel segmentation method [38]. (a) the video with frame number on top-left, (b) the 5th layer, (c) the 10th layer, (d) the 14th layer segmentations.

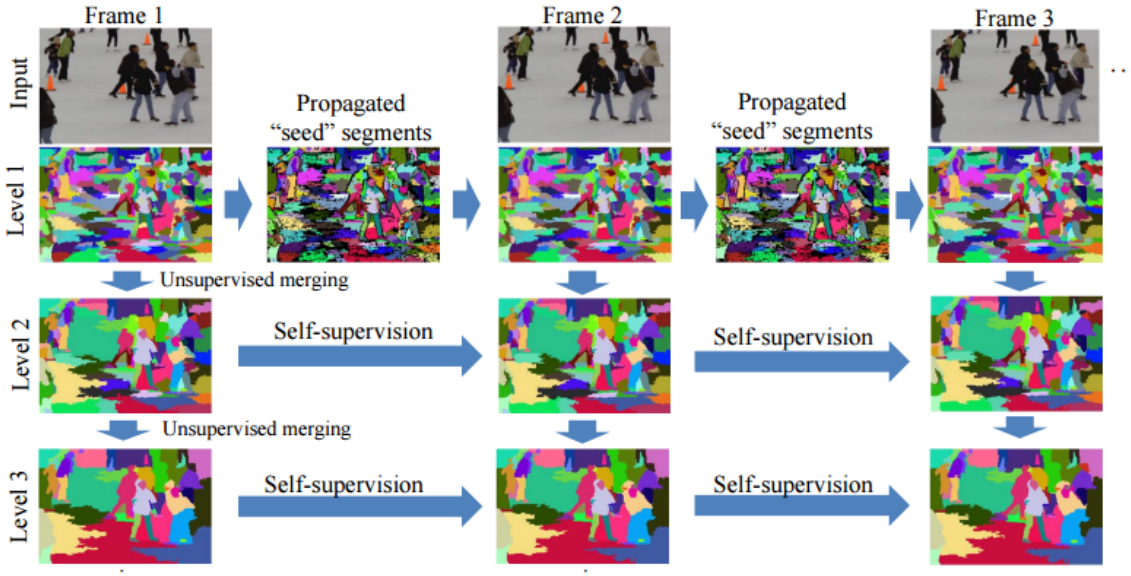

Figure 2.5: An illustration of the processing flow of [39].

The most recent method proposed by [39] introduces an online hierarchical supervoxel segmentation method for time-critical applications (Figure 2.5). It segments a video stream up to the latest frame as soon as it arrives without the need of streaming buffer. The approach is similar to [38], except that the segment labels are propagated from one frame to the next frame based on both motion (dense optical flow) and appearance cues. Then, a new graph is built for the next frame, and new segments (if any) are generated using the graph-based merging scheme. Finally, higher-level segmentation are generated using a self-supervision merging scheme based on the higher-level segmentation of the previous frame.

2.3 Video object co-segmentation

Object co-segmentation task was first introduced by [28], which worked on a collection of images. The idea is to use mutual information of the target object (e.g. color/shape similarity) that appears in many images to make up for the absence of

supervision. Later, the object co-segmentation task is extended to handle multiple object categories [22], as well as the absence of common objects, especially in internet images collections [10][30][37].

Recently, a few researchers extend the idea of object co-segmentation into video domain. Unfortunately, many challenges still have not be (or partially be) addressed. First problem is the existence of common object. Some of the works assume that all video frames must contain common object [16]. Several works [23][36] treat the frames that lack common object as outlier, and try to learn a classifier to identify whether a frame contains an object or not.

By contrast, our work assumes that there is not always an object of common interest since a group of people do not necessarily look at the same object for a whole time. Also, we cannot use a classifier to identify whether a frame contains a common object or not, because even if the frame contains a common object, that object might not be an object of common interests. For example, a person looks at a book on the table and another person looks at a laptop on the same table. In this case, first-person videos from these two people will display both the book and the laptop. Thus, both objects will be identified as the common objects. However, they are not the objects of common interests.

Another challenge in video co-segmentation is the variability of object appearances across viewpoints. Many works assume that the object appearance and shape must be consistence both within a video and across videos. In our case, object appearance across videos might be varied greatly due to the change of viewpoint.

The next challenge is the number of categories of objects. Many prior researches [18][23] co-segment videos into foreground, i.e., a common object, and background. In other words, they assume the videos consist of only one object category. However, we aim to find all objects of common interests, which can be more than one.

One recent work in video object co-segmentation is proposed by [31]. They proposed a method to extract objects of a single class that move in a similar manner across multiple videos. Given input videos, pixels are grouped at two levels: the higher level groups pixels into space-time tubes, and the lower level groups pixels into regions within each frame. An initial estimation of the foreground and background labeling is calculated from an objectness and saliency measure, and it is used to construct a probabilistic distribution of the feature vectors of tubes and regions. Then, a probabilistic framework is modeled to obtain the co-segmentation results.

The method of [11] co-segments multiple objects from videos by formulating a non-parametric Bayesian model, which is based on a video segmentation prior. It can be used with multiple classes of objects and can deals with the absence of common object.

One interesting work of [40] proposed am improved method of [11]. The method consists of two main steps: object tracklets generation, and tracklets grouping. In the first step, video frames are segmented into superpixels. The superpixels are selected based on their objectness score, shape similarity and color similarity to form object proposal tracklets that are spatially salient and temporally consistent. In second step, a graph is constructed, where the nodes are object proposal tracklets, and the weights are calculated by tracklets similarity measure (Figure 2.6). Then, graph-based clustering is used to group objects with similar shape and appearance together. Each group corresponds to each common object. Finally, the detected tracklets in each group is used to initialize per-pixel segmentation to get the final

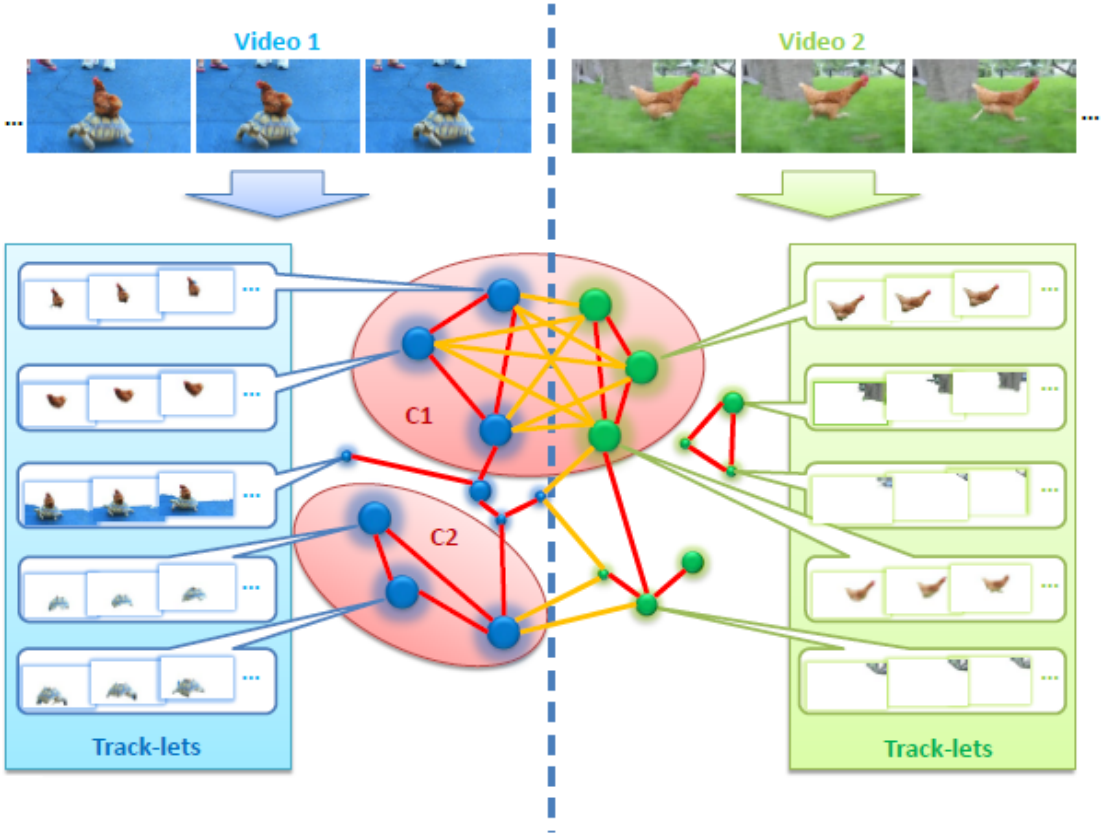

Figure 2.6: Graph-based tracklets grouping step [40].

co-segmentation results (Figure 2.7).

Another approach from [36] presents a spatio-temporal energy minimization formulation to simultaneously discover and co-segment common video objects across multiple videos containing irrelevant frames. This method is weakly-supervised method that requires 1 to 3 frame-level labels in order to identify the irrelevant frames.

The task that is closely related to object co-segmentation is object co-localization. Instead of obtaining per-pixel segmentation of common objects, object co-localization aims to locate the bounding boxes that cover the common objects. Several works in this area are [18][19] and [23].

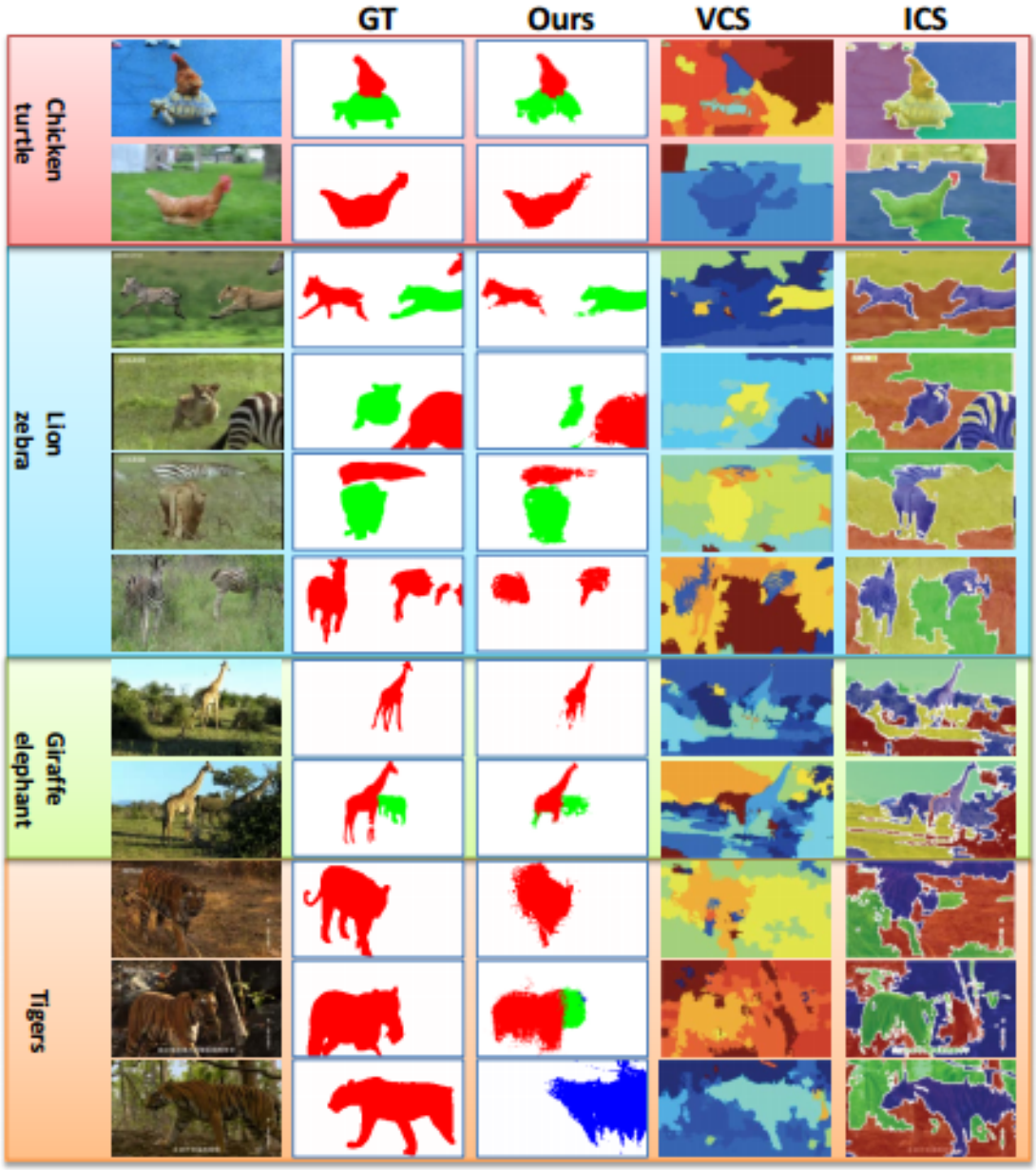

Figure 2.7: An example of video co-segmentation results [40]. Column 1 is the original video frames; Column 2 ('GT') is the ground truth for co-segmentation; Column 3 ('Ours') is the results of [40]; Column 4 ('VCS') is the results of [11]; Column 5 ('ICS') is the results of [22].

Chapter 3

Proposed Method

3.1 Overview

In this chapter, we present a detailed description of our proposed method. The input of our method is multiple first-person videos recorded by the participants, and their accompanying eye tracking data. The videos are a sequence of frames, each frame has one (or zero if missing) 2d point of gaze. Currently, we record first-person videos from two participants. The output is a frame-by-frame segmentation of the objects of common interests.

The proposed method consists of three main steps. In the first step, we segment video frames into over-segmented supervoxels, and segment eye tracking data into a hierarchy of fixations. Then, we use a fixation event to determine a set of supervoxel segments. They are further combined to generate a candidate for objects of common interest. This process is repeated for all fixations in a hierarchy, and we obtain many object candidates. In the second step, the object candidates are clustered, and the objects of common interests are detected. The clustering is done by an adaptation of the video object co-segmentation approach [40]. Finally, the objects of common interests' regions are re-segmented using [29] in the post-processing step.

3.2 Generating object candidates based on gaze events

The first step of the proposed method is to generate candidates for objects of common interest across multiple videos. The simplest approach is to use object detectors [3][13][25], which generate object proposals based on geometric and visual saliency cues. Unfortunately, current object detectors sometimes fail to detect small or inconspicuous objects even if they are fixated by the camera wearers. They also provide many proposals that are unrelated to the objects of common interests, such as objects in the background.

To solve the problems, we propose an approach to construct object proposals according to gaze events. We segment fixation events from the eye tracking data, and construct the candidates by combining a set of supervoxels based on the detected events. We also measure the quality of the candidates and screen out the low-quality ones. With this gaze-guided approach, we can obtain the proposals of all (and only) objects that are looked at by the camera wearer, regardless of their sizes, colors, or

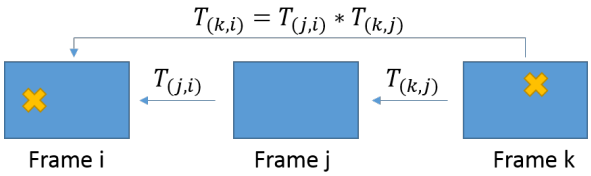

Figure 3.1: Relative motion calculated from consecutive video frames when gaze data is missing.

conspicuousness.

3.2.1 Detecting fixation events in a hierarchy

Eye tracking data normally contains data loss and spatial error due to blinking, rapid head motion, and tracking error. Moreover, there are points of gaze that do not reflect the attention of the camera wearer, such as rapid, unintentional eye movements. The missing points of gaze can be predicted by interpolation such as linear interpolation and median filtering. Then, the periods of gaze that convey camera wearer's attentions can be identified by fixation events detection method.

In first-person videos, detecting fixation events is not an easy task due to a large amount of head motions. To overcome the difficulty, recent work [24] proposed a method to compensate for the head motions. Following [24], we compensate for head motions in fixation detection. More specifically, relative motions of two consecutive frames are first estimated to transform the points of gaze into the same coordinate to negate the head motion. Each frame is used as a reference coordinate for the next frame. Then, the velocity of eye motions is calculated from the transformed points of gaze. A sequence of fixation periods is finally obtained by segmenting points-of-gaze stream temporally based on the velocity.

To compute relative motion between two consecutive frames F_i and F_{i+1}, we extract local features such as SURF [6] and match the features between F_i and F_{i+1}. Then we estimate the geometric transformation T_i that maps F_{i+1} to F_i. The point of gaze in F_{i+1} is then transformed to F_i using T_i. If the point of gaze in F_{i+1} is missing, we accumulate the transformation T_i to be used the next frame (Figure 3.1).

We compute angular velocity from the transformed points of gaze (x_i, y_i) and (x_{i+1}, y_{i+1}):

$$v_i = \frac{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{t_{i+1} - t_i}, \tag{3.1}$$

where t_i and t_{i+1} are the timestamp of frame i and frame $i + 1$ correspondingly. Since videos have a constant frame rate, $t_{i+1} - t_i$ will also be a constant and can be omitted from the equation:

$$\theta_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{3.2}$$

The angular velocity is then used for detecting fixation periods. Eye tracking data is segmented into a hierarchy of fixations. The segmentation method is an adaptation of saliency primitives segmentation based on the scale-space analysis [32]. A sequence of angular velocity, $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_N\}$, is used as input. The sequence is convolved with a series of Gaussian functions with varying smoothing scales $\{\xi_1, ..., \xi_L\}$:

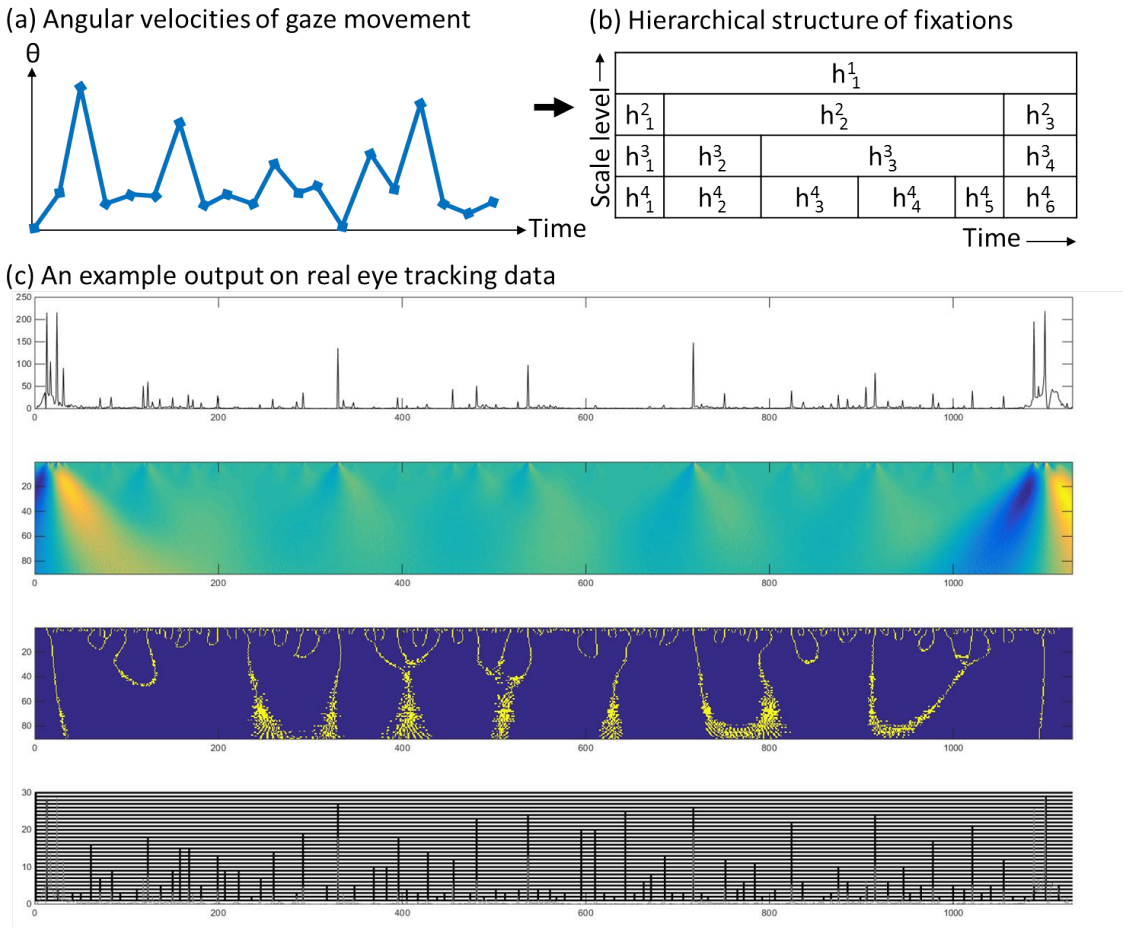

Figure 3.2: Hierarchical fixation segmentation based on scale-space analysis [32]; (a) angular velocities computed from a sequence of transformed points of gaze, (b) hierarchical structure of fixations, and (c) an example output on real eye-tracking data. Eye tracking data (1st graph) is convolved with Gaussian functions in various smoothing scales (2nd graph). Then, Local maximum points are tracked through smoothing scale levels (3rd graph). Finally, we obtain a hierarchy of fixations (4th graph).

$$\boldsymbol{\theta}^{(\xi_l)} = \boldsymbol{\theta} * Gauss^{(\xi_l)}, \tag{3.3}$$

where ξ_l is a smoothing scale of level l, $Gauss^{(\xi_l)}$ denotes a Gaussian function with smoothing scale ξ_l, and $*$ is a a convolution operation.

The local maximum points in a set of outputs $\{\boldsymbol{\theta}^{(\xi_1)}, ..., \boldsymbol{\theta}^{(\xi_L)}\}$ are tracked through the varying smoothing scales from ξ_L to ξ_1. Specifically, each local maximum point at a curtain smoothing scale ξ_l is matched with its corresponding point at scale ξ_1 by tracking the point from ξ_l to ξ_1. Then, the point is used as a segmentation point. By doing this for all local maximum points at every scale level, we can obtain a hierarchical structure of the points because new points can appear when smoothing scales decrease (Figure 3.2).

For each smoothing level, eye tracking data is segmented based on segmentation points of that level. A set of intervals generated at smoothing level l (i.e., a set of intervals generated with scale ξ_l) is denoted as a set of fixation events of the level l in a hierarchy of fixations:

$$H^l = \{h_1^l, h_2^l, ..., h_{N^l}^l\}, \tag{3.4}$$

where h_j^i denotes fixation event j at smoothing level i.

Finally, we obtain a hierarchy of fixations, $\mathbf{H} = \{H^1, H^2, ..., H^L\}$, where L is the number of levels in a hierarchy:

3.2.2 Generating object candidates

The objective of this step is to generate the candidates of objects being looked at by wearers. The naive way is to segment each video frame into several regions, then use eye tracking data to identify the region that is looked at by the wearers. The important problem is that video frame segmentation is not perfect. Thus, the selected region might contain only a part of the object or contains both object and background.

To solve the aforementioned difficulty, we use a hierarchy of fixation periods to generate the candidates of objects being looked at by wearers. The key idea is that people tends to look at every parts of object if they are interested in that object. Even if the camera wearer looks at only a segment of object in each video frame, we can discover a whole object region by combining many segments from several video frames together.

The next question is which segments from which frames should be combined together. As mentioned, we assume that people will fixate at the object that interested them, and will look at every segments of the object. In other words, the people will look at a whole object during a fixation event. According to this assumption, we select a set of supervoxel segments that are looked at by the camera wearer within each fixation period. Then, the selected segments are combined together to form one object candidate. We repeat this step for all fixation events in a hierarchy. Thus, we will obtain a set of object candidates.

Given an input video V, we segment each video frames into several over-segmented supervoxels using [38]. Each supervoxel segment should represent a part of the object, a whole object or background, but not a combination of object and background region.

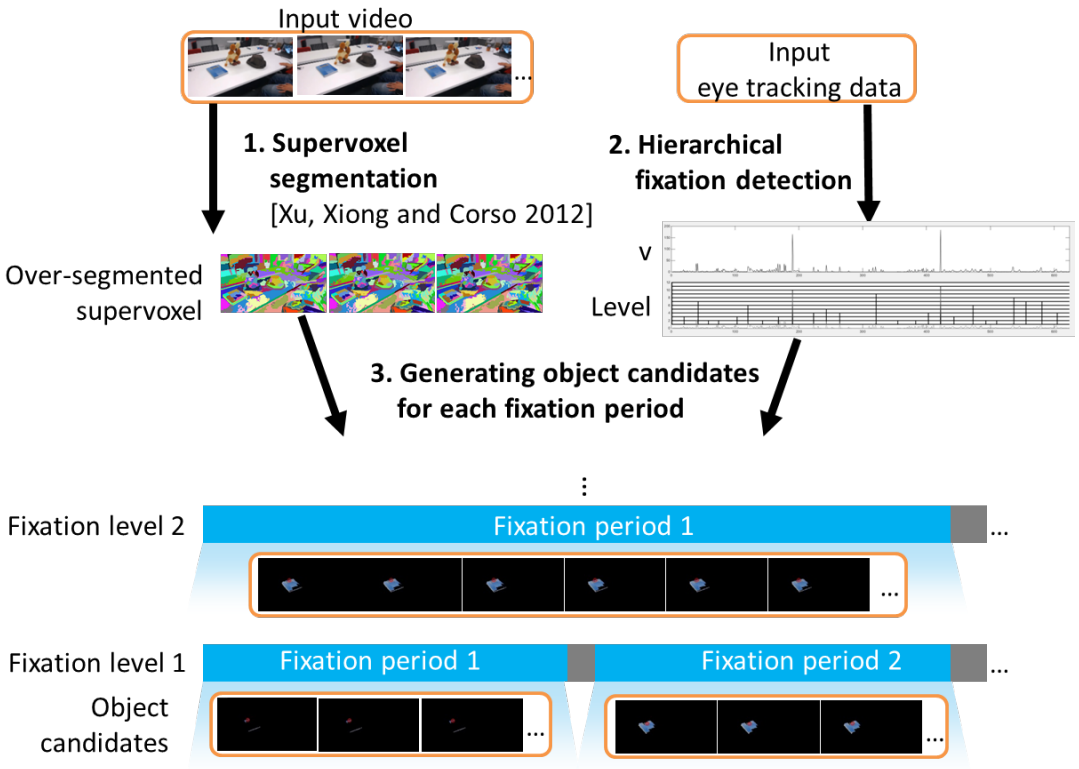

Figure 3.3: We generate object candidates from a hierarchy of fixations.

Given an accompanying eye tracking data, we discover a hierarchy of fixations, $\mathbf{H} = \{H^1, H^2, ..., H^L\}$, where L is the number of levels in a hierarchy. We define H^l as a set of fixation events at level l:

$$H^l = \{\mathcal{X}_1^l, \mathcal{X}_2^l, ..., \mathcal{X}_{N^l}^l\}, \tag{3.5}$$

where N^l is the number of fixation events in H^l.

We construct one object candidate for each fixation event. Suppose videos are decomposed into a set of supervoxels $\{s_1, \ldots, s_K\}$, where s_k defines a certain (connected) spatio-temporal region. Eye tracking data is described by a set of spatio-temporal point $\mathcal{X} = \{(x_1, y_1, t_1), \ldots, (x_N, y_N, t_N)\}$. The j-th fixation at scale l, $\mathcal{X}_j^l$, is defined by a subset of $\mathcal{X}$, say, $\mathcal{X}_j^l \subset \mathcal{X}$. Then, the object candidate of the j-th fixation at scale l, namely O_j^l, can be defined as:

$$O_j^l = \{s_k | x \subseteq s_k \quad \text{and} \quad x \in \mathcal{X}_j^l\}, \tag{3.6}$$

where $x \subseteq s_k$ means the point of gaze x is located inside the supervoxel s_k.

We discover all object candidates from all fixation events $\{\mathcal{X}_1^l, \mathcal{X}_2^l, ..., \mathcal{X}_{N^l}^l\}$. Then, we repeat this step to all levels in fixation hierarchy $\{H^1, H^2, ..., H^L\}$.

In summary, an object candidate is constructed by grouping over-segmented segments together to form a complete object. To determine which segments should be in the same group, we segment eye-tracking data into a hierarchy of fixations. The segments from the same fixation will be in the same group. Thus, each fixation will form one object candidate constructed by merging together several over-segmented supervoxel segments that are looked at by the participant (Figure 3.3).

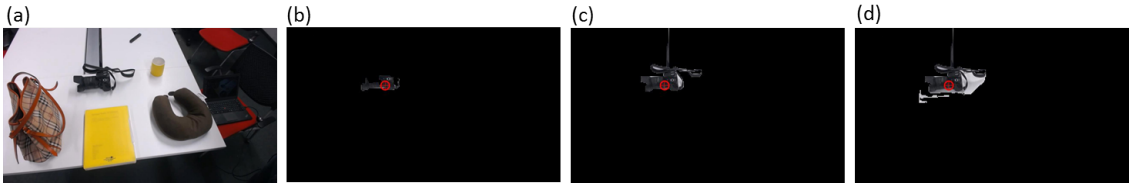

Figure 3.4: Examples of good and bad object candidates: (a) Original video frame, (b) Over-segmented candidate, (c) Good candidate, and (d) Under-segmented candidate.

3.2.3 Scoring object candidates

In this section, we introduce a method to compute a score that measures the quality of an object candidate. Good object candidates should represent a whole object segment. Bad object candidates can consist of only a part of the object (over-segmented), both object and background region (under-segmented), or a whole background regions. Figure 3.4 illustrates good and bad object candidates.

To determine a score of an object candidate, we train a regressor on four indicators: location, size, saliency and compactness.

Location indicator We believe that an object of interest is likely to be at a specific location within a video frame. For example, an object of interest tends to exist near the center of the frame than other locations. The reason is because people usually rotate their heads to focus on the object and make the object locates at the center of their field of view.

In our work, location feature LF is defined as the center of the tight bounding box that covers the object candidate:

$$LF = \{x, y\}, \tag{3.7}$$

where x and y is the x- and y-position of the center of the tight bounding box that covers the object candidate.

Size indicator Size feature is defined as a size of the tight bounding box that covers the object candidate. We believe that object sizes tend to be within a specific range. If a size of an object candidate is too small, the object candidate is likely to represent only a part of the object (over-segmented candidate). If a size of an object candidate is too large, the object candidate is likely to contain a background region (under-segmented candidate or background candidate).

The size feature SF is define as:

$$SF = \{w, h\}, \tag{3.8}$$

where w and h is the width and height of the tight bounding box that covers the object candidate correspondingly.

Saliency indicator We compute object saliency score from [21]. The saliency score measures how much a region distinct from its surrounding regions. We believe that the object region should be distinct from background, resulting in a high

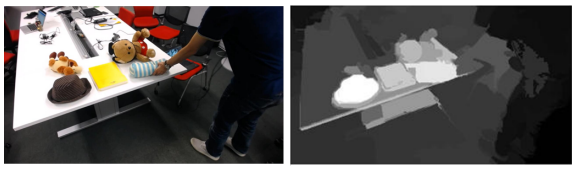

Figure 3.5: (Left) Original video frame, (Right) Saliency map computed by [21].

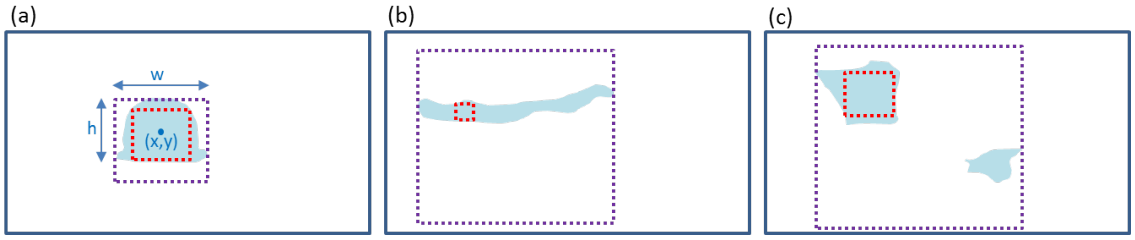

Figure 3.6: Examples of object candidate shapes: (a) A normal object shape (hat), (b) A long curved region that is unlikely to be an object, (c) A weird scattered blobs that is very unlikely to be an object. Red square represents the largest possible square that a region can contain. The size of the red square is the *scale of region*. Violet square represents the smallest possible square that cover a region R. The size of the violet square is denoted as $outer(R)$.

saliency score when the object candidate contains only a small background region. Therefore, we use the saliency scores to discourage the selection of object candidates that contain large background regions. Figure 3.5 shows an example of a saliency map of a video frame.

The feature we use from the saliency indicator is a mean and a standard deviation of the saliency scores of pixels inside candidate's region. Let s_i be a saliency score of pixel i in an object candidate, a saliency feature SLF is defined as:

$$\overline{s} = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{3.9}$$

$$\sigma_s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(s_i - \overline{s})^2} \tag{3.10}$$

$$SLF = \{\overline{s}, \sigma_s\} \tag{3.11}$$

Compactness indicator A compactness indicator measures how much a shape of object candidate looks like an object. The first key idea is that an object inclines to have a normal square-like shape than a long curved region. The second key idea is that an object tends to have a compact shape than weird scattered blobs. Following the idea of [35], we compute a compactness feature from a modified *scale of region* measurement.

The *scale of region* is defined as the size of the largest possible square that a region can contain. Given a region R, the scale of region is computed as:

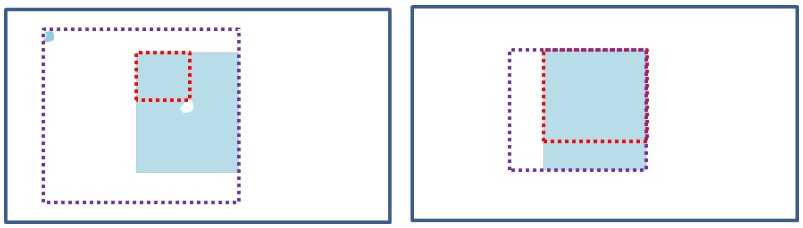

Figure 3.7: Before we compute compactness feature, small holes are filled and small blobs are removed. (Left) Original object candidate region. The $scale(R)$ is very low due to a hole, while the $outer(R)$ is very high because of a small blob. As a result, a compactness feature score is low even if the object has almost a square-like shape (Right) Object candidate region after removing small holes and blobs. The compactness feature score is higher and becomes more reasonable.

$$scale(R) = \arg\max_{t}\{R_{t\times t}|R_{t\times t} \subseteq R\}, \tag{3.12}$$

where $R_{t\times t}$ is a $t \times t$ square region, and $R' \subseteq R$ means there exists at least one location to put R' completely inside R.

The *scale of region* measures how likely an object candidate shape satisfies the first key idea. To satisfy the second key idea, we measure the ratio between the *scale of region* and the smallest square that covers the candidate region, denoted as $outer(R)$:

$$outer(R) = \arg\min_{t}\{R_{t\times t}|R \subseteq R_{t\times t}\}, \tag{3.13}$$

where $R_{t\times t}$ is a $t \times t$ square region. Finally, the compactness feature CF is defined as:

$$CF = \frac{scale(R)}{outer(R)} \tag{3.14}$$

Figure 3.6 illustrates three types of object candidate shapes: (a) A normal object shape (hat), (b) A long curved region that is unlikely to be an object, and (c) A scattered blobs that is certainly not an object. The ratio between the sizes of red square and violet square in each image is the compactness feature. The shape of hat results in highest compactness feature score, while other shapes have much lower scores.

Please note that we fill small holes and remove small blobs from the region R before we compute a compactness feature (Figure 3.7). Small holes and blobs make compactness feature unstable because small holes greatly affect the $scale(R)$ and small blobs greatly affect the $outer(R)$.

Given the training features $\{LF, SF, SLF, CF\}$, we feed the features into a random forest classifier [8] with 1000 trees. We use cross-validation scheme with training and testing sets at the ratio of 70-30. In testing phase, we compute a score of each candidate region in each video frames. Since an object candidate is a set of regions in a consecutive video frames (a time-space volume), the total object candidate score

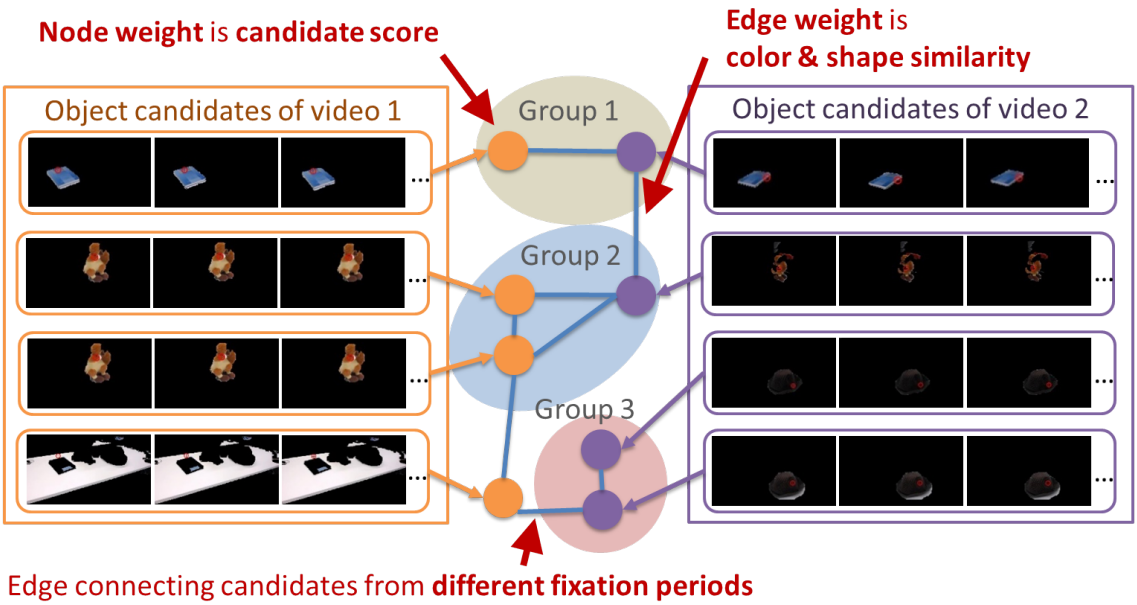

Figure 3.8: Grouping object candidates using graph-based clustering.

is obtained by averaging the scores over all of their regions. The candidates that have lower score than a predefined threshold are removed, leaving only high-scored candidates for the next step.

3.3 Grouping object candidates with video object co-segmentation

Candidates of objects obtained from previous step are clustered into several groups. This clustering allows us to detect objects appearing across the videos, i.e., the objects of common interests. We modified the recent video object co-segmentation method [40] to handle fixation hierarchies. The method uses graph-based clustering to group similar candidates together. Each group should represents each object (Figure 3.8).

Specifically, we first construct a graph structure that describes relationships between object candidates. The nodes (candidates) are weighted by candidate scores in order to discourage the selection of candidates that are likely to correspond to background regions. There will be an edge connecting two nodes, where the edge weight is color and shape similarity between the nodes. The shape similarity follows definition in [40]. However, we use a different definition of color similarity. We extract a color feature as a 16-bins color histogram in HSV color space. We define the color similarity as a dot product of H channels in HSV color space:

$$E(X,Y) = \prod_{i=\{L\}} (hist(HSV_i(X)) \cdot (hist(HSV_i(Y))^T), \tag{3.15}$$

where X and Y are two object proposals.

The main difference between our method and [40] is that we do not put an edge between nodes that are temporally overlapped (Figure 3.9). Namely, a set of

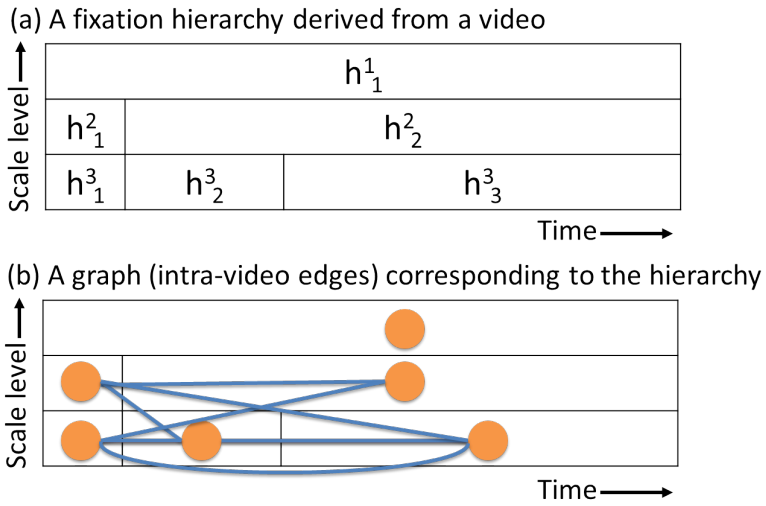

Figure 3.9: A fixation hierarchy and its corresponding graph. (a) An example of a fixation hierarchy derived from a video. (b) A corresponding graph structure (nodes and intra-video edges). There aren't any edges connecting a node to its parent and children, i.e., temporally overlapping nodes.

candidates (nodes) representing the same object in different scales are not connected with each other. This allows us to select one candidate at the most appropriate scale for each object.

After the graph is constructed, we remove edges that have a score lower than a predefined threshold. We denote the threshold as *edge removal threshold*.

We want to cluster nodes in the graph into several groups. Each group should have high consistency in term of color and shape. We use regulated maximum weight clique approach [40] to extract groups of nodes. Let G denotes a graph structure, the definition from [40] is as follows:

Clique is a complete subgraph of G.

Maximal clique is the clique of largest possible number of vertices that cannot be extended by including one more adjacent vertex. In other words, maximal clique is not a subset of any larger clique.

Maximum weight clique is a maximal clique that has maximum weight. The maximum weight clique problem is an NP-hard problem of finding a maximum weight clique, and can be solve by Bron-Kerbosch Algorithm [9] with time complexity of $O(3^{(n/3)})$.

Following [40], we iteratively extract maximum weight clique one at a time. The object candidates in these sub-graphs all have high candidate scores and are similar to each other. In other words, they are highly likely to be the objects of common interests.

Among the results of multiple iterations, it is possible to obtain temporally overlapping sub-graphs. Specifically, one or more candidates in those sub-graphs are temporally overlapped. Since those candidates belong to the same object, we merge those sub-graphs by grouping their candidates together. Among a set of

overlapped candidates, we keep the highest-scored candidate, i.e., the candidate from the sub-graph obtained at the earliest iteration (thus having the maximum weight). Finally, we detect the objects of common interests by selecting the groups that contain nodes from many videos.

3.4 Post-processing

The results we obtain from grouping step are segments that correspond to each object of common interests in each video frame. Since the object segments are constructed by grouping several over-segmented supervoxel segments together based on fixation, the result segments might be incomplete. In order to obtain better accuracy, we refine the segments using GrabCut [29].

GrabCut receive two parameters as input: original frame image, and *trimap*. *Trimap* are H-by-W array of labels, where 0 is background, 1 is foreground, 2 is probably background and 3 means probably foreground region. GrabCut uses trimap to built separate foreground and background color models. Then GrabCut use the two models to generate a new better trimap by changing the label of the probably foreground and probably background regions according to the learned color models. The accuracy of the output trimap depends on the precision of foreground color model and background color model obtained from the input trimap. Therefore, we need to precisely label the object region as the foreground pixels and the other regions as the background pixels.

Ideally, the object segment obtained from the grouping step should correspond to foreground pixels. However, the object segment can also contain background region due to errors from candidates construction process. If we label the whole object segment as foreground pixels, the foreground color model can be incorrect due to the background pixels inside the object segment. To solve this problem, we speculate that the pixels that are near to the point of gaze are more likely to belong to the object region, and the pixels that are too far from the point of gaze might belong to the background region. Therefore, we build the foreground color model using only a few pixels around the point of gaze. Those pixels are labeled as 3 (probably foreground).

We compute a tight bounding box that covers the object segment. The pixels outside the bounding box are labeled as 0 (background). The pixels inside the bounding box are labeled as 2 (probable background), except for a few pixels labeled as 3 (probably foreground) mentioned in the previous paragraph. We input this trimap to GrabCut and get the output trimap. The refined object segment is the area that is labeled as 1 (foreground) or 3 (probably foreground) in the output trimap. Figure 3.10 illustrates input and output trimap labeled according to our instruction.

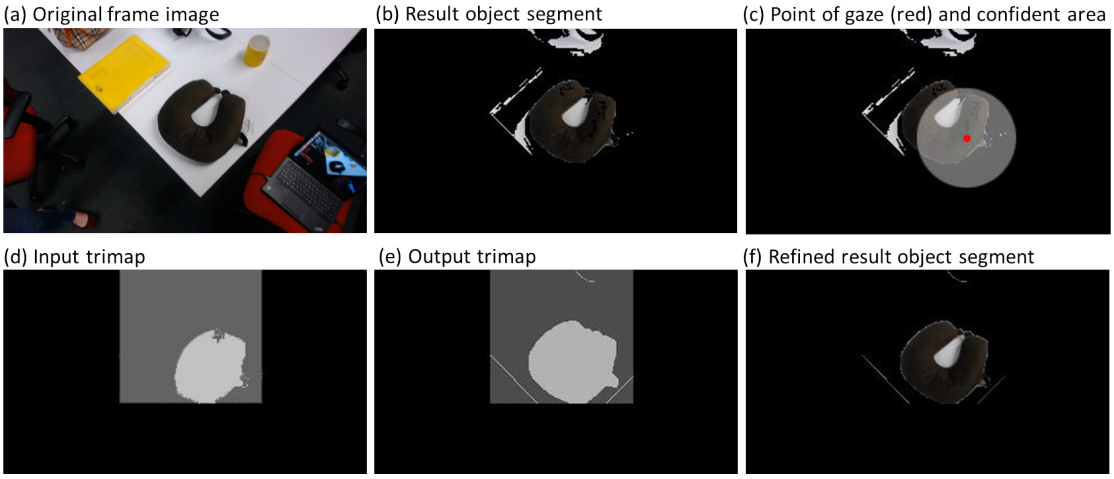

Figure 3.10: An overview of post-processing step. (a) An example input video frame. (b) A detected object of common interests segment. It contains both object and background regions. (c) Point of gaze (red) and region around point of gaze that is likely to contain object region (confident area). We label pixels inside confident area as 3 (probably foreground). (d) Input trimap. Black area is labeled as 0 (background). Dark gray area is labeled as 2 (probable background), and light gray area is labeled as 3 (probably foreground). (e) Output trimap from GrabCut *trimap*. The light gray area (probably foreground) is used as a new refined object segment. (f) Refined result object segment after post-processing.

Chapter 4

Experiments

One of our main contributions is to build a first-person videos dataset. To evaluate our method, we need a dataset that shows joint attentions of the participants in a group activity. There are no prior datasets comprising multiple first-person videos with eye tracking data that suit our need. Therefore, we decide to collect the data and build a novel dataset– Multi-View Multiple Objects (MVMO) dataset. In the first section of this chapter, we explain how we collect the data for our dataset. In the second section, we explain a specific implementation of our proposed method, such as the parameter settings and computing environment. In the third section, we describe three baselines. And, we explain how we evaluate our proposed method in the fourth section. Then we show our results and evaluation measurements in the fifth section. Finally, we describe limitations and analysis of current challenges in the final section.

4.1 Dataset

4.1.1 Multi-View Multiple Objects (MVMO) Dataset

To evaluate our proposed method, we construct a new dataset of multiple first-person videos. The dataset consists of 18 video sets, where each video set contains two first-person-view videos taken at 30 fps with a Pupil Eye Tracker. The videos are recorded at 1280x720 resolution with the length approximately between 40 sec. - 1.20 mins. Each video set has different conditions as follows:

- **Scene**: We record the data in three different environments.
  - Laboratory: Two people stand side-by-side and discuss about items on the nearby table.
  - Secretary room: Two people sit on the opposite side of a table and discuss about objects on the table.
  - Outdoor cafeteria: Two people sit on the opposite side of a wooden table. The recordings are done in open air restaurant, with natural light from the sun.
- **Interaction**: Three types of human interactions are tested.
  - Pick-up: One participate pick up an object while other look at it.

  - Joint attention: Two participants look at each object one-by-one.
  - Free-viewing: Two participant look at any object they want to.

- **Object**: We use two new sets of objects in each scene. Totally, we use six different sets of objects in the whole dataset. Each set of objects contains five different objects. The objects have various colors, shapes, and textures, ranging from the most simplest one-colored object to object with complex color and texture.

The recordings are done by three participants. We ask participants to look at an object one at a time. We also ask them to carefully look at every part of the objects since we want to demonstrate an advantage of using eye tracking data in detecting a whole object region.

We manually label the ground truths as tight bounding boxes that cover the objects of common interests. Then, GrabCut [29] is used to segment the object region. The final segmentation are manually correct by hand to obtain better accuracy.

Figure 4.1 shows a summary of our dataset, including multi-view videos in each video set, and their corresponding ground truth.

Please note that only 17 out of 18 video sets are used to evaluate our method. The reason is because one of them contains too many eye tracking error that it becomes almost impossible to determine locations of objects of common interests.

4.1.2 MVMO Dataset with Synthetic Eye Tracking Data (MVMO-SYN)

We propose another dataset which is a modification of our MVMO dataset. The objective of this dataset is to test our method when eye tracking data does not contain any spatial error. We replace the eye tracking data from MVMO dataset with our synthetic eye tracking data. We artificially generate eye tracking data by hand. The point of gaze in each video frame are manually labeled using mouse pointer. In addition, we try to move the mouse to every parts of the target object to simulate real eye movement pattern.

The artificial eye tracking data are recorded for six video sets, specifically, two video sets per scene representing first two types of interaction: two participants look at each object one-by-one, and one participate pick up an object while other look at it. We also label the ground truth according to the artificial eye tracking data.

4.2 Implementations

We implement our method in Matlab code. We work in Windows 8 64-bits platform. Our videos have a constant frame rate of 30 fps. To reduce running time, we separate videos in a video set into several small subshots, where each one contains 300 frames. The objects of common interests are detected for each subshot, and we evaluate the results of the video set by combining all detected objects of common interests from all subshots together.

In the first step, we detect gaze events from eye tracking data. To resolve the loss eye tracking data, we use linear interpolation technique to interpolate the missing points of gaze when the data is missing for less than 2 consecutive frames ($\sim$66.67

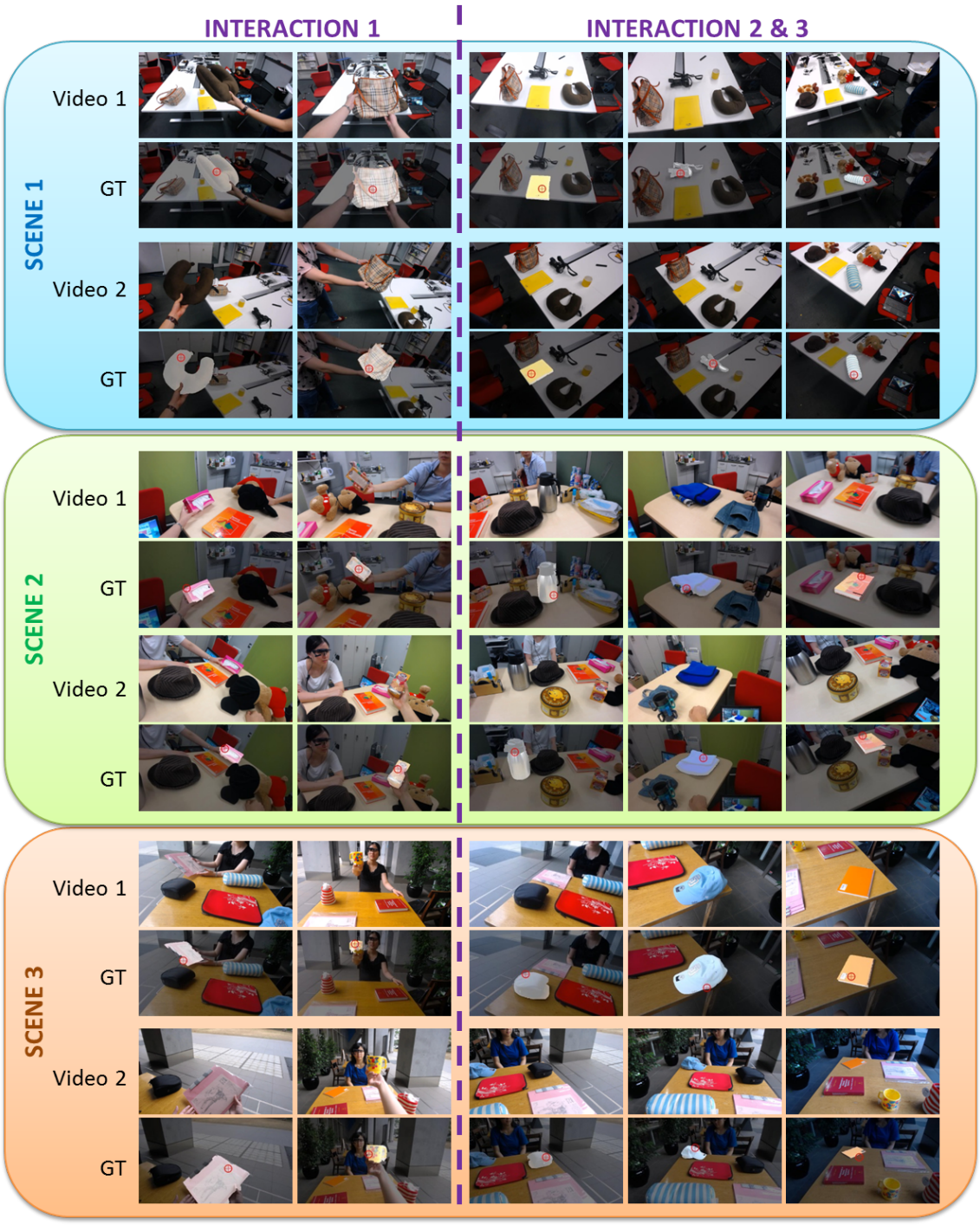

Figure 4.1: A summary of our MVMO dataset. The videos are recorded in 3 different scenes: Scene 1 is laboratory. Scene 2 is secretary room, and Scene 3 is outdoor cafeteria. Interaction 1 is picking-up each object one-by-one. Interaction 2 and 3 are looking at each object, and free-viewing. GT indicates the ground truth data.

ms). We also compensate from head motions in first-person videos. The feature we use to match the video frames is SURF [6]. Then, we segment the video frames and generate the candidates of objects of common interests. We use ovesegmented supervoxel extracted by [38]. Specifically, we use supervoxel at level 5. After we obtain all object candidates, we remove candidates that have score lower than 0.4.

In grouping step, we compute color similarities and shape similarities between candidates, and use them as edge weight. We set edge removal threshold at 0.3.

In post-processing step, we define a confident area as the area around point of gaze that locate within 25-pixel radius.

4.3 Baselines

We compare the proposed method, the approach based on a hierarchy of fixations (HF), with 2 baselines:

A recent Generative Multi-Video Model (GMM) method [11]
We test our MVMO dataset with the state-of-the-art video object co-segmentation method of [11]. We define objects of common interests as the regions that have the same class label across multiple videos.

Naive objects of common interests detection based on a hierarchy of supervoxels (HS)
Our naive approach detects objects of common interests using eye tracking data on video object co-segmentation framework [40]. This approach is almost the same as our proposed method. The difference is that the way we generate object candidates is *lightly* relies on eye tracking data. We discover object candidates from a hierarchy of supervoxels [38]. Lower segmentation levels correspond to more finely segmented regions, while higher levels correspond to more coarsely segmented regions. This naive approach assumes that good object candidates are present at least in one level in the hierarchy of supervoxels. Good candidates should represent whole object segments, while bad ones are fragments of objects or background.

First, a sequence of fixation periods is obtained by segmenting points-of-gaze stream temporally based on the angular velocity. Second, We use a series of fixation periods to generate the candidates of objects being looked at by wearers. We divide the video into several short sequences of frames based on fixation periods. For each level of supervoxels in each fixation sequence, we select the supervoxel segment that are looked at by the camera wearer for the maximum number of frames as the final object candidate. Then, the object candidates are scored and the low-scored ones are removed (Section 3.2.3). Finally, we cluster all object candidates in the same way as our proposed method (Section 3.3) and do the post-processing (Section 3.4). Figure 4.2 illustrates our naive method.

Please note that, while our method is based on [40], running [40] with supervoxel hierarchies was infeasible as their code required too heavy computational resource.

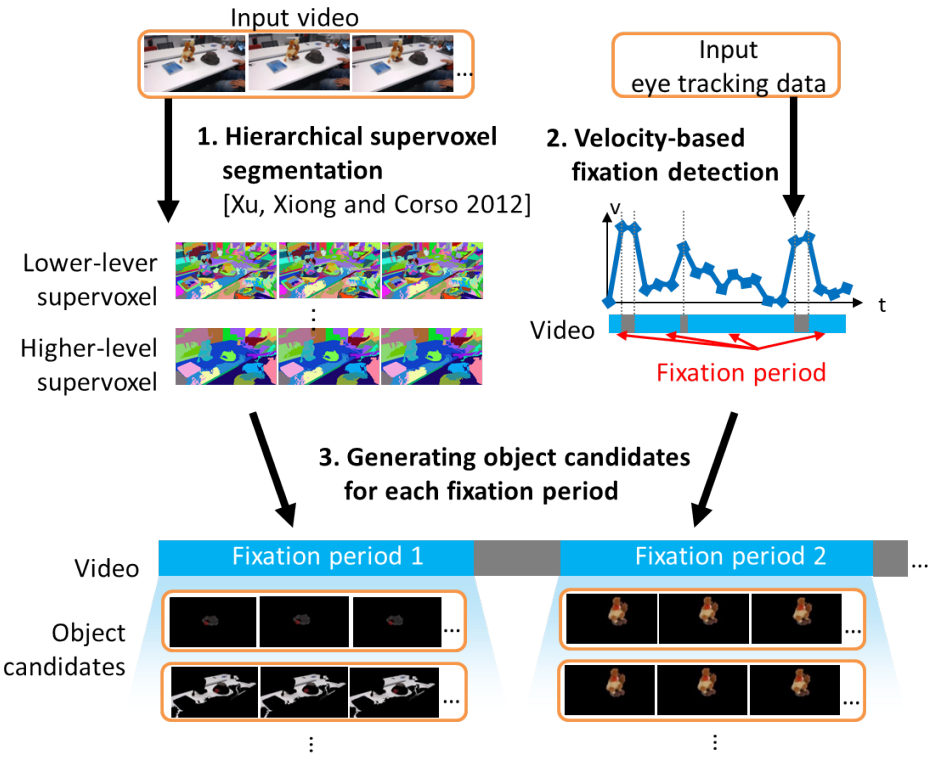

Figure 4.2: A naive objects of common interests detection based on a hierarchy of supervoxels (HS)

4.4 Evaluation methods

We design two evaluation methods:

Per-pixel segmentation performance evaluation
In this scheme, we evaluate the performance of our method pixel-by-pixel. In other words, this measurement evaluate our method in regard to per-pixel segmentation performance. Pixels in each video frame are labeled as 0 (not object of common interests) or 1 (object of common interest). After all video frames are labeled, all labels are concatenated together into a single vector. Then, we compare it to the ground truth vector constructed in the same way. Finally, we compute precision, recall, and F-score:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Localization performance evaluation
Per-pixel segmentation performance measurement is not the best way to evaluate our method. It cannot cope with different interpretations of objects of common interests. For illustration, Figure 4.3 shows example video frames and three pairs of regions that can be used as ground truth (the object of common interests). The video frames (fig. 4.3(a)) are recorded from two viewpoints, showing two participants looking at a bottle of tea. The red crosses are the points of gaze of the participants. Figure 4.3(b) is one possible ground truth, specifying that two people are looking at

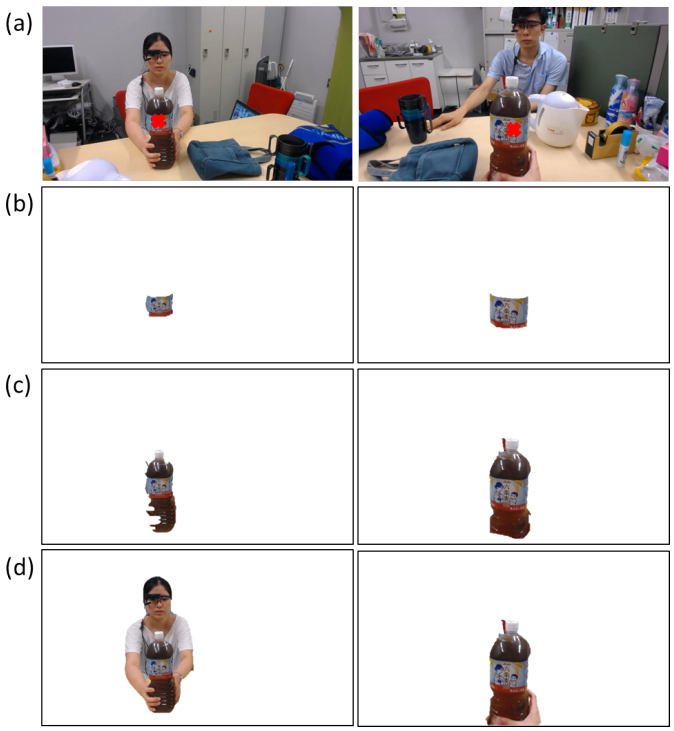

Figure 4.3: Example video frames and three possible interpretations of ground truth (object of common interests): (a) Original video frames from two viewpoints and points of gaze (red cross), (b) Interpretation 1, (c) Interpretation 2, and (d) Interpretation 3.

the logo on the bottle. Another plausible ground truth is Figure 4.3(c), which shows a whole bottle. The third probable interpretation is Figure 4.3(d), which assumes that all connecting areas represent one whole object. Therefore, the participant and the bottle is labeled together as the object of common interests.

Because there are many plausible ground truths, per-pixel evaluation method tends to be biased. To solve this problem, we propose a novel evaluation method. Our evaluation method measures how effective the proposed method can locate objects of common interests, while ignores the per-pixel segmentation performance.

In each video frame, there are seven situations that possibly occur when we evaluate our method. Figure 4.4 illustrates the seven situations. R is the result (object of common interests) detected in the frame. GT is the ground truth, which is the whole object region (e.g., the whole bottle of tea). In Figure 4.4(a), the detected region is a part of the object, which corresponds to Interpretation 1 in Figure 4.3(b). In Figure 4.4(b), the detected region is deviated from the correct result. It includes both a part of the object and a small background region. Figure 4.4(c) is similar to Figure 4.4(b), except that the background region is large. In Figure 4.4(d), the detected region covers the ground truth region. Figure 4.4(e) is similar to Figure 4.4(d), but the detected region includes a large background region. Figure 4.4(f) is true negative case, where our method fails to detect the object of common interest. Finally, Figure 4.4(g) illustrates false positive case, where our method incorrectly labels background region as the object of common interests.

We believe that Figure 4.4(a), Figure 4.4(b), and Figure 4.4(d) are acceptable results. In these cases, the detected regions cover most of the ground truth area.

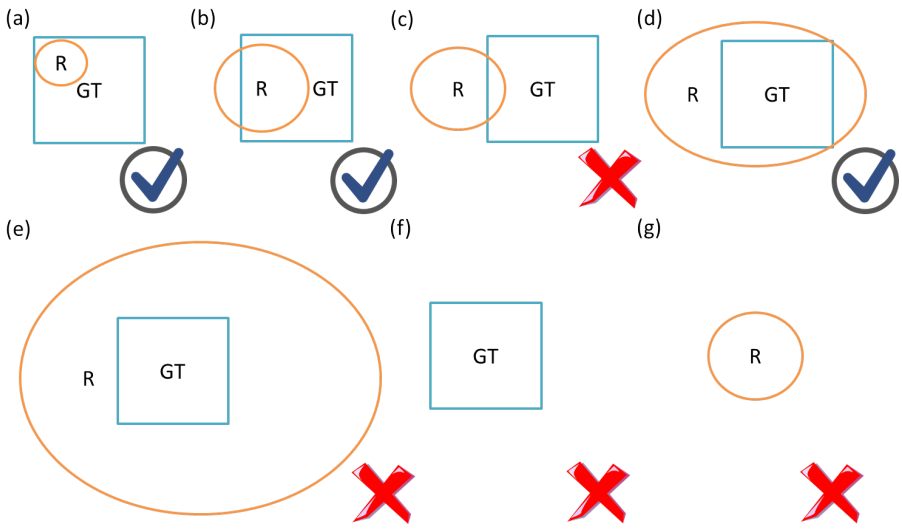

Figure 4.4: Seven situations that can occur when we compare our detected region R with ground truth GT in each video frame. In this case, ground truth represents a whole object region, e.g., a whole bottle of tea. (a) Detect region is a part of the object, (b) Detect region overlaps a part of the object more than 50 percent, (c) Detect region overlaps a part of the object less than 50 percent, (d) Detect region covers the object, (e) Detect region covers the object, but it also include background region for more than 50 percent, (f) Miss detection (true negative case) and (g) Incorrect detection (false positive case).

More specifically, more than 50% of the detected regions are ground truth region. We use this definition to create a new evaluation method. First, we define *overlapping score* between detected region R and ground truth GT:

$$OVERLAP = \frac{R \cap GT}{R} \tag{4.1}$$

For ground truth, we label each video frame as 1 if an object of common interests exists in the frame, and 0 if there aren't any objects of common interests.

For the result of our method, we label each video frame as 1 or 0 according to its *overlapping score*:

$$LABEL = \begin{cases} 0, & \text{if } OVERLAP < 0.5 \text{ or } area(R) = 0 \\ 1, & \text{if } OVERLAP >= 0.5 \text{ or } area(GT) = 0 \end{cases} \tag{4.2}$$

We define two exceptions for true negative and false positive cases. First, $area(R) = 0$ is true negative case (Figure 4.4(f)), which results in *divide by zero* when we compute *overlapping score*. The frame label will be 0 since the method fails to detected the object of common interests. Second, $area(GT) = 0$ is false positive case (Figure 4.4(g)). The frame label will be 1, while the ground truth will be labeled as 0.

After we obtain all frame labels, we compare them with ground truth labels. Then, we compute precision, recall, and F-score. With our evaluation method, precision score will measure how many detected regions are objects of common interests. And, recall score will represent how many objects of common interests are detected.

	Per-pixel segmentation			Localization		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
GMM	0.028	0.771	0.053	0.001	0.001	0.001
HS (Ours)	0.772	0.193	0.298	0.779	0.250	0.369
HF (Ours)	**0.551**	**0.385**	**0.434**	**0.613**	**0.549**	**0.564**

Table 4.1: Comparison between our method and baselines.

4.5 Results & Discussion

In this section, we provide evaluation metrics on our MVMO dataset and MVMO-SYN dataset. We also provide a comparison between our method and baselines. In addition, we analyze the performance of our proposed method in various conditions.

Table 4.1 shows the precision, recall, and F-score of our method and baselines. GMM denotes Generative Multi-Video Model approach [11]. HS is our naive approach based on a hierarchy of supervoxels (Section 4.3). HF is our proposed method, which relies on a hierarchy of fixations. Our method achieve the best performance compare to other baselines.

GMM method cannot differentiate objects from background regions when background regions are similar across viewpoints. Most of the results contain both the objects and background regions. As a result, GMM method usually has a high recall and a very low precision. GMM has very low localization metrics due to the same reason. In most GMM results, the detected region is overlap with ground truth for less than 50% of its total area. Thus, they do not pass the localization performance evaluation criteria. Another limitation of GMM method occurs when there are many objects in the scene. GMM method cannot determine which one object is the object of common interests since there are many possible choices. That is, any objects that can be commonly seen from many videos are all detected as the objects of common interests. Therefore, GMM method gives the very low precision. This emphasizes the advantage of using eye tracking data, which can help locating a single target object from each video frame.

HS method constructs object candidates using eye tracking data. By comparing GMM with HS method, we can conclude that eye tracking data is remarkably beneficial for detecting objects of common interests. It helps locating the important regions that are likely to be the target objects. By combining eye tracking data with video object co-segmentation framework, HS method gives much better performance than GMM method.

Unfortunately, HS approach still has a limitation. HS method generates various sizes of object candidates from a hierarchy of supervoxels. It assumes that there is a good candidate in at least one level of supervoxel. However, this assumption does not always hold true. Consequently, HS method provides low recall when the assumption fails.

Our proposed method (HF) constructs object candidates by combining a lot of regions together. It relies on a hierarchy of fixations. HF method performs better than HS approach, thus signifies the importance of a fixation hierarchy. Still, the performance of HF method heavily depends on an accuracy of eye tracking data. It achieves the best performance when the eye tracking data does not contain any spatial error. Also, the camera wearer needs to look at every parts of the target

	Per-pixel segmentation			Localization		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
MVMO	0.551	0.385	0.434	0.613	0.549	0.564
MVMO-SYN	**0.867**	**0.516**	**0.628**	**0.898**	**0.579**	**0.691**

Table 4.2: Performance of our proposed method on MVMO and MVMO-SYN.

	Per-pixel segmentation			Localization		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Laboratory	**0.599**	**0.583**	**0.588**	**0.685**	**0.675**	**0.676**
Secretary room	0.535	0.249	0.331	0.594	0.443	0.494
Outdoor cafeteria	0.526	0.356	0.409	0.571	0.550	0.542

Table 4.3: Performance of our proposed method on MVMO dataset grouped by scene.

object.

Table 4.2 presents a comparison of our method on MVMO and MVMO-SYN dataset. It is clear that MVMO-SYN gives better scores. It is because the eye tracking data in MVMO-SYN dataset is very precise and the points of gaze are scattered through all parts of the target objects.

Figure 4.5, Figure 4.6, and Figure 4.7 visualize the performance of the baselines and the proposed method on our MVMO dataset. The fist column shows the input video frames. Second column is the ground truth regions, which are overlaid on the input frames. Red crosses denote the points of gaze recorded from the participants. Third column is the results of Generative Multi-Video Model (GMM) approach [11]. Each color represents each object label. If two regions from different videos have the same color, the two regions will be regarded as an object of common interests. Fourth column is our naive approach based on a hierarchy of supervoxels (HS). And, the last column shows the results of the proposed method. We highlight the detected objects of common interests with a light color, and cover them with yellow bounding boxes.

Table 4.3 depicts the average precision, recall, and F-score grouped by scene. We can see that the video sets recorded in laboratory room get better score than the ones recorded in the secretary room and the outdoor cafeteria. The cause of these differences is the relative distances between the camera wearers and the objects. The camera wearers are standing in the laboratory, while they are sitting in the secretary room and the outdoor cafeteria. As a result, the cameras are relatively further from the objects in the laboratory scene than in other scenes. When objects are far from the cameras, they become smaller. Thus, the camera wearers can easily look at every parts of the object within a few glances. In contrast, camera wearers need to move their eyes and head a lot in order to look at every part of the larger objects. As a result, it is more difficult to cover the larger objects in the secretary room and the outdoor cafeteria scenes.

Our HF method constructs an object candidate by grouping together all parts being looked at by the camera wearer. The method can construct a complete candidate, which represents the whole object, if the camera wearer looks at every part of the target object. Therefore, the probability of getting complete object candidates in the laboratory scene is higher than in other scenes. The evaluation metrics in

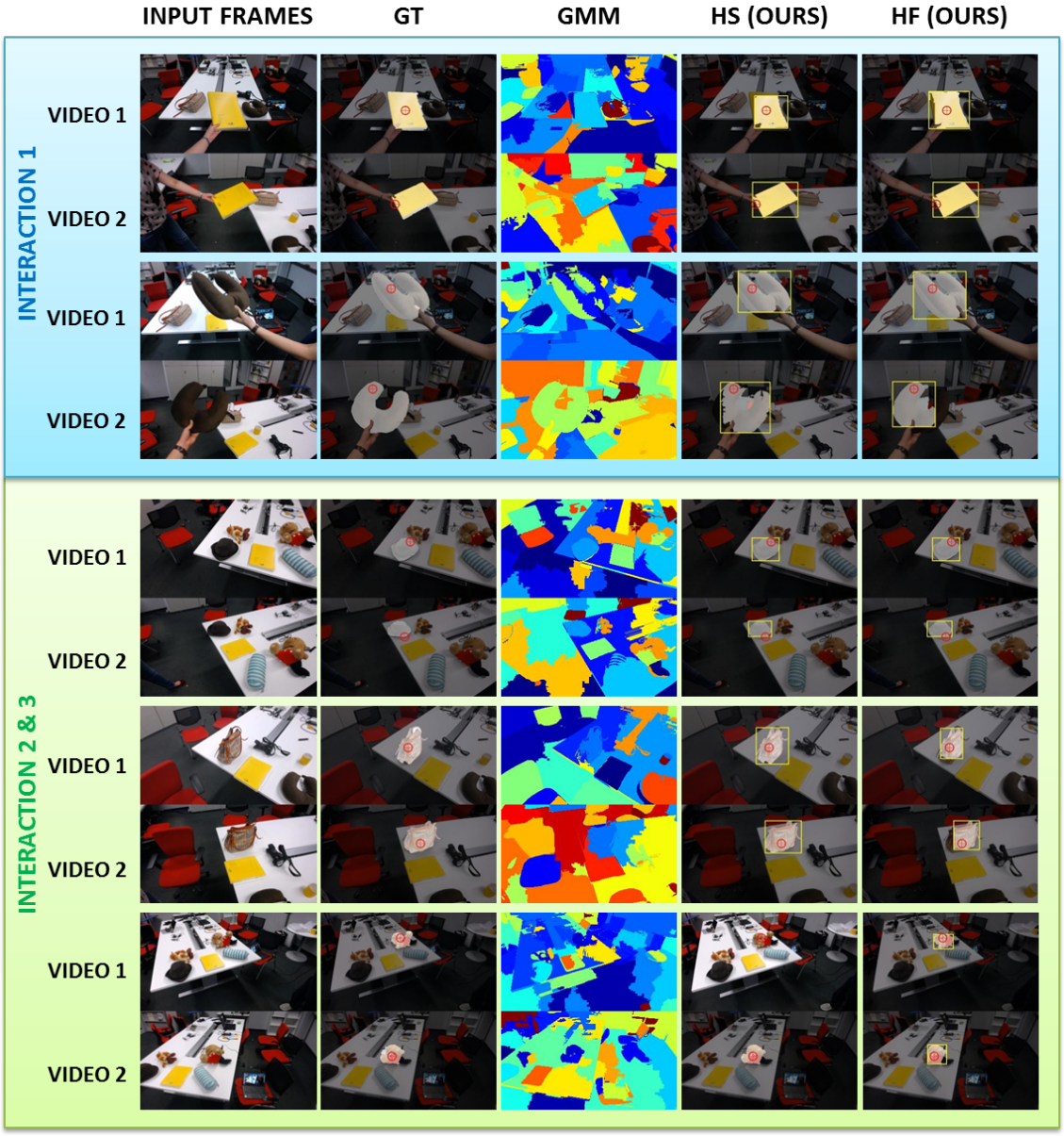

Figure 4.5: Example results from laboratory scene. Column 1 is the input video frames, Column 2 ('GT') denotes the ground truths, Column 3 ('GMM') is the results from Generetive Multi-Video Model approach [11], Column 4 ('HS') is our naive hierachical supervoxels method, and Column 5 is our proposed method.

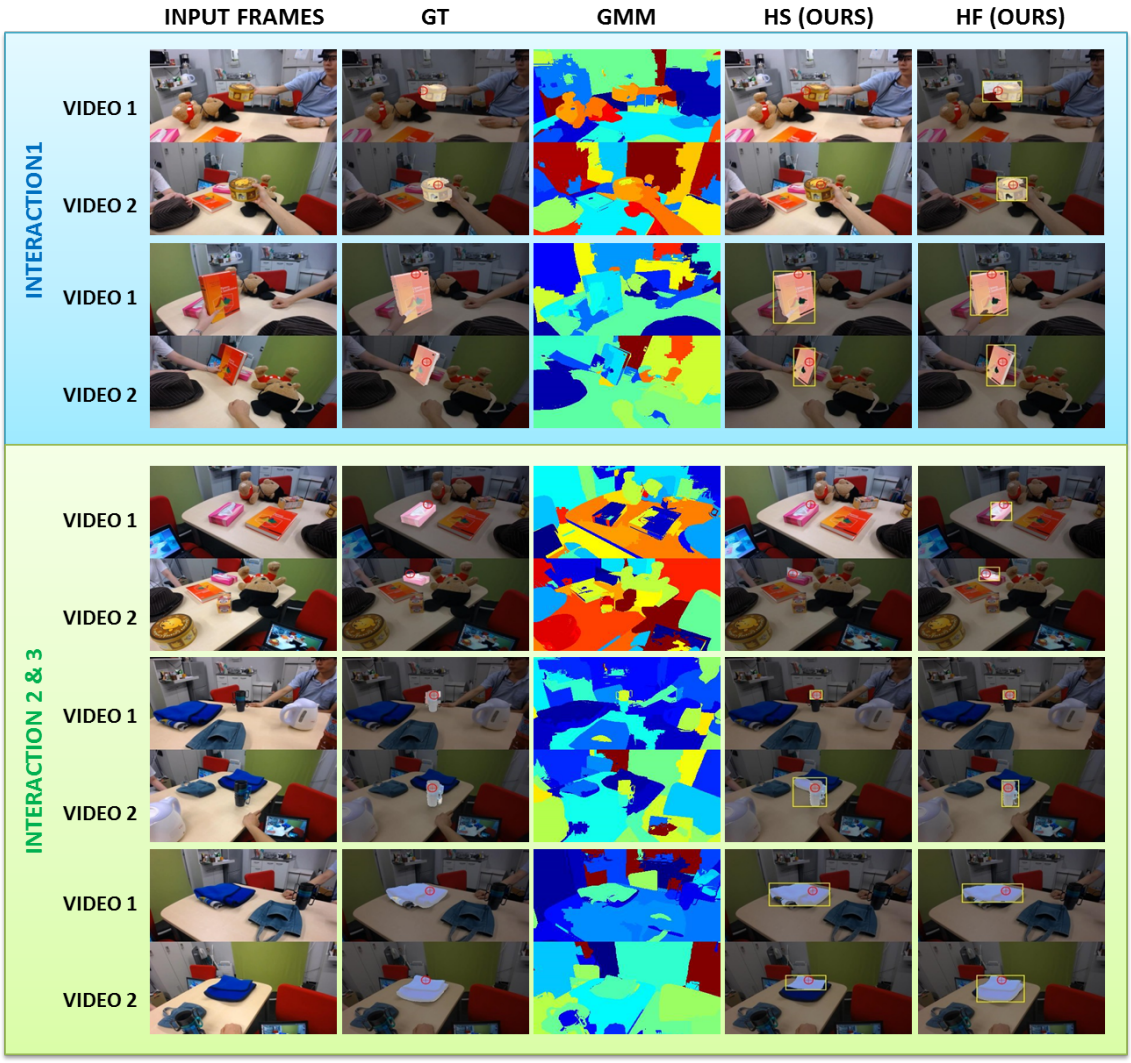

Figure 4.6: Example results from secretary room scene. Column 1 is the input video frames, Column 2 ('GT') denotes the ground truths, Column 3 ('GMM') is the results from Generetive Multi-video Model approach [11], Column 4 ('HS') is our naive hierachical supervoxels method, and Column 5 is our proposed method.

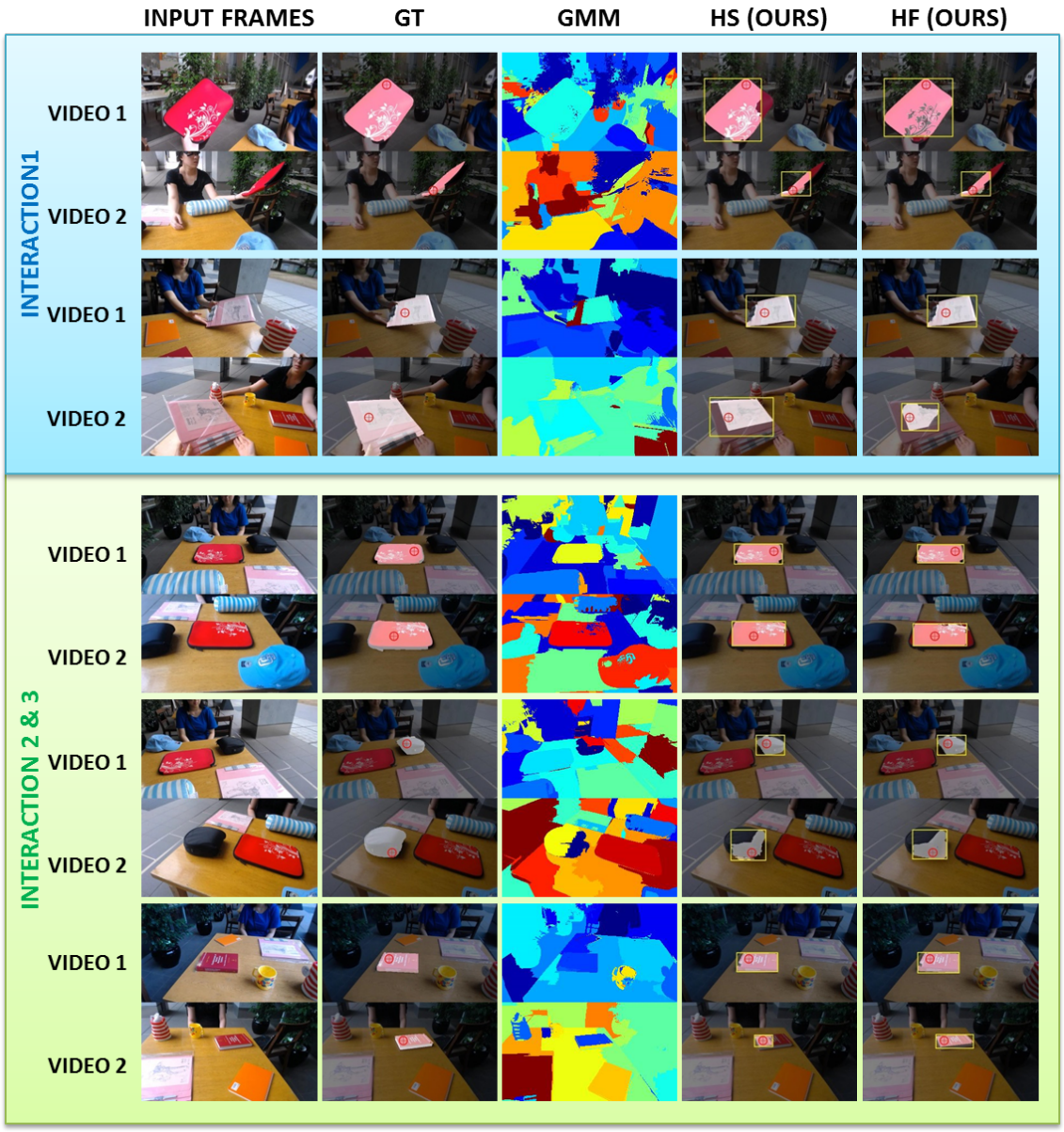

Figure 4.7: Example results from outdoor cafeteria scene. Column 1 is the input video frames, Column 2 ('GT') denotes the ground truths, Column 3 ('GMM') is the results from Generetive Multi-video Model approach [11], Column 4 ('HS') is our naive hierachical supervoxels method, and Column 5 is our proposed method.

	Per-pixel segmentation			Localization		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Pick-up	**0.659**	**0.408**	**0.481**	**0.710**	**0.549**	**0.601**
Joint attention	0.546	0.408	0.451	0.610	0.554	0.577
Free-viewing	0.427	0.329	0.356	0.499	0.542	0.506

Table 4.4: Performance of our proposed method on MVMO dataset grouped by type of interaction.

Table 4.3 support the aforementioned reason.

Table 4.4 displays the evaluation metrics of our proposed method on MVMO dataset based on type of interactions. The best results are from the video sets that show *Pick-up* interaction; one camera wearer picks up the target object, while others look at it. When the camera wearers pick up an object, they usually position the object in a way that they can see it clearly. For example, they tend to move the object to the center of their field of view, and rotate it to get the best view. As a result, our method can easily obtain good object candidates in this interaction. The object candidates of the target object can obtain high scores from location, size and saliency indicators.

The worst results are from the video sets that show *Free-viewing* interaction; camera wearers move the eyes naturally as they pleased. In this interaction, there are a lot of eye tracking error due to excessive head motions. In addition, the camera wearers tend to look at an object and then switch to its surrounding (i.e., background regions). Therefore, our proposed method is less likely to obtain complete object candidates from this type of interaction.

The aforementioned analysis indicates several problems that limit our method performance. However, it also indicates a potential for further improvements. In Section 4.6, we provide further analysis on current problems. And, suggestions for future development can be found in Section 5.2.

4.6 Performance Analysis & Limitation

We present a deep discussion on several issues that affect our method performance, the limitation of our method, and various ideas for improvement.

4.6.1 Types of error

In this section, we describe types of error and their effects on our algorithm. There are mainly six types of error that commonly appear in our method:

Error from supervoxel segmentation
We construct object candidates from over-segmented supervoxels. However, it is difficult to decide how much 'over-segmented' it should be. In our work, we segment each video frame by hierarchical supervoxel segmentation approach [38]. Then, we select one low level supervoxel, and use it to construct the candidates based on fixation events. If the supervoxel level is too low, we will get many small supervoxel segments. Consequently, it will be difficult to obtain a complete object candidate (i.e., a candidate that contains a whole object region) because we will need to look

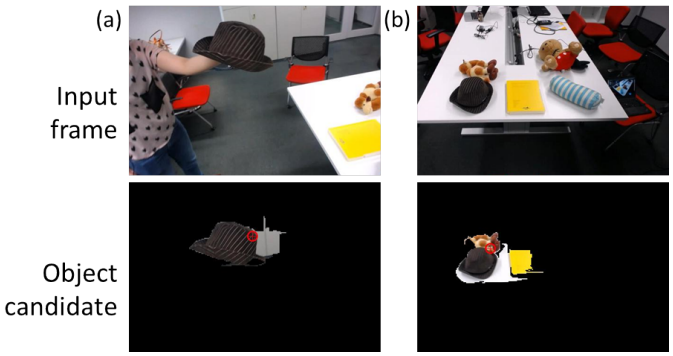

Figure 4.8: Low-quality object candidates due to two types of error: (a) Spatial error of points of gaze, and (b) Error from fixation segmentation.

at every supervoxel segments inside the object region. In contrast, if the level is too high, some supervoxel segments will contain both object regions and background regions, resulting in under-segmented object candidates.

Error from eye tracking data
There are two types of error from eye tracking data:

- **Spatial error of points of gaze**

  Points of gaze contain spatial error. They sometimes are incorrectly located near the border of the target object. In our work, object candidates are constructed by merging all supervoxels that are looked at within each fixation period. The spatial error of points of gaze will result in an object candidate that contains both object region and its surrounding background (Figure 4.8(a)).

  In the worst case, the points of gaze are totally located outside the object region for the whole duration. Normally, people tend to infer the target object using logical reasoning and experiences to guess the location of the target object. However, it is much more difficult for the computers, especially when it needs to work in an unsupervised manner. This type of error is not in the scope of our work. Please also note that the current proposed method is unable to generate an object candidate when this type of error exists.

- **Eye movement pattern**

  We assume that people tend to look at every parts of the object that they are interested in. If a participant does not look at a whole object, our method cannot generate a complete object candidate. Also, if the participant rapidly swap his/her gaze between two or more objects continuously, we will obtain many incomplete object candidates. Each candidate will consist of only a small part of the object.

Error from fixation segmentation
We construct object candidates from a hierarchy of fixation. Thus, a correct hierarchy fixation segmentation result is required. Fixations in higher levels of the hierarchy tend to provide under-segmented objects candidates, i.e., object candidates that consist of both object region and background region. Also, fixations

in lower levels of the hierarchy incline to generate over-segmented candidates, i.e., candidates that contain only a part of the object.

In addition, the fixation segmentation approach can incorrectly segment two fixation periods together into one long period. As a result, two objects are combined together as one object candidate (Figure 4.8(b)). This error occurs because the speed of eye motion between the two periods are too small. It usually occurs when two objects are located near each other, or are overlapped from the viewpoint of the camera wearer.

Error from candidate scoring

We propose a regressor to compute scores of object candidates, and remove low-scored candidates since they are likely to contain background region. However, it is still difficult to completely eliminate all bad object candidates. First, a threshold that we use to remove the candidates can be different in each video set. If the threshold are too low, we will get a lot of bad object candidate, resulting in low precision and high recall. On the other hand, we will eliminate good candidates if the threshold is too high, resulting in high precision but low recall. The threshold also vary depending on which features are used as input to the regressor. The discussion about problem of each feature are presented in the next section.

Error from candidate grouping

In grouping step, we use graph-based clustering approach to group object candidates based on color similarity and candidate scores. Error in this step occurs when similarity scores and candidate scores are incorrect due to over-segmented candidates. Color feature derived from a background region in the over-segmented candidate does not represent the real color of the object. Thus, the color similarity score becomes less meaningful. The problem intensifies when there are a lot of over-segmented candidates, resulting in a high probability of incorrect matches.

More importantly, two object candidates that represent two distinct objects can be incorrectly grouped together when the background region inside each candidate is larger than the object region.

Another important problem exists when two or more clusters (sub-graphs) are merged together when they are temporally overlapping (as explained in Section 3.3). If any of them contains wrong matches. the merged cluster will also become erroneous. It can be said that clusters merging step escalates this type of error.

Error from post-processing

We use GrabCut [29] to refine the result segments. However, GrabCut can sometimes fail to segment the object from background. The precision depends on the input trimap that is used to build foreground and background color models. If the trimap includes a lot of background regions, GrabCut will fail to obtain correct color models, thus result in incorrect segmentation. We have mentioned in Section 3.4 that we only use a few pixels around gaze point to make trimap in order to avoid background pixels. The challenge is how many pixels are needed, and how accurate the point of gaze is.

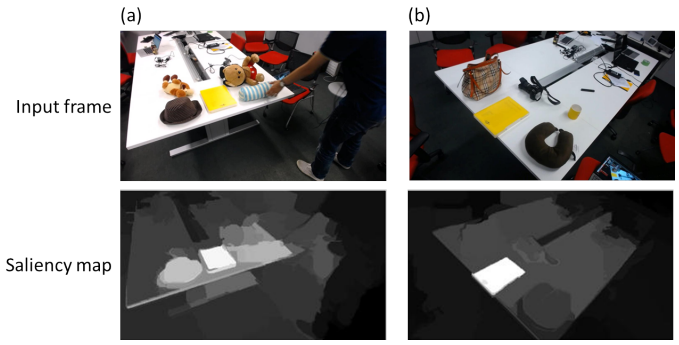

Figure 4.9: Input video frames and their corresponding saliency map: (a) Correct saliency map, (b) Incorrect saliency map.

4.6.2 Features selection problem

In Section 3.2.3, we introduce a scoring function that measures the quality of object candidates. The scoring function is a random forest regressor based on a set of features, namely location, size, saliency and compactness indicators. Therefore, we need to select the best subset of features that can precisely reflect the quality of object candidates.

Furthermore, we introduce a color feature to measure candidates similarities in grouping step (Section 3.3). As a result, we need to find the best way to represent the color of object candidates as a feature vector.

In this section, we discuss the pros and cons of each selected features.

Location & Size indicators

Location & Size features prefer large object candidates that located near the center of the video frame. This works well in most cases because people tend to rotate their head so that the focused object is located at the center of their field of view.

However, the size indicator may work in a negative way when the target object is very far from the camera wearer. For example, when camera wearer walks in the street and looks at a faraway landmark (e.g., Tokyo Skytree). The target object (Tokyo Skytree) might be very small and might not always be at the center of the frame. In this case, the location & size indicators will give a low score.

Another problem rises when the target object is moving rapidly (e.g., running car). In this case, the target object can be located at any places around the frame. As a result, the location indicator will give a low score when the target object moves near the frame border. Anyway, please note that locating moving object is not in our scope of work. This will be left for future improvements.

Saliency indicator

Saliency indicator prefers object candidates that are spatially distinct from their surrounding regions. This indicator fails when the target object has low visual salience. For example, objects with dark colors are less distinguishable than light-colored objects.

Figure 4.9 shows two video frames and their saliency maps. Light-colored regions denote highly salient areas. Figure 4.9(a) illustrates a correct saliency map. All objects have high saliency score. Figure 4.9(b) is a failure case, where only light-

	H (16 bins)	H (24 bins)	A,B (8 bins)	A,B (16 bins)
Precision	**0.484**	0.456	0.454	0.470
Recall	**0.397**	0.366	0.360	0.343
F-score	**0.424**	0.393	0.391	0.385

Table 4.5: Average precision, recall and F-score from per-pixel segmentation performance evaluation. The test is done for 4 cases: 16-bins H channel in HSV color space, 24-bins H channel in HSV color space, 8-bins A,B channel in LAB color space, and 16-bins A,B channel in LAB color space.

colored object, namely yellow notebook, has high saliency score. The remaining objects have dark color, resulting in lower saliency scores.

Compactness indicator
Compactness indicator prefers square-like, non-scattered object candidates. It works well in most cases because most background regions have uncommon shapes. However, it sometimes gives high ratings to square-like background regions.

Color feature
The parameters that should be considered when using color feature are color space and the number of bins. We decide to use H channel from HSV color space. We do not use S and V channels because the target object can have different saturation and lightness based on viewpoints. The problem mainly arises when the object surface is reflective. The target object reflects different amount of light according to the relative directions between the camera wearers and the light-source. Consequently, the target object in different viewpoints shows different colors. We can avoid this problem by selecting the suitable channels and color space when we construct the color feature vector.

The number of bins decide how strict we want to compare the similarity between two object candidates. If the number is too large, two candidates that correspond to the same object may have low color similarity. If the number is too small, two irrelevant candidates may be grouped together due to high color similarity. We believe that 16 bins is the appropriate number in our case.

Table 4.6.2 shows the average precision, recall, and F-score from per-pixel segmentation performance evaluation. The test is done on 12 out of 18 video sets. Four combinations of color spaces and numbers of bins is used: 16-bins H channel in HSV color space, 24-bins H channel in HSV color space, 8-bins A,B channel in LAB color space, and 16-bins A,B channel in LAB color space. 16-bins H channel in HSV color space gives the best result.

4.6.3 Limitation of video co-segmentation in multi-view videos scheme

Video object co-segmentation segments a target object from multiple videos by assuming two assumptions: (1) the target object has similar appearance throughout all videos, and (2) Background in each video are different from background in other videos.

In our case, we need to detect a target object across multiple viewpoints. The target object may have varying appearance across viewpoints. Also, the backgrounds can be similar to one another depending on the points of view because the videos are recorded in the same scene. As a result, the two assumptions that make video co-segmentation works well might not be completely true.

To solve the problem, we need a similarity measure that can cope with appearance differences due to the change of viewpoint. Moreover, we need a way to eliminate background regions when video co-segmentation fails to detect background regions. With our proposed method, we define a 16-bins color similarity as the similarity measure. We also introduce a regressor to score object candidates (Section 3.2.3), and eliminate background candidates according to the scores.

While our method can partly solve the problem, we believe that there is a room for improvement. Currently, there are several works [2][12][15][34][41] that try to match an object in multi-view videos. One idea is to find a set of object features that is viewpoint invariant. Another idea is incorporate geometric methods, e.g., by calibrating camera sensors and compute the transformation matrices to match the viewpoints. These two ideas, however, still need more research and will be left for future directions.

4.6.4 Effect of edge removal threshold in candidate grouping step

Edge removal threshold provides a trade-off between region completeness and detection rate. High threshold means that two object candidates must be highly similar in order to be clustered into the same group. Low threshold relaxes the similarity requirement, thus allows low-similarity candidates to be grouped together.

Figure 4.10 illustrates the detected results when the edge removal threshold equals 0.1, 0.3, 0.6, and 0.9. The highlighted regions are the object candidates. If an object of common interests is detected, the regions will be covered by yellow bounding boxes. Figure 4.10(a) and Figure 4.10(e) shows the results when the threshold is too low. Irrelevant regions are clustered together, and are detected as an object of common interests. The threshold between 0.3 and 0.6 provides quite similar results (Figure 4.10(b), Figure 4.10(c), Figure 4.10(f), and Figure 4.10(g)). However, the detection rate falls rapidly when the threshold exceeds 0.9 (Figure 4.10(d)).

Figure 4.11 shows an effect of edge removal threshold on the precision, recall, and F-score. Top left graph illustrates the precision when the edge removal threshold changes from 0 to 1. The higher threshold results in the higher precision. At the same time, the recall becomes lower as the threshold increases (Top right graph of Figure 4.11). However, the recall rapidly falls when the threshold is too high because almost all of the edges are removed from the graph. Bottom left graph shows that F-score remains stable until the threshold becomes 0.8. This is because the well-balanced trade-offs between the precision and recall. When the threshold gets higher than 0.8, recall becomes so low that F-score subsequently falls. The bottom right graph presents the precision-recall curve.

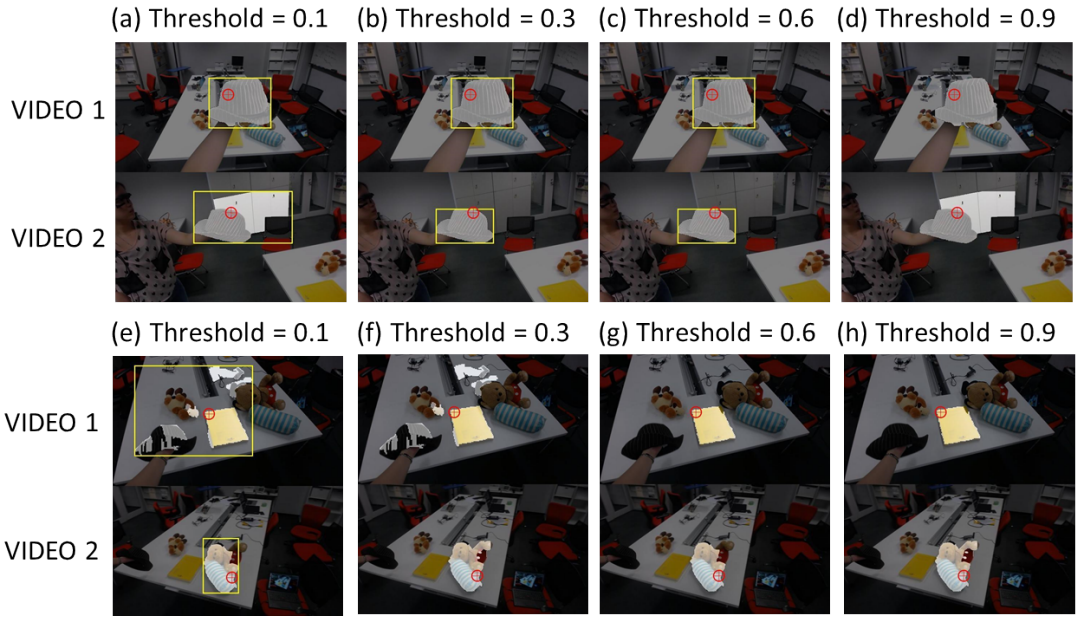

Figure 4.10: Detection results with edge removal threshold = 0.1, 0.3, 0.6, and 0.9 respectively. The highlighted regions are the object candidates. Yellow bounding boxes denote that the highlighted regions are detected as an object of common interests.

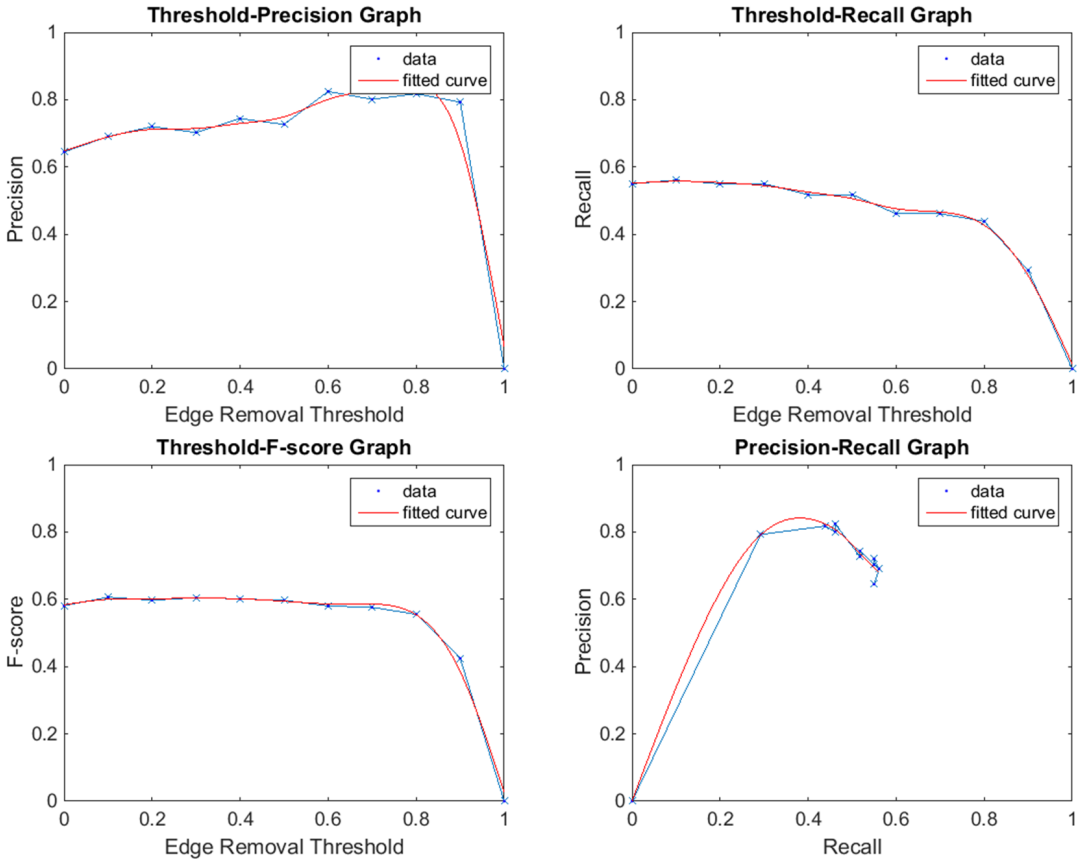

Figure 4.11: Effect of edge removal threshold on the precision, recall, and F-score.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we introduce a novel problem of detecting objects of common interests and develop a method to solve the problem. The objects of common interests are good cues to determine group attentions.

Our proposed method utilizes eye tracking data on a video object co-segmentation framework. The main idea is to use eye tracking data to guide object candidates construction process. First, several object candidates are generated based on gaze events. We combine a set of supervoxel segments according to each fixation event to form an object candidate. Second, the candidates are feed to the video object co-segmentation framework. A graph-based clustering approach is used for detecting the objects of common interests. Finally, we re-segment the detected objects of common interests using GrabCut [29] to obtain the final results.

Another main contribution of our work is a newly recorded dataset, Multi-View Multiple Objects (MVMO) Dataset, which is used for evaluating our proposed method. The dataset consists of 18 sets of first-person videos and their accompanying eye tracking data. Each set shows joint attentions of two participants on five objects. Moreover, we provide a modified version of the MVMO dataset, MVMO-SYN dataset. The eye tracking data of the MVMO-SYN dataset is manually synthesis in order to avoid eye tracking error.

We provide two measurements for evaluating our method in two aspects, namely, per-pixel segmentation and localization performance. We compare our method with two baselines: a recent Generative Multi-Video Model (GMM) method [11] and our naive objects of common interests detection approach based on a hierarchy of supervoxels (HS). The proposed method outperforms both baselines, which shows that eye tracking data plays an important role in detecting objects of common interests. It provides a significant cue for generating candidates of the objects of common interests, and helps limiting the locations of the candidates in a cluttered scene.

In addition, we present an analysis on several factors that affect our method performance, such as eye tracking error, types of interaction, and the distances between the cameras and the objects. We conclude that eye tracking accuracy and eye movement pattern has a large effect on our proposed method. We also find that the position, size, and location of the objects affect our method performance. This implies a room for future improvements. Moreover, we discuss about the impact of

edge removal threshold, which provides a trade-off between the precision and the recall.

5.2 Future directions

As we discuss major challenges and their causes in the previous chapter, we gain several insights for further development. In this section, we list up our recommendation for future works as follows:

Eye tracking data correction
In previous chapter, we conclude that eye tracking data accuracy has a large impact on our method performance. To reduce the impact, we suggest a deep study on two aspects: (1) Eye tracking error correction, and (2) Eye tracking data segmentation method. The former refers to a method that can correct the spatial error of eye tracking data due to calibration error and head motions. The latter indicates a study on eye tracking data segmentation approaches. The aim is to find the best method that can correctly segment points of gaze stream into a sequence of gaze events.

Gaze event detection
Currently, we detect two gaze events, which are saccade and fixation events. We suggest that another gaze event, namely *smooth pursuit*, should also be detected. Smooth pursuit is a movement of eye that track the moving object. In many cases, smooth pursuit is more suitable than fixation event, for example, when the camera wearer picks up the target object. Smooth pursuit event is necessary for segmenting portion of gaze that follows moving object.

Graph-based candidates clustering
Our proposed method clusters object candidates according to the graph structure. Therefore, the user-defined edge removal threshold highly affects the clustering performance. We believe that current clustering method is too strict. Specifically, it gives too much emphasis on the edge removal threshold. We suggests the study of alternative clustering methods (e.g., random walk), which rely on a probability model instead.

Apart from clustering approaches, we also suggest to use more features in clustering step. In addition to color histogram, we recommend texture features, such as SIFT, PHOW [7], or Gabor feature. We also suggest a study on viewpoint-invariant feature sets, which should be beneficial for our research on multi-view videos.

Group merging step
Group merging step helps recognizing clusters that correspond to the same objects. However, it accumulates and amplifies the error when some of the original clusters contains error. We believe that it is a good idea to introduce a more complicate merging step. For illustration, we can merge two temporally overlapping clusters based on their color models or other features.

Extension to three or more people

Currently, our method detects joint attentions of a pair of people. We believe our method can work with a group of people (3 or more people) as well. However, we speculate that there are more cues we can use when there are many people joining a group activity since we will be able to obtain more videos recorded in many viewpoints. One idea is to incorporate 3D information from 3D reconstruction of the multi-view videos, which is easier to obtain as the number of the videos increases. Another benefits of multiple videos is to detect the objects that are partly obstructed by other objects. For example, we can locate an object that are partly behind another object if the object can be fully seen from other viewpoints. One interesting idea is to construct a map that reflects the relative positions of all objects in the scene by detecting all objects in all viewpoints and relating them together. Then, we can use the map to locate the hiding objects even if they cannot be fully seen in some viewpoints.

Bibliography

[1] Pupil: Open source mobile eye tracking platform. `http://pupil-labs.com/pupil/`. Accessed: 2015-04-18.

[2] Sudipta N. Sinha Adarsh Kowdle and Richard Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *European Conf. on Computer Vision*, October 2012.

[3] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, Nov 2012.

[4] S. Alletto, G. Serra, S. Calderara, F. Solera, and R. Cucchiara. From ego to nos-vision: Detecting social relationships in first-person views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 594–599, June 2014.

[5] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics*, 33(4):81:1–81:11, July 2014.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[7] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Proc. IEEE International Conf. on Computer Vision*, pages 1–8, Oct 2007.

[8] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

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

[10] Xinlei Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2035–2042, June 2014.

[11] Wei-Chen Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 321–328, June 2013.

[12] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Perez. Multi-view object segmentation in space and time. In *Proc. IEEE International Conf. on Computer Vision*, pages 2640–2647, Dec 2013.

[13] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):222–234, Feb 2014.

[14] A. Fathi, J.K. Hodgins, and J.M. Rehg. Social interactions: A first-person perspective. In *Proc. IEEE Conf. on. Computer Vision and Pattern Recognition*, pages 1226–1233, June 2012.

[15] M. Ferecatu and H. Sahbi. Multi-view object matching and tracking using canonical correlation analysis. In *IEEE International Conf. on Image Processing*, pages 2109–2112, Nov 2009.

[16] Huazhu Fu, Dong Xu, Bao Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3166–3173, June 2014.

[17] M. Grundmann, V. Kwatra, Mei Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Proc. IEEE Conf. on. Computer Vision and Pattern Recognition*, pages 2141–2148, June 2010.

[18] Xin Guo, Dong Liu, B. Jou, Mojun Zhu, A. Cai, and Shih-Fu Chang. Robust object co-detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3206–3213, June 2013.

[19] Zeeshan Hayder, Mathieu Salzmann, and Xuming He. Object co-detection via efficient inference in a fully-connected crf. In *Proc. European Conf. on Computer Vision*, pages 330–345, 2014.

[20] Y. Hoshen, G. Ben-Artzi, and S. Peleg. Wisdom of the crowd in egocentric video curation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 587–593, June 2014.

[21] Zejian Yuan Tie Liu Huaizu Jiang, Jingdong Wang and Nanning Zheng. Automatic salient object segmentation based on context and shape prior. In *Proc. of the British Machine Vision Conference*, pages 110.1–110.12, 2011.

[22] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 542–549, June 2012.

[23] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *Proc. European Conf. on Computer Vision*, pages 253–268, 2014.

[24] Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane, and Jeff Pelz. Ego-motion compensation improves fixation detection in wearable eye tracking. In *Proc. of the Symposium on Eye Tracking Research and Applications*, pages 221–224, 2012.

[25] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algorithm. In *Proc. IEEE International Conf. on Computer Vision*, pages 2536–2543, Dec 2013.

[26] Marcus Nyström and Kenneth Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1):188–204, 2010.

[27] Hyun Soo Park, E. Jain, and Y. Sheikh. Predicting primary gaze behavior using social saliency fields. In *Proc. IEEE International Conf. on Computer Vision*, pages 3503–3510, Dec 2013.

[28] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 993–1000, June 2006.

[29] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.

[30] M. Rubinstein, A. Joulin, J. Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1939–1946, June 2013.

[31] J.C. Rubio, J. Serrat, and A. Lopez. Video co-segmentation. In *Proc. Asian Conference on Computer Vision*, pages 13–24, 2013.

[32] Yonetani Ryo, Kawashima Hiroaki, and Matsuyama Takashi. Modeling spatiotemporal correlations between video saliency and gaze dynamics. *IPSJ SIG Notes. CVIM*, 2014(32):1–16, May 2014.

[33] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proc. of the Eye Tracking Research and Applications Symposium*, pages 71–78, 2000.

[34] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proc. European Conf. on Computer Vision-Part I*, pages 414–431, 2002.

[35] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2015.

[36] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, and Nanning Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *Proc. European Conf. on Computer Vision*, 2014.

[37] Le Wang, Gang Hua, Jianru Xue, Zhanning Gao, and Nanning Zheng. Joint segmentation and recognition of categorized objects from noisy web image collection. *IEEE Transactions on Image Processing*, 23(9):4070–4086, Sept 2014.

[38] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *Proc. European Conf. on Computer Vision*, pages 626–639, 2012.

[39] Yiliang Xu, Dezhen Song, and Anthony Hoogs. An efficient online hierarchical supervoxel segmentation algorithm for time-critical applications. In *Proc. British Machine Vision Conference*, 2014.

[40] Dong Zhang, Omar Javed, and Mubarak Shah. Video object co-segmentation by regulated maximum weight cliques. In *Proc. European Conf. on Computer Vision*, pages 551–566, 2014.

[41] Qian Zhang and King Ngi Ngan. Multi-view video based multiple objects segmentation using graph cut and spatiotemporal projections. *J. Vis. Comun. Image Represent.*, 21(5-6):453–461, July 2010.

List of Publications

[1] Nattawan Tantirujananont, Ryo Yonetani and Yoichi Sato, "Finding Objects of Common Interests from Multiple First-Person Videos", in *Meeting on Image Recognition and Understanding (MIRU 2015)*, July 2015.