Relating First-person and Third-person Vision

2018

Shervin Ardeshir Behrostaghi
University of Central Florida

RELATING FIRST-PERSON AND THIRD-PERSON VISION

by

SHERVIN ARDESHIR
M.Sc. University of Central Florida, 2016
B.Sc. Sharif University of Technology, 2012

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2018

Major Professor: Mubarak Shah and Ali Borji

ABSTRACT

Thanks to the availability and increasing popularity of wearable devices such as GoPro cameras, smart phones and glasses, we have access to a plethora of videos captured from the first person (egocentric) perspective. Capturing the world from the perspective of one's self, egocentric videos bear characteristics distinct from the more traditional third-person (exocentric) videos.

In many computer vision tasks (e.g. identification, action recognition, face recognition, pose estimation, etc.), the human actors are the main focus. Hence, detecting, localizing, and recognizing the human actor is often incorporated as a vital component. In an egocentric video however, the person behind the camera is often the person of interest. This would change the nature of the task at hand, given that the camera holder is usually not visible in the content of his/her egocentric video. In other words, our knowledge about the visual appearance, pose, etc. on the egocentric camera holder is very limited, suggesting reliance on other cues in first person videos.

First and third person videos have been separately studied in the past in the computer vision community. However, the relationship between first and third person vision has yet to be fully explored. Relating these two views systematically could potentially benefit many computer vision tasks and applications. This thesis studies this relationship in several aspects. We explore supervised and unsupervised approaches for relating these two views seeking different objectives such as identification, temporal alignment, and action classification. We believe that this exploration could lead to a better understanding the relationship of these two drastically different sources of information.

EXTENDED ABSTRACT

Relating first and third person videos in terms of tasks such as identification and action classification could be very challenging due to the fundamental differences between these two domains. In a third person video, our knowledge on the action being performed, and the person performing it is mostly based on our understanding on the information within the foreground of the video. The pose and patterns in the motion of the actor lead us to reason about the actors identity, and the action being performed. In an egocentric video however, we see the world from the actor's perspective. Thus, the foreground of the video often does not lead to the same information. In fact, the main cue is the change of (global) background motion patterns, hinting toward the action being performed and potentially to the identity of the camera holder. Having a third-person video containing an egocentric camera-holder, we know that not everything visible in the exocentric video will be visible to the egocentric camera holder. Similarly, not everything visible to the egocentric camera holder is necessarily visible in the exocentric video. In other words, comparing and relating the contents of an egocentric and an exocentric view of an event is non-trivial.

In chapter 3, we make the first attempt in this direction by performing identification across egocentric and top-view videos. We address this by answering two main questions, given a set of egocentric videos and a top-view video. First, does the top-view video contain all or some of the egocentric viewers? In other words, have these videos been shot in the same environment at the same time? Second, if so, can we identify each individual egocentric viewer in the top-view video? We answer these questions by modeling each view (egocentric and top) with a graph, and using spectral graph matching techniques in order to attain correspondences across the two views. Even though we obtain promising assignment results in many cases, we observe that these problems can become extremely challenging when videos are not temporally aligned. Hence, in chapter 4 we propose two iterative approaches taking into account the time-delays in the spectral graph matching formulation. We evaluate our methods in terms of ranking the egocentric viewers, and assigning them to identities present in the top-view videos over a dataset of 50 top-view and 188 egocentric videos captured under different conditions.

In chapter 5 we address a shortcoming in the identification task mentioned before. We show that our previous methods are highly dependent on the completeness of the egocentric set and thus, are not suitable for scenarios in which there is only one egocentric and one top-view video available. Given such scenario, we : a) identify the egocentric camera holder in the top-view video (self-identification), b) identify the humans visible in the content of the egocentric video in the top-view video (re-identification), and c) temporally align the two videos. We show that each of these tasks is highly dependent on the other two. Thus, we propose a unified framework to simultaneously address all three problems.

Another task explored in this thesis is relating action information across first-person (egocentric) and third-person (exocentric) views. In chapter 6, we investigate two different, yet highly interconnected problems including cross-view action classification and action based video retrieval. We perform action classification in one domain using the knowledge transferred from the other domain. Also, given a video in one view, we retrieve videos from the same action class in the other view. In order to evaluate our models, we collect a new cross-domain dataset of egocentric-exocentric action videos containing 14 action classes and 3569 videos (1676 collected egocentric videos and 1893 exocentric videos borrowed from UCF 101[1]). Our results demonstrate the possibility of transferring action information across the two domains and performing action based matching and retrieval in seen and unseen classes.

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Mubarak Shah and Dr. Ali Borji, for their great support and guidance throughout these years. Also, I would like to thank committee members, Dr. Haiyan Hu, and Dr. George Atia, for accepting to be a part of my committee, and their helpful guidance throughout the process of proposing and defending my dissertation. I'd like to thank my mother Fattaneh Soltani, my father Mohammad Ardeshir, and my sister Azadeh Ardeshir for their endless support and ecouragement. Also, I would like to thank all of the past and present members of the Center for Research in Computer Vision (CRCV), Tonya LaPrarie, Brittany Kaval, Dr. Amir Roshan Zamir, Dr. Afshin Dehghan, Dr. Omar Oreifej, Dr. Subhabrata Bhattacharya, Dr. Enrique Ortiz, Dr. Haroon Idrees, Dr. Nasim Souly, Dr. Gonzalo Vaca, Dr. Berkan Solmaz, Dr. Khurram Soomro, Dr. Sarfaraz Hussein, Shayan Modiri, Mahdi Kalayeh, Amir Mazaheri, Aidean Sharghi, Alejandro Torroella, Kofi-Malcolm Collins-Sibley, Aisha Orooj Khan, Krishna Regmi, Mohhamed Elfiki, Sandesh Sharma, Fawad Ahmed Keyani for their support, the good times, and the great memories.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The widespread use of wearable devices such as GoPro cameras, smart glasses, and cellphones has created the opportunity to collect first person (egocentric) videos easily and in large scale. People tend to collect large amounts of visual data using their cell phones and wearable devices from the first person perspective. Analysis of these videos has become an interesting and rapidly-growing research area in computer vision, from detecting and recognizing actions (e.g., [3, 4]) to localizing the field of view of an egocentric viewer (e.g., [5]).

From a computer vision standpoint, there is a drastic difference between first-person videos and more traditional third-person videos. Third person videos contain mostly static backgrounds with foregrounds containing the actor(s) and other objects. Motion and pose of the actor visible in the video are discriminative features for identifying the actor and categorizing his/her action Also a person’s visual appearance in a third person video is the main cue for recognizing him/her. In first person videos however, the actor is often the camera holder, and is usually not visible in the content of the video. Hence, the traditional analysis done on third person videos, such as human pose estimation, foreground segmentation and local foreground motion, are not directly applicable. Furthermore, egocentric videos contain a large amount of global motion that differ based on the type of action being performed by the camera holder.

The problem of relating first and third person perspectives, although challenging from a computer vision standpoint, is very intuitive to humans. Watching a human run in an environment, we can easily imagine how the visual world would look like from the actors perspective. This could be related to the mirror neuron phenomena [6]. According to Wikipedia ”A mirror neuron, or cubelli neuron, is a neuron that fires both when an animal acts and when the animal observes the same action performed by another. Thus, the neuron mirrors the behavior of the other, as though the observer were itself acting. Such neurons have been directly observed in primate species."

Even though a lot of research has been done studying the first-person and third-person domains independently, relating the two views systematically has yet to be fully explored. From a computer vision perspective, establishing correspondence between these two views and successful transfer learning across the two views can be very beneficial. Given that the history of third person vision is much longer than first person vision, there are more frameworks designed for solving computer vision tasks in the third-person domain. Also, there are more large scale benchmarks for different tasks, providing the means to evaluate the designed methods more effectively. Successful knowledge transfer from third person to first person vision, leads to applicability of third person methods and data to the first person domain.

In this thesis we aim to explore relating these two drastically distinct views in different aspects, namely, identification, temporal alignment, and action classification. In the following, we provide a more detailed description of each task.

1.1 Matching Viewers in Egocentric and Top-view Videos

Surveillance cameras and unmanned aerial vehicles capture a lot of visual information about daily activities and events taking place in different locations over long periods of time. Top-view vision has a long history in the computer vision research, from human detection and re-identification (e.g., [7, 8, 9]) to object tracking (e.g., [10]). Here we take the first step towards relating the egocentric and top-view vision, which is to establish correspondences between them. To take the first step in this direction, we consider a specific scenario which is localizing and identifying people recording the egocentric videos in a top-view reference video, as illustrated in Fig1.1.

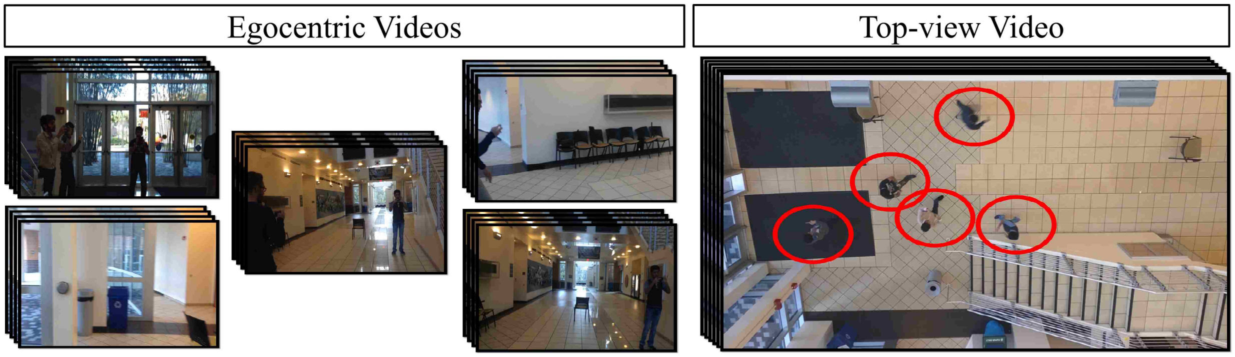


Figure 1.1: Left) a set of 5 egocentric videos. Right) A top-view video capturing the scene. The viewers are highlighted using red circles in the top-view video. We aim to answer the two following questions: 1) Does this set of egocentric videos belong to the viewers visible in the top-view video? 2) Assuming they do, which viewer is capturing which egocentric video?

Given a set of egocentric videos and a top-view video, we ask the following two questions: 1) Does this set of egocentric videos belong to the viewers visible in the top-view video? and 2) If yes, which viewer is capturing which egocentric video? To answer these questions, we need to compare a set of egocentric videos to a set of viewers visible in a single top-view video. To find a matching, each set is represented by a graph and the two graphs are compared using a spectral graph matching technique [11]. In the egocentric graph, each node is an egocentric video. In the top-view graph, each node corresponds to a visible viewer. In general, this problem can be very challenging due to the nature of egocentric cameras. The camera-holder is not visible in his own egocentric video leaving us with no cues regarding his visual appearance.

Evaluating our method over a dataset of 50 top-view and 188 egocentric videos taken in different scenarios demonstrates the efficiency of the proposed approach in assigning egocentric viewers to identities present in top-view camera. We also study the effect of different parameters such as the number of egocentric viewers and visual features. The collected dataset contains several test sets. In each set, multiple people, hereafter referred to as ego-centric *viewers*, are walking around while recording videos. Simultaneously, a top-view camera is recording the entire arena including all or some of the egocentric viewers and possibly some intruders (See Figure 1.1).

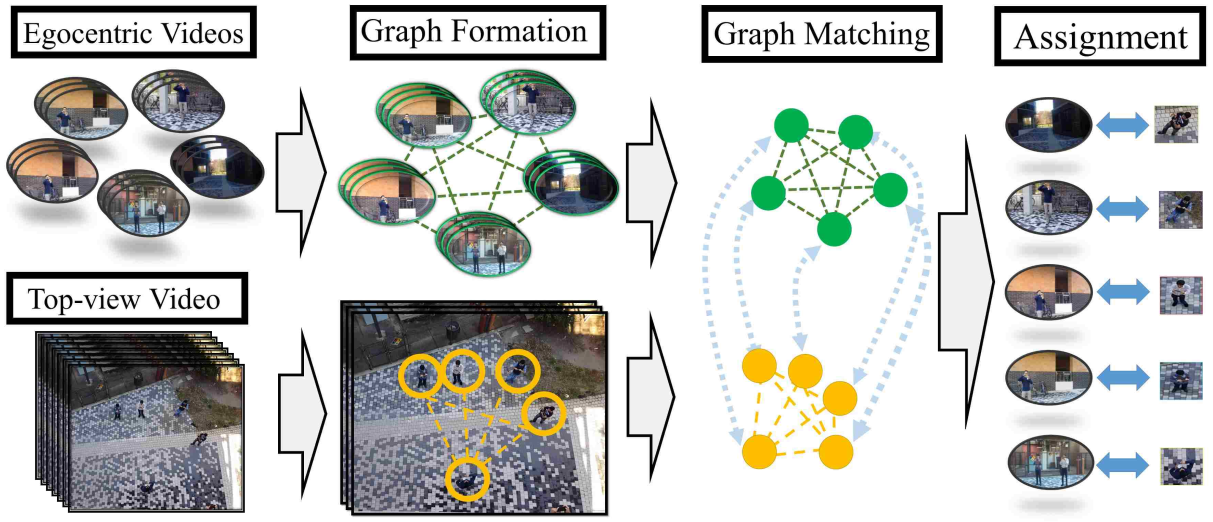


Figure 1.2: The input to our framework is a set of egocentric videos (in this case 5 videos), and one top-view video. The goal is to assign the egocentric videos to the people recording them. A graph is formed on the set of egocentric videos (each node being one of the egocentric videos), and another graph is formed on the top-view video (each node representing one of the targets present in the video). Using spectral graph matching, a soft assignment is found between the two graphs, and using a soft-to-hard assignment, each egocentric video is assigned to one of the viewers in the top-view video. This assignment addresses the second question in Figure 1.1.

In order to have an understanding of the behavior of each individual in the top-view video, we use the multiple object tracking method in [12] to extract viewers trajectories in the top-view video. Note that an egocentric video captures a person's field of view rather than his spatial location. Therefore, the content of a viewer's egocentric video, a 2D scene, corresponds to the content of the viewer's field of view in the top-view camera. For the sake of brevity, in what follows we refer to a viewer's top-view field of view as Top-FOV. Since trajectories computed by multiple object tracking do not provide us with the orientation of the egocentric cameras in the top-view video, we assume that for the most part humans tend to look straight ahead (i.e., front-looking head and torso) and therefore shoot videos from the world in front of them. This assumption often holds when viewers wear the camera on their body (Please see Figure 3.1). Having an estimate of a viewer's orientation and Top-FOV, we then encode the changes in his Top-FOV over time and use it as a descriptor. We show that this feature correlates with the change in the global visual content of the scene observed in his corresponding egocentric video.

We also define pairwise features to capture the relationship between a pair of egocentric videos, and similarly the relationship between a pair of viewers in the top-view camera. Intuitively, if an egocentric viewer observes a certain scene and another egocentric viewer comes across the same scene some time later, this could hint as a relationship between the two cameras. If we match a top-view viewer to one of the two egocentric videos, we are likely to be able to find the other viewer using the mentioned relationship. As we experimentally show, this pairwise relationship significantly improves the assignment accuracy. This assignment leads us to define a score for measuring the similarity between the two graphs. Our experiments demonstrate that the graph matching score could be used for verifying if the top-view video is in fact capturing the egocentric viewers (See the diagram shown in Figure 3.5). Details of this effort is included in chapter 3.

1.2 Simultaneous Identification and Temporal Alignment

A shortcoming of the method mentioned above, is that it does not systematically take the temporal correspondence between the videos into account. Here, we propose two different iterative algorithms to enforce consistent time-delays among the videos. In both methods we estimate the assignment alongside with the time-delays. In the first method, we initially estimate the time delays based on spectrally reasoning about the affinity matrix and seeking a more favorable state of the affinity matrix, and then proceed with the assignment. In the second method we iteratively go back and forth between the output of the hard-assignment, and time-delay estimation. We evaluate and compare the proposed methods on both tasks. We also evaluate the proposed methods in terms of jointly addressing the two problems, temporal alignment, and effect of video length in the assignment accuracy. We also evaluate the run-time and scalability both analytically and practically.

Applications: Establishing reliable correspondences, and temporal alignment between egocentric and top-view videos can have several important real world applications.

1. Sport analysis: Videos of athletes equipped with body-worn cameras alongside with videos captured by static top-view cameras can offer additional insights for sport analysis, which might not be available from each individual source analyzed separately. Having access to the egocentric videos captured by athletes in a group sport activity (e.g. a soccer match), and top and oblique view broadcasting cameras, our method allows finding the correct sport event containing those players in a dataset of soccer matches (by answering the first question). In the same context, knowing that these egocentric videos have been captured by soccer players in a certain soccer match, we can identify them from their egocentric videos i.e., determine who has recorded each egocentric video (question 2).

2. Security and surveillance: Another possible application of this study can be in law enforcement. Securing crowded events such as parades, riots, concerts, etc has always been a challenging task. Plenty of data can be collected by users participating in such events, and also by drones and surveillance cameras in oblique or top-down views. Augmenting the data collected by users with overhead data systematically and establishing some sort of spatial and temporal correspondence between the two sources is very beneficial for tasks such as identification, tracking, discovering suspicious activities. During the past few years, there have been instances of crimes committed by people who were recording the event using first person videos, or have been recorded by other people. Given the known issue of uncertainty with GPS tags (up to several meters) and their unavailability in indoor environments, localization within reference surveillance cameras could potentially be a localization alternative. Also, given the prevalence of the use of body worn cameras by officers, there could be multiple officers equipped with wearable cameras, we might be looking for surveillance cameras containing that group (question 1). In the same setup, we could identify each officer in the surveillance video(question 2). Once this correspondence is discovered, the information from surveillance and egocentric cameras could be augmented more efficiently. We present the details of this effort in chapter 4.

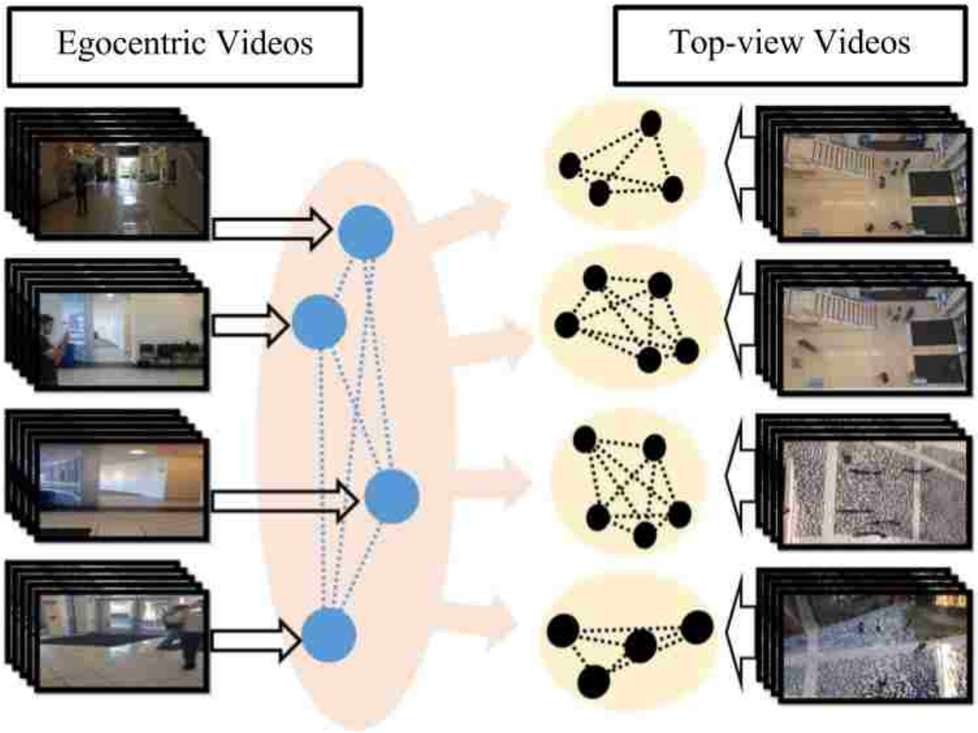


Figure 1.3: Adapting our method for evaluating top-view videos. We form a graph on the set of egocentric videos and compare this graph to other graphs built on different top-view videos. The top-view videos are ranked based on how similar their graph is to the egocentric graph. The performance of this ranking helps us answer our first question.

1.3 Simultaneous Self-identification, Re-identification and Temporal Alignment

In this chapter, we relate ego- and top-view videos from a surveillance standpoint, and address the shortcomings of the earlier methods in a more realistic and challenging scenario. More specifically, given only one egocentric and one top-view video, we address the three following problems:

Self-identification: The goal here is to identify the camera holder of an egocentric video in the reference top-view video (example in Fig. 5.1).

Human re-identification: The goal here is to identify the humans seen in one video (here an egocentric video) in another reference video (here a top-view video). This problem has been studied extensively in the past. It is considered a challenging problem due to variability in lighting, view-point, and occlusion. Yet, existing approaches assume a high structural similarity between captured frames by the two cameras, as they usually capture humans from oblique or side views. This allows a rough spatial reasoning regarding parts (e.g., relating locations of head, torso and legs in the bounding boxes). In contrast, when performing human re-identification across egocentric and top-view videos, such reasoning is not possible (examples are shown in Figs. 5.1 and 1.5).

Temporal alignment: Performing temporal alignment between the two videos directly is non-trivial as the top-view video contains a lot of content that is not visible in the egocentric video. We leverage the other two tasks (self identification and re-identification) to reason about temporal alignment and estimate the time-delay between them.

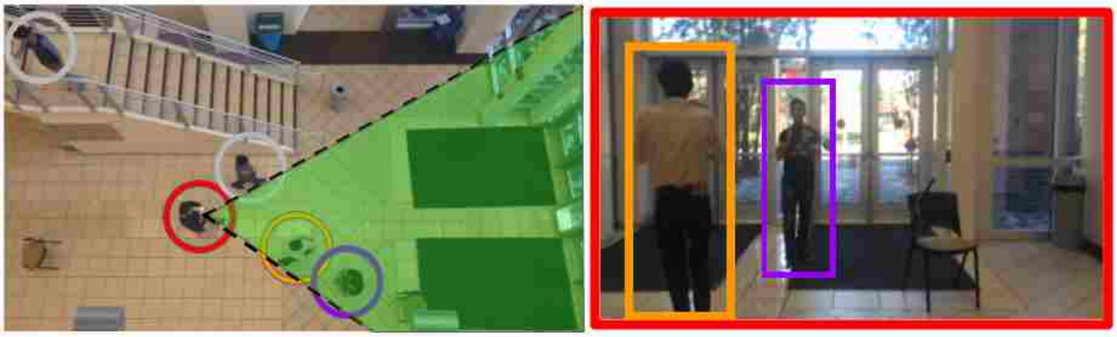


Figure 1.4: A pair of top- (left) and egocentric (right) views. Self identification is to identify the egocentric camera holder (shown in red). Human re-identification is to identify people visible in the egocentric video, in the content of the top-view video (orange and purple).

The interdependency of the three tasks mentioned above encourages designing a unified framework to address all simultaneously. To be able to determine the camera holder's identity within the content of the top-view video (task 1), it is necessary to know the temporal correspondence between the two videos (task 3). Identifying the people visible in the egocentric video in the content of the top-view video (task 2), would be easier if we already knew where the camera holder is in the top-view video at the corresponding time (tasks 1 and 3), since we can reason about who the camera holder is expected to see at any given moment. Further, knowing the correspondence between the people in ego and top views, and temporal alignment between two videos (tasks 2 and 3), could hint towards the identity of the camera holder (task 1). Finally, knowing who the camera holder is (task 1) and who he is seeing at each moment (task 2) can be an important cue to perform temporal alignment (task 3). The chicken-and-egg nature of these problems, encourage us to address them jointly. Thus, we formulate the problem as jointly minimizing the total cost $C_{tot}(l_s, L_r, \tau)$, where l_s is the identity of the camera holder (task 1), L_r is the set of identities of people visible in the egocentric video (task 2), and τ is the time offset between the two videos (task 3). Details of this effort is included in chapter 5.

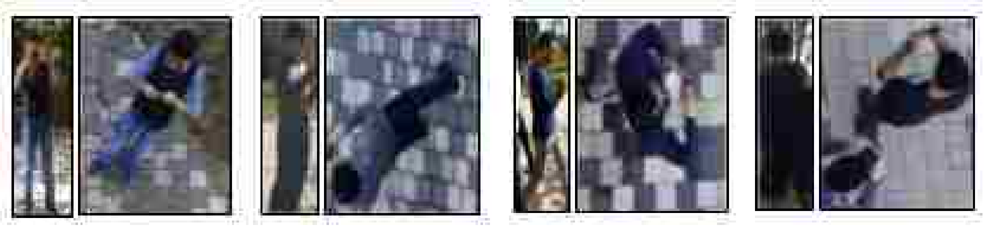


Figure 1.5: Sample ego- and top-view bounding boxes. Unlike conventional re-identification instances, rough spatial alignment assumptions do not hold.

1.4 Relating Exocentric and Egocentric Actions

Here we explore the relationship between egocentric and exocentric videos in terms of transferring action information. Given videos of egocentric and exocentric actions, we explore the possibility of performing action classification, matching and retrieval using the knowledge transferred across the two domains.

In order to evaluate the capability of the proposed framework, we collect a dataset of egocentric and exocentric actions. Each action class is categorized as one of the following depending on the model's exposure to egocentric/exocentric videos of that class during training (as depicted in Figure 1.6).

- *Ego-seen Exo-seen:* Our network will be exposed to instances of those classes in both egocentric and exocentric views during training (e.g., *Bowling* and *playing violin* in Figure 1.6),
- *Ego-seen Exo-unseen & Ego-unseen Exo-seen:* Our network will be exposed only to one of the two views (e.g., *Horse riding* for the former, and Tennis for the latter),
- *Ego-unseen Exo-unseen:* For these action classes, no example from any of the views will be available during training.

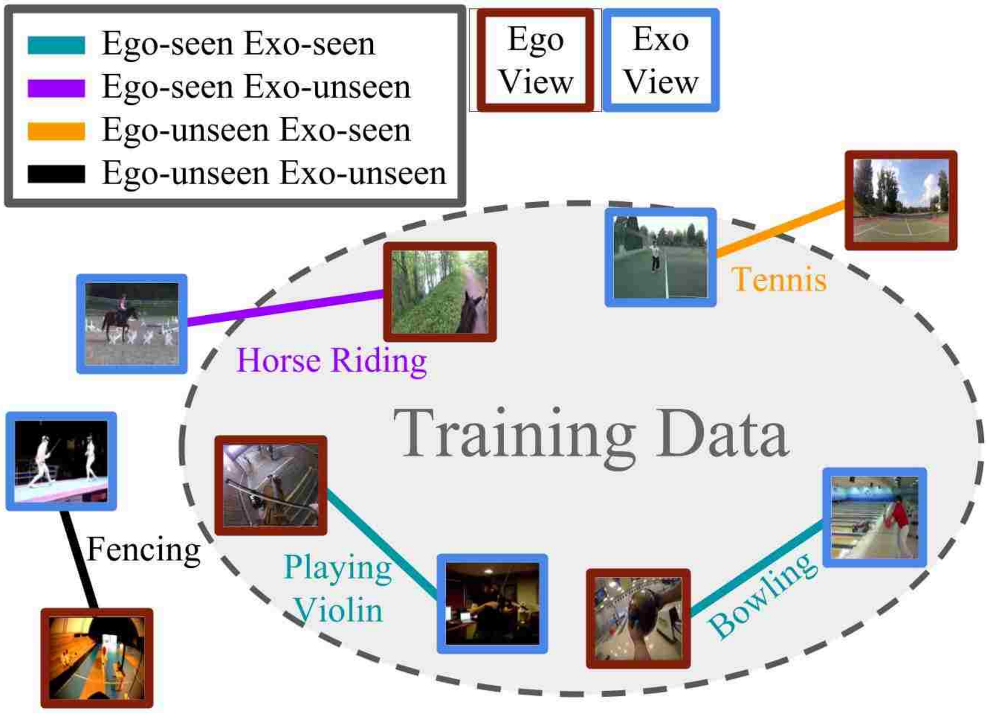


Figure 1.6: Our experimental setup. Four scenarios are considered for each action class. As an example, the class *Horse Riding* can be an ego-seen exo-unseen class. Therefore, the egocentric videos of class *Horse Riding* are split up ($80\%, 10\%, 10\%$ ratio) and used for training, validation, and testing, respectively. However, the network is not exposed to exocentric videos of this class during training, and all of the exocentric videos of this class are used for testing. During training, data from all scenarios are used (except from ego-unseen exo-unseen) to train the model, while during testing we only assess the model using data from only one scenario (all four cases; one at a time). Notice that in X-seen Y-unseen, only X part is used to train the model.

We evaluate the capability of our network in terms of performing action recognition in each view using the transferred knowledge from the other view. Also, we perform matching and retrieval by learning a similarity metric for each egocentric-exocentric video pair based on their actions. In other words, we aim to verify if the egocentric camera holder performs the same action as the human(s) visible in the exocentric video (i.e., action matching). Also given a video from one domain, we aim to retrieve its most similar videos, in terms of action content, in the other domain (i.e., action based retrieval). Details of this effort is included in chapter 6.

CHAPTER 2: LITERATURE REVIEW

Visual analysis of egocentric videos has recently become a hot research topic in computer vision [13, 14], from recognizing daily activities [4, 3] to object detection [15], video summarization [16], and predicting gaze behavior [17, 18, 19]. In the following, we review some earlier work related to ours spanning *Relating exocentric and egocentric videos*,*self-identification and re-identification*, *social interactions among egocentric viewers*, *action recognition*, as well as *domain adaptation and knowledge transfer*.

2.1 Relating Exocentric and Egocentric Videos:

Studying the relationships between moving and static cameras, is a fairly new research topic. As one of the earliest attempts, we [2] perform human identification across egocentric and top-view videos in an unsupervised manner using graph matching. Fan *et al.*, [20] identifies the egocentric camera holder in a simultaneously recorded third person video using a two stream neural network. Mobile and static videos are combined in [21, 22] for the purpose of improving object detection performance. [23] combines information from first and third person cameras with laser range data to improve depth perception and 3D reconstruction. Park *et al.* [24] predict gaze behavior in social scenes using first and third-person cameras. Soran *et al.*, [25] perform action recognition using one egocentric and multiple static videos from the same actor, by combining the information acquired from different views.

2.2 Self Identification

and self localization of egocentric camera holder have been studied during the last few years. Poleg *et al.* [26] use the head motion of an egocentric viewer as a biometric signature to determine which videos have been captured by the same person. In [27], egocentric observers are identified in other egocentric videos by correlating their head motion with the ego-motion of the query video. Authors in [5] localize the field of view of egocentric videos by matching them against Google street view. Landmarks and map symbols have been used in [28] to perform self localization on a map. Yonetani *et al.* [29] propose a self search framework for recognizing egocentric viewers in other egocentric videos.

2.3 Human Re-identification

This problem has been studied heavily in the past (e.g., [30, 31, 9, 32, 33, 33, 34, 35]). Deep learning methods have recently been applied to person re-identification [36, 37, 38]. Yi [39] uses a Siamese network for learning appearance similarities. Similarly Ahmed [40] uses a two stream deep neural network to determine visual similarity between two bounding boxes. Cheng *et al.* [41] uses a multi-channel CNN in a metric learning based approach. Cho *et al.* [42] proposes using pose priors to perform comparison between different candidates, and Matsukawa *et al.* [43] uses a region descriptor based on hierarchical Gaussian distribution of pixel features for this task. In the egocentric domain, the study reported in [44] performs person re-identification in a network of wearable devices, and [45] addresses re-identification across time-synchronized wearable cameras. To the best of our knowledge, our work in Chapter 5 is the first attempt in addressing this problem across egocentric and top-view domains. Visual appearance is often the main cue for human re-identification. This cue can change from one camera to another due to occlusion, viewpoint, and lighting. However, the variation is often relatively low across different static surveillance cameras as the nature of the data is the same (both cameras being ground level or oblique viewpoints). In contrast, in situations where a set of surveillance and egocentric cameras are used, appearance variation is more severe due to egocentric camera motion, more drastic difference in field of views, lighting direction, etc.

2.4 Social Interactions among Egocentric Viewers:

To explore the relationship among multiple egocentric viewers, [46] combines several egocentric videos to achieve a more complete video with less quality degradation by estimating the importance of different scene regions and incorporating the consensus among several egocentric videos. Fathi *et al.*, [47] detect and recognize the type of social interactions such as dialogue, monologue, and discussion by detecting human faces and estimating their body and head orientations. [48] proposes a multi-task clustering framework, which searches for coherent clusters of daily actions using the notion that people tend to perform similar actions in certain environments such as workplace or kitchen. [49] proposes a framework that discovers static and movable objects used by a set of egocentric users. Recent work in [50] identifies the person who draws the most attention in a set of egocentric viewers, given a set of time-synchronized egocentric videos interacting with each other.

2.5 Action Recognition

Action recognition has been a popular area of research and has been studied extensively by the computer vision community[51, 52]. Exocentric actions are mostly recognized based on the human pose[53] and foreground motion[54] extracted from the spatial regions of the video containing the person. Action classification [4, 3, 55, 56] has been one of the hot topics in this area. Egocentric action recognition has also been studied separately. Ogaki et al., [57] use eye and ego motion to estimate global optical flow from first person videos. Singh et al., [58] perform action recognition using hand pose, head motion and low-level saliency. Matsuo et al., [59] propose an attention based approach for activity recognition based on saliency cues. In [60], a method has been proposed to recognize what action is being performed to an egocentric viewer. 3D convolutional neural networks have been used by Poleg et al. [61] for long-term activity recognition and egocentric video segmentation. Kitani et al., [62] propose an unsupervised method for classifying sports actions in egocentric videos and [63] associates functionality to physical spaces and predicts action maps. Activity forecasting has recently become popular. Su et all., [64] predict the next action being performed by the egocentric viewer using 3D reconstruction and social cues. Inverse reinforcement learning has been used in [65] to perform activity forecasting in egocentric videos.

2.6 Knowledge Transfer and Domain Adaptation

The goal here is to transfer information from a source domain to a target domain. There has been extensive research in this area including metric learning approaches [66] and deep learning methods [67, 68]. Aytar et al., [69] use scene labels to perform alignment across different modalities. Eitz et al., [70] perform sketch based image retrieval and [71] performs 3D shape retrieval based on sketches. [72] performs domain adaptation on RGBD data and [73] performs cross-modal domain adaptation using skeleton, and RGB-D data. Knowledge transfer techniques have been adopted for the problem of multi-view action recognition [74, 75, 76] where multiple third person videos of an action are related to each other. This setting allows geometrical and visual reasoning across the views due to two main reasons. First, the nature of the data is the same in different views as they are all exocentric views focusing on the actor. Second, the actor is always visible in the different views, allowing a direct matching across the views. In contrast, in our work the nature of the data in terms of appearance and specifically motion is significantly different in the two views. Further, the egocentric actor is not visible in the egocentric video which prevents doing geometrical reasoning. To the best of our knowledge, our work in chapter 6 is the first attempt in transferring action information across egocentric and exocentric domains, in which we use a two stream network to achieve a view-invariant representation of actions. Two stream networks have been used for tasks such as data fusion between RGB and depth information [77], and person re-identification [78]. Two stream networks have proved to be efficient in extracting view specific features while unifying the global features across different domains. Unsupervised domain adaptation approaches such as geodesic flow [79] subspace alignment[80, 81] have explored the possibility of aligning distributions of unlabeled target data to labeled source data. [82] performs multi-view transfer learning by training a large margin classifier in the source domain and adapting it to the target domain. More recent approaches such as [83, 84] have explored the possibility of unsupervised domain adaptation using adversarial learning. Tzeng et. al [85] use a discriminator do differentiate between features from labeled source and unlabeled target domains, and retrain the target feature extractor to adapt its representation to the source domain.

CHAPTER 3: MATCHING VIEWERS IN EGOCENTRIC AND TOP-VIEW VIDEOS

The results of this work has been published in the following paper:
”Ego2top: Matching viewers in egocentric and top-view videos.” Shervin Ardeshir, and Ali Borji. In European Conference on Computer Vision (ECCV), 2016.

Here we describe the details of our first attempt in relating first-person and third-person videos. The goal is to related egocentric and top-view videos from a surveillance stand-point. More specifically, we aim to identify egocentric viewers in the content of top-view videos. In the following, we describe our proposed framework, the dataset and experiments performed to evaluate the proposed approach, and the achievements and shortcomings of the proposed approach.

3.1 Framework

The block diagram in Figure 1.2 illustrates different steps of our approach. **First**, each view (egocentric or top-down) is represented by a graph which defines the relationship among the viewers present in the scene. These two graphs may not have the same number of nodes for two reasons: a) some of the egocentric videos might not be available, b) some individuals, present in the top-view video, might not be capturing videos. Each graph consists of a set of nodes where each node represents a viewer (egocentric or top-view), and each edge represents a pairwise relationship between two viewers.

We represent each viewer in the top-view by describing his expected Top-FOV, and in egocentric view by the visual content of his video over time. These descriptions are encoded in the graph nodes. We also define pairwise relationships between pairs of viewers, which are encoded as the edge features of the graph (i.e., how two viewers' visual experience relate to each other).

Second, we use spectral graph matching to compute a score measuring the similarity between the two graphs, alongside with an assignment from the nodes of the egocentric graph to the nodes of the top-view graph.

Our experiments show that the graph matching score can be considered as a measure of similarity between the egocentric and the top-view graphs. As a result, it can be used for verifying whether a set of egocentric videos are recorded by the viewers visible in the top-view video (i.e., the capability of our method in terms of answering our first question). In addition, the assignment obtained by the graph matching suggests an identification label for each egocentric viewer in the top-view video (i.e., an answer to our second question).

3.1.1 Graph Representation

Each view, egocentric or top-view, is described using a single graph. The set of egocentric videos is represented using a graph in which each node represents one of the egocentric videos, and an edge captures the pairwise relationship between the content of the two videos.

In the top-view graph, each node represents the expected visual experience of a viewer being tracked (in the top-view video), and an edge captures the pairwise relationship between the two visual experiences over time. *Visual experience* refers to what a viewer is expected to observe during the course of his recording seen from the top-view camera.

3.1.1 Modeling the Top-View Graph: In order to model the visual experience of a viewer in the top-view camera, knowledge about his spatial location (i.e., trajectory) throughout the video is needed. We employ the multiple object tracking method presented in [12] to extract a set of trajectories, each corresponding to one of the viewers in the scene. Similar to [12], we use annotated bounding boxes, and provide their centers as an input to the multiple object tracker. Our tracking results here are nearly perfect due to several reasons: the high quality of videos, high video frame rate, and lack of challenges such as occlusion in the top-view videos.

Each node represents one of the individuals being tracked. Employing the general assumption that people often tend to look straight ahead, we use a person's speed vector as the direction of his camera at time t (denoted as θ_t). Further, assuming a fixed angle (θ_d), we expect the content of the person's egocentric video to be consistent with the content included in a 2D cone formed by the two rays emanating from the viewer's location between angles $\theta - \theta_d$ and $\theta + \theta_d$. Figure 3.1 illustrates the expected Top-FOV for three different individuals present in a frame. In our experiments, we set θ_d to 30 degrees. In theory, angle θ_d can be estimated more accurately by knowing intrinsic camera parameters such as focal length and sensor size of the corresponding egocentric camera. However, since we do not know the corresponding egocentric camera, we set it to a default value. As a post-processing, we perform a Gaussian temporal filter on top of the estimated orientations in order to handle the possible stationary torso rotation moments between two dominant directions of movement.

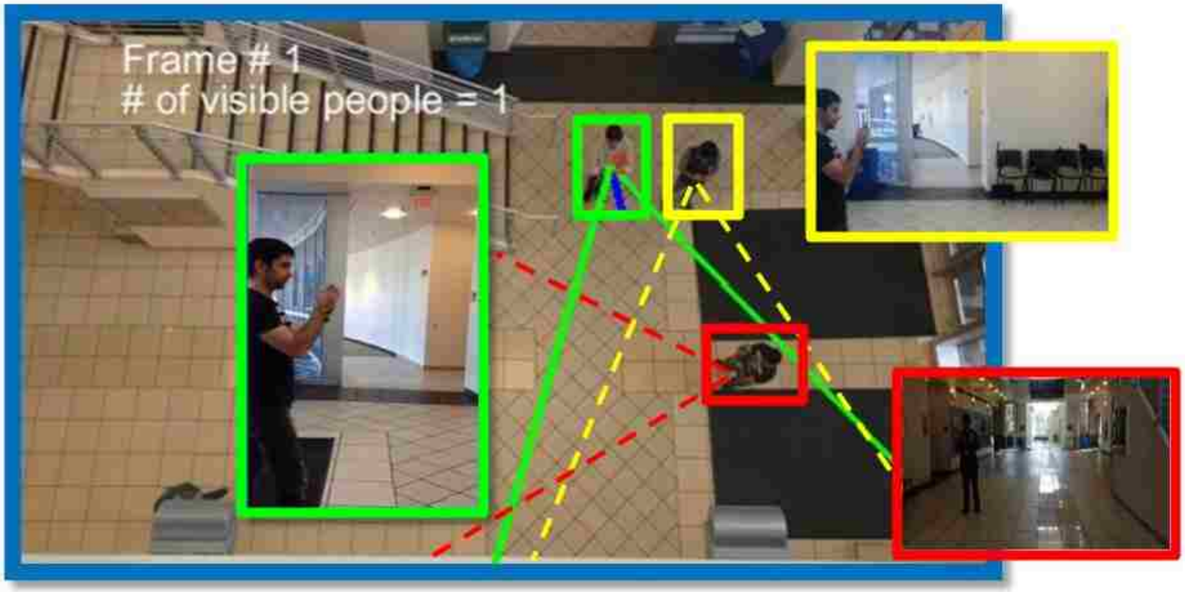


Figure 3.1: Expected FOV for three different viewers in the top-view video alongside with their corresponding egocentric frames. The Top-FOV shown for the identity highlighted in green has a high overlap with the one highlighted in yellow, therefore we expect their egocentric videos (color-coded accordingly) to have relatively similar visual content. In contrast, the FOV of the identity highlighted in red doesn't have as much overlap with other FOVs, thus we don't expect their egocentric videos to look similar visually.

Top-FOVs are not directly comparable to viewers' egocentric views. The area in the Top-FOV in a top-view video mostly contains the ground floor which is not what an ego-centric viewer usually observes in front of him. *However, what can be used to compare the two views is the relative change in the Top-FOV of a viewer over time. This change should correlate with the change in the content of the egocentric video.* Intuitively, if a viewer is looking straight ahead while walking on a straight line, his Top-FOV is not going to have drastic changes. Therefore, we expect the viewer's egocentric view to have a stable visual content with subtle changes.

Node Features: We extract two unary features for each node, one capturing the changes in the content covered by his FOV, and the other one capturing the number of visible people in the content of the Top-FOV.

To encode the relative change in the visual content of viewer i visible in the top-view camera, we form the $T \times T$ matrix (T denotes the number of frames in the top-view video) U_i^{IOU} whose elements $U_i^{IOU}(p,q)$ indicate the IOU (intersection over union) of the Top-FOV of person i in frames p and q. For example, if the viewer's Top-FOV in frame 10 has high overlap with his FOV in frame 30 (thus $U_i^{IOU}(10,30)$ has a high value), we expect to see a high visual similarity between frames 10 and 30 in the egocentric video. Two examples of such features are illustrated in Figure 3.2 (a).

Having the Top-FOV of viewer i estimated, we then count the number of people within his Top-FOV at each time frame and store it in a $1 \times T$ vector U_i^n. To compute the number of visible people, we count the number of annotated bounding boxes within his Top-FOV. Figure 3.1 illustrates three viewers who have one human in their Top-FOV. A few examples of this feature are visualized in Figure 3.2a.

Edge Features: Pairwise features are designed to capture the relationship among two different individuals. In the top-view videos, similar to the unary matrix U_i^{IOU}, we can form a $T \times T$ matrix B_{ij}^{IOU} to describe the relationship between a pair of viewers (viewers/nodes i and j), in which $B_{ij}^{IOU}(p,q)$ is defined as the intersection over union of the Top-FOVs of person i in frame p and person j in frame q. Intuitively, if there is a high similarity between the Top-FOVs of person i in frame p and person j in frame q, we would expect the qth frame of viewer j's egocentric video to be similar to the pth frame of viewer i's egocentric video. Two examples of such features are illustrated in the middle column of Figure 3.2 (b).

3.1.2 Modeling the Egocentric Graph: As in the top-view graph, we also construct a graph on the set of egocentric videos. Each node of this graph represents one of the egocentric videos. Edges between the nodes capture the relationship between a pair of egocentric videos.

Node Features: Similar to the top-view graph, each node is represented using two features. First, we capture how the overall visual experience is evolving. We compute pairwise similarity between GIST features [86] of all video frames (for one viewer) and store the pairwise similarities in a $T_{E_i} \times T_{E_i}$ matrix $U_{E_i}^{GIST}$, in which the element $U_{E_i}^{GIST}(p,q)$ is the GIST similarity between frames p and q of egocentric video i, and T_{E_i} is the number of frames in the ith egocentric video. Two examples of such features are illustrated in the left column of Figure 3.2 (b). The GIST similarity is a function of the Euclidean distance of the GIST feature vectors.

$$U_{E_i}^{GIST}(p,q) = e^{-\gamma|g_p^{E_i} - g_q^{E_i}|}. \tag{3.1}$$

In which $g_p^{E_i}$ and $g_q^{E_i}$ are the GIST descriptors of frames p and q of egocentric video i, and γ is a constant which we empirically set to 0.5.

The second feature is a time series counting the number of visible people in each frame. In order to have an estimate of the number of people, we run a pre-trained human detector using deformable part model DPM [8] on each egocentric video frame. To make sure that our method is not including humans in far distances (which are not likely to be present in the top-view camera), we exclude bounding boxes whose sizes are smaller than a certain threshold (determined considering an average human height of 1.7m and length of the diagonal of the area being covered in the top view video frame.). Each of the remaining bounding boxes, has a detection score which is rescaled into the interval [0 1]. The rescaled score has the notion of the probability of that bounding box containing a person. Scores of all detections in a frame are added and used as a count of people in that frame. Therefore, similar to the top-view feature, we can represent the node E_i of egocentric video i with a $1 \times T_{E_i}$ vector $U^n_{E_i}$. A few examples of this feature are visualized in Figure 3.3.

Edge Features: To capture the pairwise relationship between egocentric videos i (containing T_{E_i} frames) and j (containing T_{E_j} frames), we extract GIST features from all of the frames of both videos and form a $T_{E_i} \times T_{E_j}$ matrix B^{GIST}_{ij} in which $B^{GIST}_{ij}(p,q)$ represents the GIST similarity between frame p of video i and frame q of video j.

$$B^{GIST}_{ij}(p,q) = e^{-\gamma|g^{E_i}_p - g^{E_j}_q|}. \tag{3.2}$$

Two examples of such features are illustrated in the left column of Figure 3.2 (b).

3.1.2 Graph Matching

Our goal in this section is to find a binary assignment matrix $\mathbf{X}_{N^e \times N^t}$, in which N^e is the number of egocentric videos and N^t is the number of people in the top-view video. $\mathbf{X}(i,j)$ equal to 1 means that egocentric video i has been matched to viewer j in the top-view video. To capture the similarities between the elements of the two graphs, we define the affinity matrix $A_{N^eN^t \times N^eN^t}$ in which $a_{ik,jl}$ is the affinity of edge ij in the egocentric graph with edge kl in the top-view graph. Reshaping matrix $\mathbf{X}$ as a vector $\mathbf{x}_{N^eN^t \times 1} \in \{0,1\}^{N^eN^t}$, the assignment problem could be defined as maximizing the following objective function:

$$\operatorname*{argmax}_{x} \mathbf{x}^T\mathbf{A}\mathbf{x}, \;\; s.t. \;\; \Sigma^{N^e}_{p=1} x_{pq} \leq 1 \; and \; \Sigma^{N^t}_{q=1} x_{pq} \leq 1. \tag{3.3}$$

We compute $A_{ik,jl}$ based on the similarity between the feature descriptor of edge ij in the egocentric graph B_{ij}^{GIST} and the feature descriptor for edge kl in the top-view graph B_{kl}^{IOU}. Once the affinity matrix is known we can measure the probability of each of the nodes in the first graph being matched to each of the nodes in the second graph. This probabilistic assignment is commonly known as soft-assignment.

Soft Assignment We employ the spectral graph matching method introduced in [11] to compute a soft assignment between the set of egocentric viewers and top-view viewers. In [11], assuming that the affinity matrix is an empirical estimation of the pairwise assignment probability, and the assignment probabilities are statistically independent, **A** is represented using its rank one estimation which is computed by:

$$\underset{\mathbf{p}}{\operatorname{argmin}}\, |\mathbf{A} - \mathbf{p}\mathbf{p}^T|. \tag{3.4}$$

In fact, the rank one estimation of **A** is no different than its leading eigenvector. Therefore, **p** can be computed either using eigen decomposition, or estimated iteratively using power iteration. Considering vector **p** as the assignment probabilities, we can reshape $\mathbf{p}_{N^e N^t \times 1}$ into a $N^e \times N^t$ soft assignment matrix **P**, for which $\mathbf{P}(i,j)$ represents the probability of matching egocentric viewer i to viewer j in the top-view video after row normalization.

Hard Assignment Any soft to hard assignment method can be used here to convert the soft assignment result (generated by spectral matching) to the hard binary assignment between the nodes of the graphs, essentially converting soft assignments p into final solutions x in Eq 3.3. We used the well-known Munkres (also known as Hungarian) algorithm [87] to obtain the final binary assignment.

3.1.3 Solving Viewer Assignment

As described in the previous section, each of the nodes and edge features is a 2D matrix. B_{ij}^{GIST} is a $T_{E_i} \times T_{E_j}$ matrix, T_{E_i} and T_{E_j} are the number of frames in egocentric videos i and j, respectively. B_{kl}^{IOU} is a $T_t \times T_t$ matrix where T_t denotes the number of frames in the top-view video. Note that B_{ij}^{GIST} and B_{kl}^{IOU} are not directly comparable as the two matrices are not of the same size (the videos do not necessarily have the same length). Also, the absolute time in the videos do not correspond to each other as the videos are not time-synchronized. In fact, the relationship between viewers i and j in the 100th frame of the top-view video does not correspond to frame number 100 of the egocentric videos. Due to this, we expect to see a correlation between the GIST similarity of frame $100+d_i$ of egocentric video i and frame $100+d_j$ of egocentric video j, and the intersection over union of in Top-FOVs of viewers k and l in frame 100. d_i and d_j are the time delays of egocentric videos i and j with respect to the top-view video, respectively.

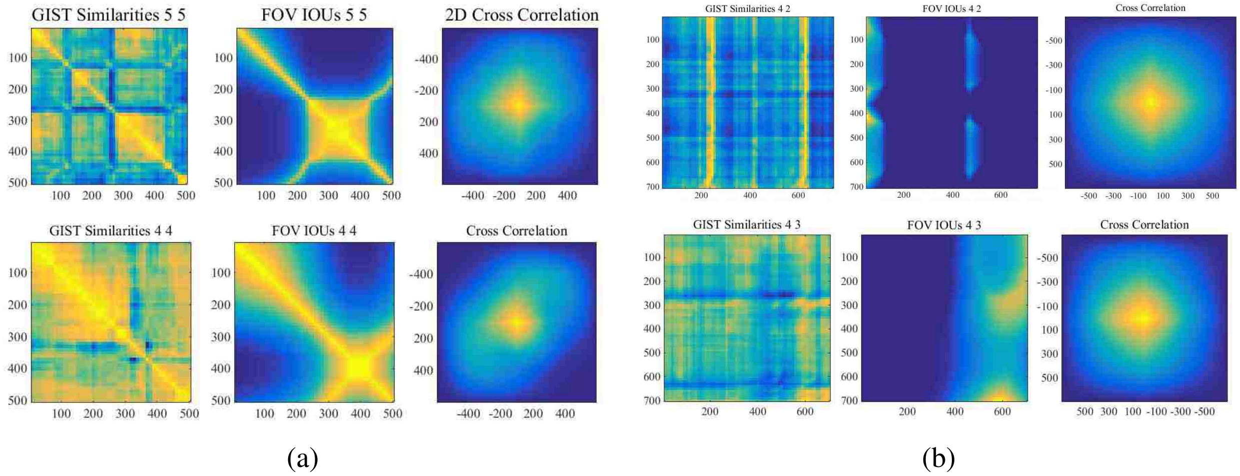
We define the affinity between two edges is defined as the following:

$$A_{ikjl} = \max(B_{ij}^{GIST} * B_{kl}^{IOU}). \tag{3.5}$$

where $*$ denotes cross correlation. For the elements of A for which $i=j$ and $k=l$, the affinity captures the compatibility of node i in the egocentric graph, to node k in the top-view graph. The compatibility between the two nodes is computed using 2D cross correlation between U_k^{IOU} and $U_{E_i}^{GIST}$ and 1D cross correlation between U_k^n and $U_{E_i}^n$. The overall compatibility of the two nodes is a weighted linear combination of the two:

$$A_{ikik} = \alpha \max(U_{E_i}^{GIST} * U_k^{IOU}) + (1-\alpha)\max(U_{E_i}^n * U_k^n), \tag{3.6}$$

where α is a constant between 0 and 1 specifying the contribution of each term. In our experiments, we set α to 0.9. Figure 3.2 illustrates the features extracted from some of the nodes and edges in the two graphs. Where maximum of cross correlation occurs is interpreted as the best offset (or delay) that makes the two matrices the most similar. The time delay problem is handled properly by assuming that each cross-correlation is maximized on an offset equal to the time-delays of its corresponding egocentric videos. This assumption might not always hold as it does not enforce consistency among the assumed time-delays. We will address this issue using the approaches described in the next section.



(a)

(b)

Figure 3.2: (a) shows two different examples of the 2D features extracted from the **nodes** of the graphs for which the values are color-coded. Left column shows the 2D matrices extracted from the pairwise similarities of the GIST feature descriptors U^{GIST}, middle shows the 2D matrices computed by intersection over union of the FOV in the top-view camera U^{IOU}, and the rightmost column shows the result of the 2D cross correlation between the two. (b) shows the same concept, but between two **edges**. The leftmost figure shows the pairwise similarity between GIST descriptors of one egocentric camera to another B^{GIST}. Middle, shows pairwise intersection over unions of the FOVs of the pair of viewers B^{IOU}, and the rightmost column illustrates their 2D cross correlation. The similarities between B^{GIST} and B^{IOU} capture the affinity of two nodes/edges in the two graphs.

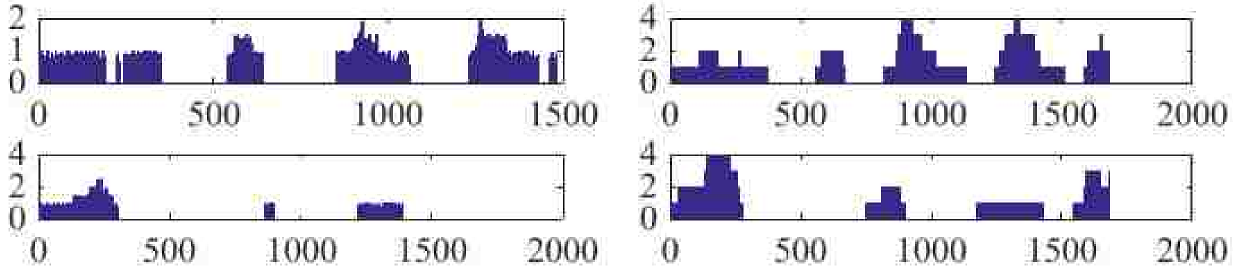


Figure 3.3: Examples of 1D features capturing the number of humans in different frames. Left column shows the summation of the detection scores at every frame for their corresponding egocentric videos. Right column shows the number of visible people in three different viewer's Top-FOV over time. The x axis encodes the frame number in which the number of humans was measured. Both vectors are normalized and then compared to each other. The similarity between the two patterns shows the discriminative power of this feature, especially if the video is long enough. However, our experiments show that in most cases where human detection results are not that confident or the video length is too short, this feature by itself does not results in a high assignment accuracy.

3.2 Experimental Results

In our experiments, we tried to evaluate the performance of our framework, in terms of answering to the two questions we introduced. First, having a top view video and a few egocentric videos, which we know are captured by the people visible in the top view camera, can we say which person has captured which egocentric video? And second, if we have a set of egocentric videos, and a set of top view videos, can we say which top view video, contains the people recording the egocentric videos? In order to verify that, we collected a large dataset and performed extensive experiments to answer these questions.

3.2.1 Dataset

We collected a dataset containing 50 set of videos. Each set of videos, contains one top view video and several egocentric videos captured by the people visible in the top view camera. Depending on the number of egocentric cameras, we can generate up to 2,862 instances of our assignment problem. Each instance containing one top view camera and from 1 to *n* egocentric cameras, where *n* is the number of egocentric cameras in that set. Overall, our dataset contains more than 225,000 frames. Number of people visible in the top view cameras vary from 3 to 10, number of egocentric cameras vary from 1 to 6, and ratio of number of available egocentric cameras to number of visible people in the top view camera varies from 0.16 to 1. Length of the videos also vary from 320 frames (10.6 seconds) up to 3132 frames (110 seconds) with mean of X and standard deviation of dX.

3.2.2 Evaluation

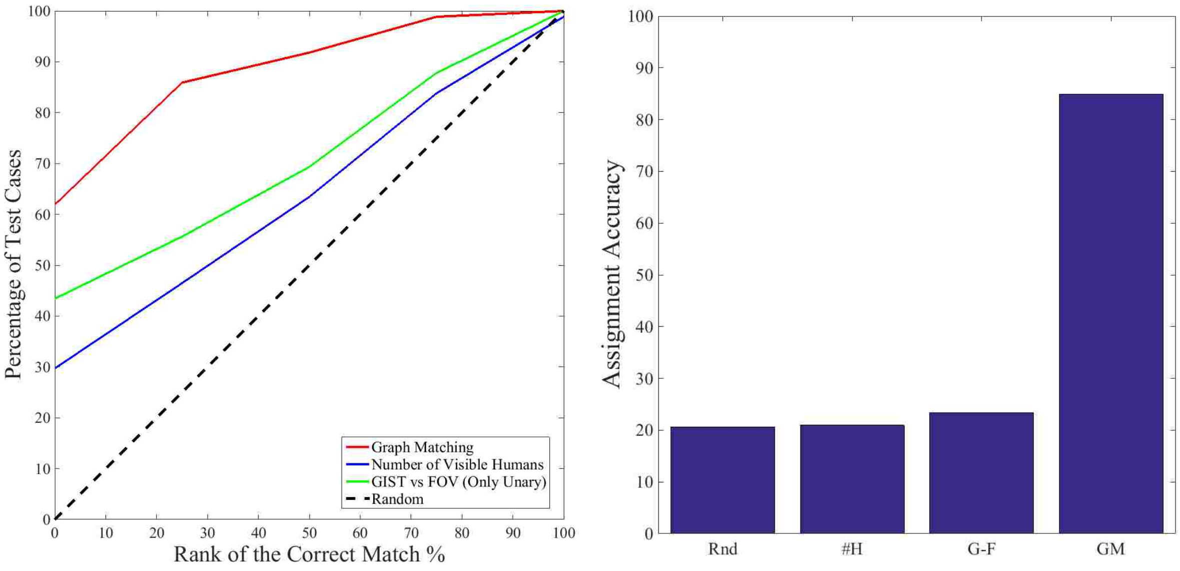
We evaluate our method in terms of it's hard-assignment accuracy and also it's ranking accuracy.

3.2.2.1 Assignment Precision:

We evaluate the accuracy of our method by computing it's assignment accuracy, which in essence captures the percentage of the egocentric cameras which were correctly matched to their corresponding targets. We evaluate the assignment accuracy for our method (graph matching soft assignment + Hungarian) and compare it with three baselines: random assignment and Hungarian only, ignoring the pairwise relationships (edges) in the graphs. The consistent improvement of our method over the baselines, justifies the effectiveness of our representation.

3.2.2.2 Ranking Accuracy:

We evaluate our assignment problem, in terms of ranking. In other words, we can look at our soft assignment as a measure to sort the targets in the top view videos based on their assignment probability to a certain egocentric video. Computing the ranks of the correct matches, we can plot cumulative matching curves (CMC) to illustrate their performance. We also compute the area under curve for having a quantitative measurement of our framework.

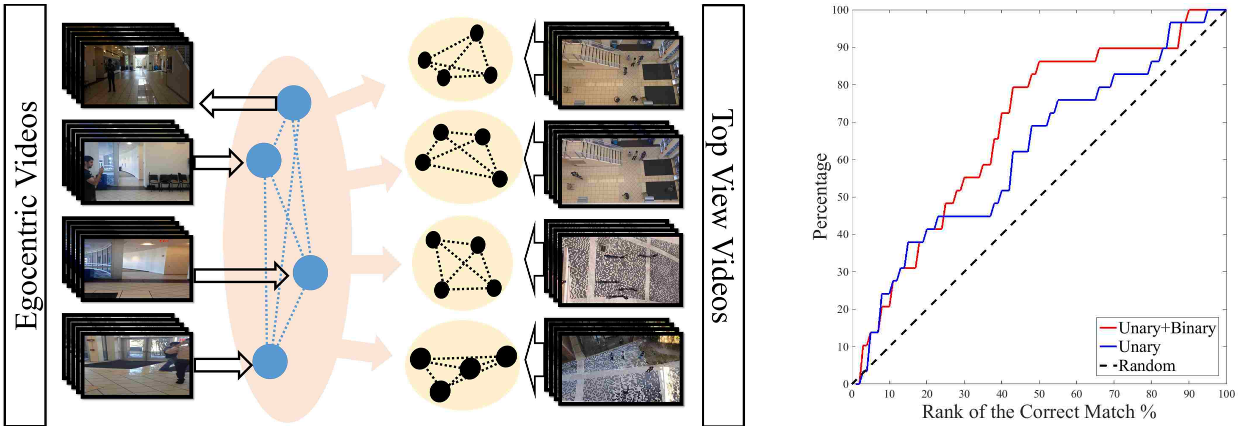


(a) (b)

Figure 3.4: (a) shows the cumulative matching curve for our ranking scheme. The red, green and blue curve belong to ranking based on spectral graph matching scores, cross correlation between the unary scores, and cross correlation between the 1D number of visible humans signals. The dashed clack curve shows random ranking accuracy (b) shows the assignment accuracy based on randomly assigning, using the number of humans, using unary features, and using spectral graph matching.

3.2.2.3 Scene Ranking Accuracy:

Our method is designed with the goal to perform an assignment from the egocentric cameras to the people present in a top view camera, assuming the top down video is capturing the same scene. But what if we do not know which top view video is capturing the scene of egocentric videos? We perform an experiment in which having a set of egocentric videos from the same event, and a set of top view videos, we aim to find the correct top view camera, by comparing the similarity of the graph formed on the egocentric cameras to the graph formed on each top view video. We associate the score $x^T Ax$ to the scene and rank the scores of different scenes based on that score. We evaluate the ranking accuracy by comparing the area under curve of the cumulative matching curve shown in figure.



(a)

(b)

Figure 3.5: (a) illustrates how we adapt our method to rank the top view videos based on the graph similarity score. We form a graph on the set of egocentric videos and compare that graph to all the graphs, each built on one of the top view surveillance videos. Finally, the top view videos are ranked based on their graph similarity score to the egocentric graph. (b) shows the ranking performance with its cumulative matching curve.

CHAPTER 4: SIMULTANEOUS IDENTIFICATION AND TEMPORAL ALIGNMENT

The results of this work has been published in the following paper:
”Egocentric Meets Top-view”, Shervin Ardeshir, Ali Borji, Patterns Analysis and Machine Intelligence (PAMI) 2018.

The drawback of the similarity definition in [2] is that it does not enforce any sort of consistency among the time-delays assigned to different egocentric videos. In fact, the problem of viewer assignment, and time-delays of the egocentric videos are interconnected. On one hand, we need to have an estimation of the time-delays, to be able to correctly measure the node-to-node/edge-to-edge similarities of the corresponding nodes/edges. On the other hand, we need to know the correct assignment to be able to estimate the time-delay between two videos.

Theoretically, if we assume the top-view video as a reference of absolute time (as shown in Figure 4.1), each cross-correlation maximization is suggesting one (for nodes), or two (for edges) egocentric time delays with respect to the top-view video’s absolute time. As an example, if the edge between egocentric videos i and j has its cross correlation with its corresponding top-view edge maximized at d'_i and d'_j, then these values are the time-delays of egocentric videos i, and j. Therefore, if the cross-correlation of edge ik being maximized in the first dimension at d''_i (which $d''_i \neq d'_i$), we are assuming egocentric video i is starting at two different absolute times, which is self-contradictory. Therefore, the framework needs to enforce consistency among the time-delays of all the egocentric videos, suggesting a unique time-delay for each individual egocentric video. As a result, we define the objective to jointly optimize the time-delays and the assignment. Intuitively, putting constraints on the time-delays, will put constraints on the solution space, as some of the solutions using [2] might implicitly assign invalid and inconsistent time-delays to the egocentric videos.

Having n egocentric videos, we can represent their unknown time delays, using a $1 \times n$ vector $\mathbf{t}_d$. Taking the time delays into account, then the objective has the form of:

$$\underset{\mathbf{x},\mathbf{t}_d}{\operatorname{argmax}}\ \mathbf{x}^T \mathbf{A}(\mathbf{t}_d)\mathbf{x}. \tag{4.1}$$

This brings us back to the chicken and egg nature of the problem, which suggests an iterative-alternative approach. We Initialize the time delays, and iteratively estimate the assignments and refine the time-delays based on that. Intuitively, we should seek the optimum assignment, in addition to a time delay for each egocentric video. $\mathbf{A}(\mathbf{t}_d)$ is the affinity matrix, assuming the i_{th} egocentric video has time delay $\mathbf{t}_d(i)$. Changing $\mathbf{t}_d$ will alter the elements of the affinity matrix, and $\mathbf{x}$ will decide which elements of the affinity matrix should contribute to the graph matching score. We employ two iterative-alternative methods for this objective. First, we explore a faster algorithm which first seeks an optimal time delay vector, and then proceeds to the assignment problem. The second algorithm alternates between the assignment and time-delay estimation.

First Approach: Spectral Optimization. In the first approach, we find an optimum affinity matrix $\mathbf{A}$ resulting from the optimal time delays for the egocentric videos, and then solve the assignment using the obtained affinity matrix. In other words we assume:

$$\underset{\mathbf{x},\mathbf{t}_d}{\operatorname{argmax}}\ \mathbf{x}^T \mathbf{A}(\mathbf{t}_d)\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmax}}\ \mathbf{x}^T\ \mathbf{A}(\mathbf{t}_d^*)\ \mathbf{x}. \tag{4.2}$$

Intuitively, as explained in Section 3.1.2, the affinity matrix is estimated using its rank one approximation. This rank one approximation is more accurate when the leading eigenvalue of the affinity has a larger value. Therefore, the higher the leading eigenvalue of the affinity matrix is, the more confident our spectral graph matching will be. According to this intuition, we can find $\mathbf{t}_d^*$ using:

$$\mathbf{t}_d^* = \operatorname*{argmax}_{\mathbf{t}_d} \lambda_{\mathbf{A}(\mathbf{t}_d)}. \tag{4.3}$$

For solving the objective function above, we initialize the time delays and iteratively refine them using a local search in the n dimensional space of the time-delay vector. The details are explained in Algorithm 1. Effectively, first we evaluate neighboring time delay vectors by analyzing their corresponding affinity matrices. Having a n dimensional time-delay vector, we compute its neighboring time-delay vectors along each dimension, by changing one of its elements (time-delay of one of the egocentric videos) by a single unit δ (which we empirically set to 0.1 sec). For each neighbor, we compute the resulting affinity matrix and its leading eigenvalue. We pick the neighboring time delay vector with the maximum leading eigenvalue in the affinity matrix and update time delays and the affinity matrix. The algorithm keeps iterating until one of the convergence criteria is met. We define the convergence criteria as either reaching a local maximum leading eigenvalue, or reaching the maximum number of iterations (which we set to 30). After convergence, soft and hard assignments are computed using the computed optimum affinity matrix. We explore the effect of the two different initializations in terms of assignment and ranking and compare the results with [2].

Second Approach: Matching Score Based Optimization. Here, we attempt to find the optimal values for $\mathbf{t}_d$ and $\mathbf{x}$ simultaneously using an iterative-alternative approach. First, we initialize $\mathbf{t}_d$, and therefore initialize the affinity matrix. Second, we compute the assignments using spectral graph matching. The assignment is then used to further refine the time delays. In other words, we observe how the graph matching score changes, using different neighboring time delay vectors,

Algorithm 1 Spectral Optimization

```
Input = $\mathbf{G}, \mathbf{G}', itr_{max}, \delta$
Output = $\mathbf{x}, \mathbf{t}_d$

procedure SPECTRAL SOLVER
    Initialize $\mathbf{t}_d$
    compute the affinity matrix $\mathbf{A}_{t_d}$
    while $\max_i \lambda_{\mathbf{A}_{\mathbf{t}_d \pm \delta u_i}} > \lambda_{\mathbf{A}_{\mathbf{t}_d}}$ and $itr < itr_{max}$ do
        update $\mathbf{t}_d$ and $\mathbf{A}_{\mathbf{t}_d}$
    compute soft and hard assignment $\mathbf{p}_{t_d}$ and $\mathbf{x}_{t_d}$
    return $\mathbf{t}_d$ and $S(\mathbf{t}_d)$.
```

and pick the best direction for the growth of the graph matching score. Similar to the previous approach, we simply evaluate the neighboring time-delay vectors along each dimension. We go back and forth between the time-delays and assignments until our termination criterion is met. Similar to algorithm 1, the termination criteria are defined as reaching a local maximum or maximum number of iterations. The details of this approach are explained in Algorithm 2. Our experiments show that this method can outperform the first approach (Algorithm 1), with the cost of slightly more computational complexity as each iteration consists of additional steps of computing the assignment vector **x**. The performance of this algorithm will be compared to the first approach in the next section.

Algorithm 2 Matching Score Based Optimization

```
Input = $\mathbf{G}, \mathbf{G}', itr_{max}, \delta$
Output = $\mathbf{x}, \mathbf{t}_d$

procedure MATCHING SCORE BASED SOLVER
    Initialize $\mathbf{t}_d$
    compute the affinity matrix $\mathbf{A}_{t_d}$, soft assignment $\mathbf{p}_{t_d}$, hard assignment $\mathbf{x}_{t_d}$ and graph matching score $S(t_d) = \mathbf{x}_{t_d}^T \mathbf{A}_{t_d} \mathbf{x}_{t_d}$
    while $\max_i S(\mathbf{t}_d \pm \delta u_i) > S(\mathbf{t}_d)$ and $itr < itr_{max}$ do
        update $\mathbf{t}_d$, $\mathbf{A}_{\mathbf{t}_d}$, $\mathbf{p}_{\mathbf{t}_d}$, $\mathbf{x}_{\mathbf{t}_d}$, and $S(\mathbf{t}_d)$
    return $\mathbf{t}_d$ and $S(\mathbf{t}_d)$.
```

Initializing time-delays: Since we locally search for the best objective, the initialization plays a significant role in the final results. Two different initialization methods are considered. First, we initialize the vector $\mathbf{t}_d$ with a vector of zeros, assuming the videos are time-synchronized. Second, we empirically estimate the time-delays by computing the median of all the values suggested by the cross-correlations. As explained in [2], each cross-correlation maximization suggests a time-delay for each of the egocentric cameras, therefore, each of the N^e node/edge involving node i, will have N^t suggested time delays (once cross-correlated with them). For each cross correlation maximization (equation 3.5) two expectations are likely to happen: a) random time-delay values suggested by incorrect corresponding nodes/edges, or b) consistent time-delay values suggested by correct correspondences. Therefore, we initialize the time delay of node i as the median of all the suggested values for that specific node. For instance, time delay of egocentric video $\mathbf{E}_i$ is initialized as the following:

$$\mathbf{t}^*{}_{d_i} = \widetilde{\mathbf{T}}_{d_i} \tag{4.4}$$

where T_{d_i} is the set of all implicitly suggested time-delays implicitly by the elements of the two graphs:

$$\mathbf{T}_{d_i} = \{\underset{\mathbf{t}_{d_i}}{\operatorname{argmax}}\ B_{ij}^{GIST} * B_{kl}^{IOU} | \forall\ j \leq N^e, k, l \leq N^t\}. \tag{4.5}$$

We evaluate the effect of this initialization by comparing it to the results of initializing $\mathbf{t}_d$ as a vector of zeros.

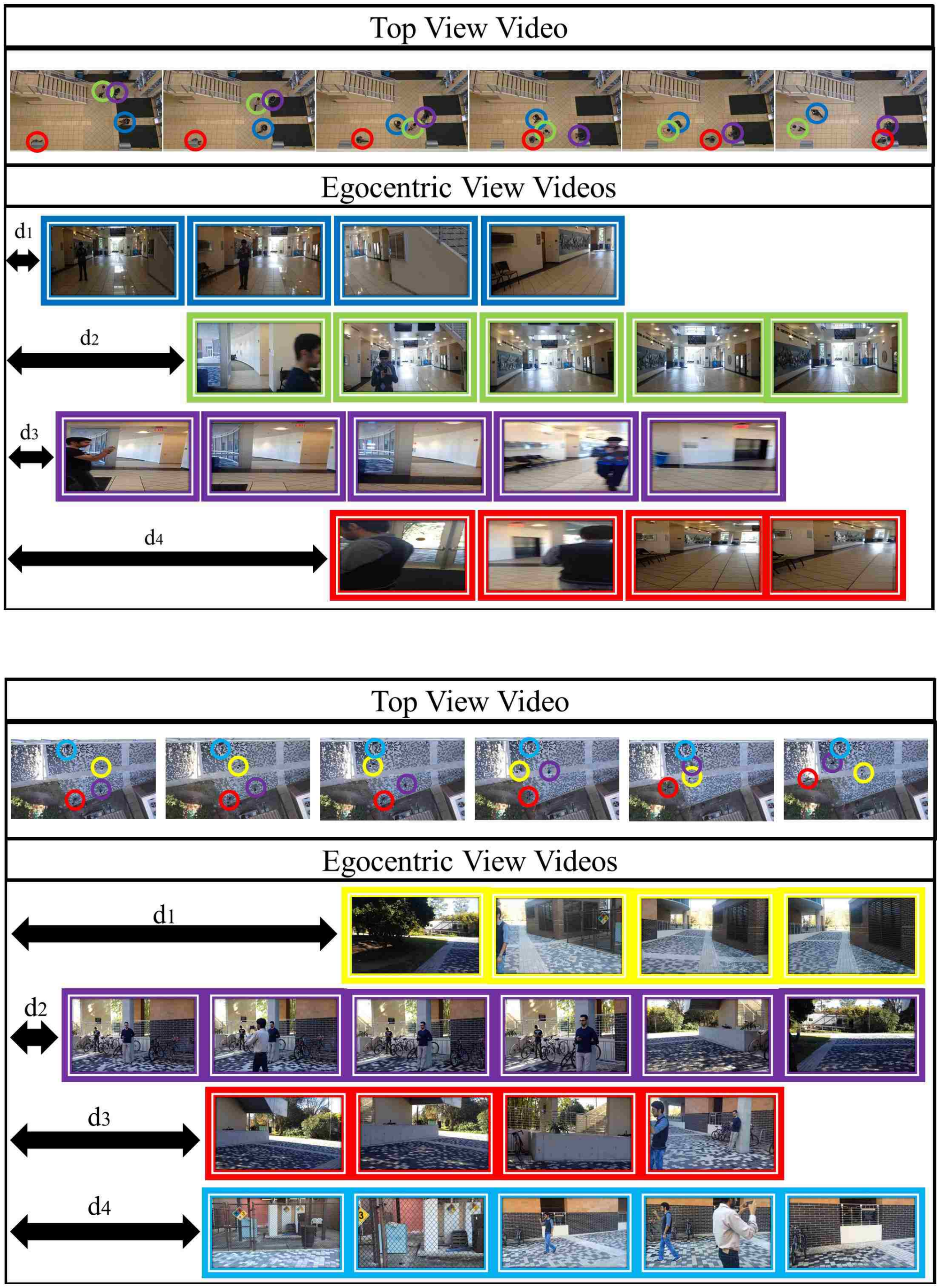


Figure 4.1: Sampled frames of the top-view, and egocentric videos are shown. The time-delays of the egocentric videos with respect to the top-view video are also shown. Color-coding denotes the viewer identities.

4.1 Experimental Results

In this section, we mention details of our experimental setup, collected data, evaluation measures, as well as some baseline methods.

4.1.1 Cross-view Video Dataset

The dataset containing 50 test cases of videos shot in three different indoor and outdoor environments. Each test case contains one top-view and several egocentric videos captured by the people visible in the top-view video. Egocentric viewers were asked to move around with normal walking pace and collect videos with no specific instructions. They could see other moving viewers and occasionally bystanders. An example test case is shown in Figure 4.1. Overall, our dataset contains more than 225,000 frames. Number of people visible in the top-view cameras varies from 3 to 10, number of egocentric cameras varies from 1 to 6, and the ratio of number of available egocentric cameras to the number of visible people in the top-view camera varies from 0.16 to 1. Lengths of the videos vary from 320 frames (10.6 seconds) up to 3132 frames (110 seconds).

4.1.2 Performance Evaluation

We validate our method in terms of answering the two questions asked in the Introduction section. First, given a top-view video and a set of egocentric videos, can we verify if the top-view video is capturing the egocentric viewers? See Section 4.1.2.2 for the answer. Second, knowing that a top-view video contains the viewers recording a set of egocentric videos, can we determine which viewer has recorded which egocentric video? (section 4.1.2.3 and 4.1.2.4). We also evaluate the performance of our method in jointly addressing the two questions. Given a set of egocentric videos and a set of top-view videos, can we identify the egocentric viewers in the set of the top-view videos? We address this in section 4.1.2.5. We evaluate the performance of our algorithms in performing temporal alignment, and evaluate the effect of parameters such as number of egocentric videos and video length. We also perform analytical and practical run-time analysis of the proposed approaches.

4.1.2.1 Methods

Proposed Methods: We evaluate our two proposed iterative methods (spectral and graph matching score based optimization) for enforcing time-delay consistency.

Baseline Methods: We compare our proposed methods to the following baselines:

- **Graph Matching (GM):** is the framework proposed in [2] upon which we build our algorithms,
- **2D Unary:** is solely using the 2D unary features of the graphs comparing FOV IOU to gist similarity 2D features, plugged in the framework proposed in [2],
- **1D Unary:** is solely using the 1D unary features of the graph comparing number of humans visible in FOV of the egocentric viewers over time.

We describe other task specific baselines in their corresponding sections.

4.1.2.2 Ranking Top-view Videos

We design an experiment to evaluate if our graph matching score is a good measure for the similarity between the set of egocentric videos and a top-view video. Having a set of egocentric videos from the same test case (recorded by a group of viewers in the same environment), and 50 different top-view videos (from different test cases), we compare the similarity of each of the top-view graphs to the egocentric graph. After computing the hard assignment for each top view video (resulting in the assignment vector x), the score $x^T Ax$ is associated to that top-view video. This score is effectively the summation of all similarities between the corresponding nodes and edges of the two graphs. All the top-view videos are evaluated and ranked using this score. The ranking accuracy is computed by measuring the rank of the ground truth top-view video, and computing the cumulative matching curves shown in Figure 4.2. The x-axis in the cumulative matching curves encodes different ranks (top-k), and y-axis encodes the percentages of the test cases where the correct match was within the top-k rank. The yellow curve is a simple visual baseline in which we compute the average visual feature of all the egocentric videos (gist features) and compare that to the average gist features of all the top-view videos. The green and blue curve show the performance if we only include the unary features in graph matching. The cyan curve shows the ranking accuracy when we apply the graph matching method of [2], where time-delay consistency is not enforced. The magenta and red curve show the ranking accuracy of our proposed algorithms, spectral optimization and matching score based optimization respectively. The dashed black line shows the accuracy of randomly ranking the top-view videos.

It can be observed that all the curves outperform the random ranking. The magenta and red curve outperforming the cyan curve indicates the effectiveness of our time-delay consistency enforcement. Overall, the red curve giving us the best results, showing that our second algorithm outperforms the spectral level. Please note that both of the proposed methods were initialized using the medians of the suggested values as described in the initialization section. In general, this experiment answers the first question. Indeed, graph matching score can be used as a cue for narrowing down the search space among the top-view videos, for finding the one corresponding to our set of the egocentric cameras.

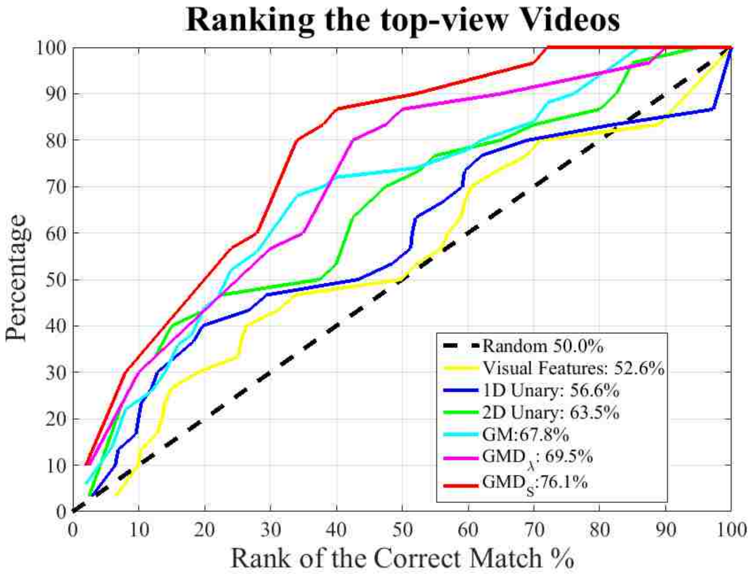


Figure 4.2: The cumulative matching curve demonstrates the performance of the proposed spectral, and matching based optimization methods (red and magenta), and compares them with the baseline graph matching method introduced in [2]. It shows that our proposed algorithms outperform the baselines by more than 1.3% and 3.3%.

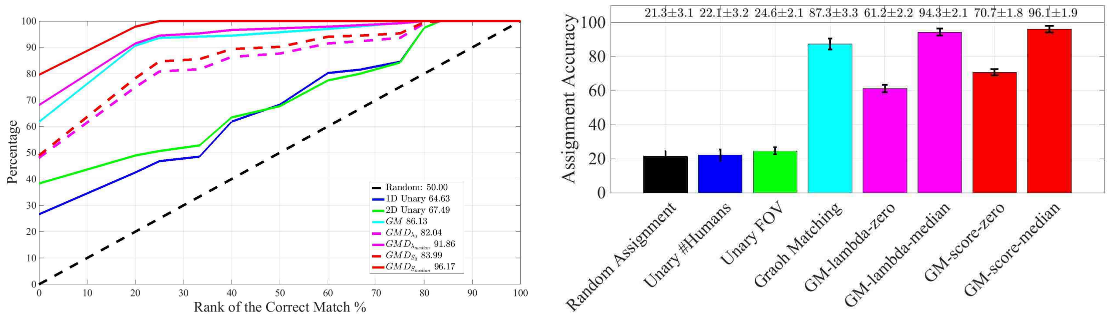
4.1.2.3 Viewer Ranking

Given the top-view video containing the egocentric viewers, we evaluate our soft assignment approach in terms of its capability in ranking top-view viewers. In other words, we can look at our soft assignment as a measure to sort the viewers in the top-view video based on their assignment probability to an egocentric video. Computing the ranks of the correct matches, we can plot the cumulative matching curves to illustrate their performance.

We evaluate the performance of our proposed methods, each with two different initializations, and compare their performance with four baselines in Figure 4.3 (a). First, random ranking (dashed black line), in which for each egocentric video we randomly rank the viewers present in the top-view video. Second, sorting the top-view viewers based on the similarities of their 1D unary features to the 1D unary features of each egocentric camera (i.e., number of visible humans illustrated by the blue curve). Third, sorting the top-view viewers based on their 2D unary features (GIST vs. FOV, shown by the green curve). Notice that, here, we are ignoring the pairwise relationships (edges) in the graphs (the blue and green curves). The cyan curve illustrates the accuracy of the method used in [2], and the magenta and red curves show the performance of our spectral based and graph matching score based methods, respectively. Solid curves are the outcomes of median initialization, while the dashed curves are results with zero initialization. It can be observed that correctly initializing the time-delays has a significant impact on the performance.

4.1.2.4 Assignment Accuracy

In order to answer the second question, we need to assess the accuracy of our method in terms of hard-assignment. Having a set of egocentric videos and a top-view video corresponding to the egocentric viewers, we compute the percentage of egocentric videos that were correctly matched to their corresponding viewer. We compare the hard-assignment accuracies of our two proposed algorithms with two different initializations, with four baselines in Figure 4.3(b). Similar to the ranking performance, the first baseline is random assignment. For this purpose, we randomly assign each egocentric video to one of the visible viewers in the top-view video. The second baseline is performing Hungarian bipartite matching only on the 1D unary feature which is the count of visible humans over time. The third baseline is performing Hungarian bipartite matching only on the 2D unary feature (GIST vs. FOV, denoted as Unary FOV), ignoring the pairwise relationships (edges) in the graphs. The fourth baseline is Graph Matching method introduced in [2]. The consistent improvement of the Graph Matching method using both unary and pairwise features (denoted as GM) over the baselines shows the significant contribution of pairwise features in the assignment accuracy. The last four columns show the assignment accuracies using the two iterative algorithms proposed in this work. We find that initializing the time delays as a vector of zeros does not improve the assignment accuracy. Instead, using the median of suggested time-delays boosts the assignment accuracy significantly. The highest accuracy is achieved by median-based initialization and the graph matching score using the iterative-alternative algorithm, which results in 96.1% assignment accuracy. The promising accuracy acquired by graph matching answers the second question. Knowing a top-view camera is capturing a set of egocentric viewers, we can use visual cues in the egocentric videos and the top-view video to decide reliably which viewer is capturing which egocentric video.



(a) (b)

Figure 4.3: (a) shows the cumulative matching curve for ranking the viewers in the top-view video. The green and blue curves belong to ranking based on the cross correlation between the 2D, and cross correlation between the 1D unary scores, respectively (Not incorporating pairwise features). The cyan curve is the graph matching method results (section 3.1.3), and the magenta and red curves are the results of the two iterative approaches with two different initializations. The dashed black line shows random ranking accuracy (b) shows the assignment accuracy mean and standard deviations based on randomly assigning, using the number of humans, using unary features, using spectral graph matching, and using our two iterative approaches with two different initializations. The best performance in both (a) and (b) is achieved by the matching score based iterative optimization, when the time-delays are initialized by the median of the suggested values.

4.1.2.5 Joint Evaluation

In the viewer ranking evaluation, we assumed that the correct top-view video is given and we solely need to rank the visible viewers. Here, we jointly evaluate the two objectives and assume that we do not know which top-view video contains the egocentric viewers. In other words, given a set of egocentric videos and a set of top-view videos, we search for each egocentric viewer in the content of all of the top-view videos. We sort all the 235 identities visible in all of the top-view videos based on their matching score to each egocentric video. We compute the matching score by combining the two similarity measures that we used for our two separate tasks. In Section 4.1.2.2, we compute a graph matching score for matching the egocentric set E to top view video k , which we call $s(E,k)$. In Section 4.1.2.3, we assumed that we know the correct top-view video and then computed the similarity of an egocentric viewer to each top-view identity. That similarity for egocentric video i and top-view identity j is defined as the soft assignment probabilities computed by the spectral graph matching ($p(i,j)$). Since we hold the assumption that the top-view video being considered (video k) is the correct top-view video, in essence we can consider $p(i,j)$ as a conditional probability of $p(i,j|k)$. Therefore we can combine the two scores and define the similarity of egocentric video i in egocentric set E to identity j in top-view video k as $s(E,k)p(i,j|k)$. Given a query egocentric video, we sort all top-view identities in all top-view videos based on this score and evaluate our the ranking performance. The cumulative matching curve is shown in Figure 4.4. Depending on which graph matching algorithm is used (method introduced in [2] or our iterative algorithms), $s(E,k)$ and $p(i,j|k)$ would be different for each egocentric video - top-view identity pair. We evaluated the performance of each method and demonstrated their performance in figure 4.4. It can be seen that the proposed iterative algorithms (highlighted in magenta and red) outperform the baseline (cyan curve) introduced in [2].

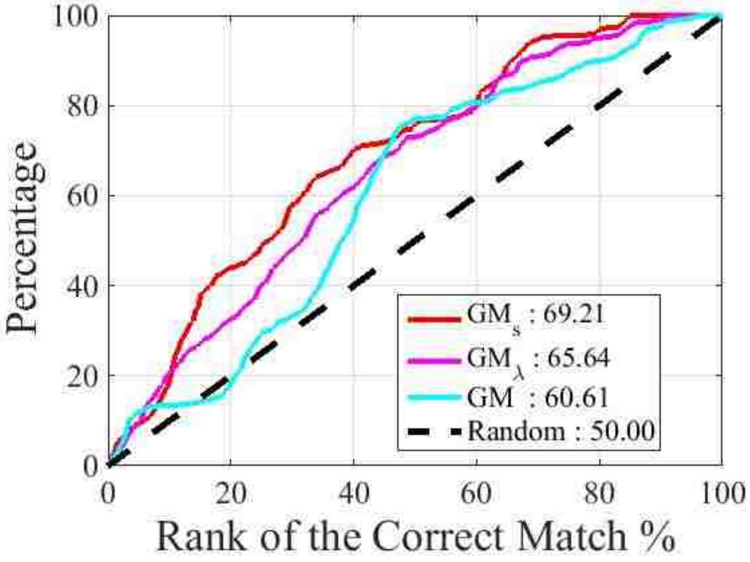


Figure 4.4: The cumulative matching curve demonstrates the performance of the proposed spectral, and matching based optimization methods (red and magenta), and compares them with the baseline graph matching method introduced in [2] (cyan) in terms of jointly performing the two tasks. It shows that our proposed algorithms outperform the baseline by more than 5.03% and 8.6%.

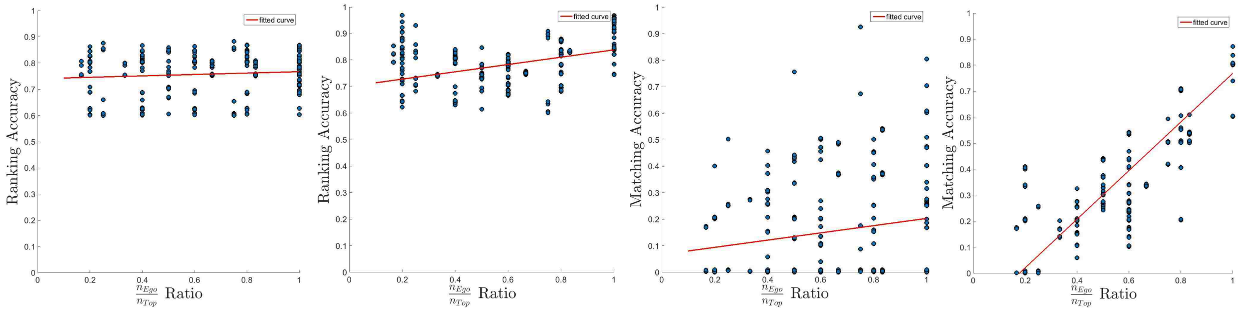
4.1.2.6 Temporal Misalignment

We evaluated the effect of our two algorithms in terms of temporal alignment error. The error distributions are shown in Figure 4.9. The shift in the distribution shows the effectiveness of the proposed algorithms in terms of reducing temporal misalignment across the egocentric and top-view videos. Similar to assignment accuracy and ranking, the best performance is achieved using the graph matching based optimization (blue distribution).

4.1.2.7 Effect of Number of Egocentric Cameras

Earlier we evaluated the performance of our method given all the available egocentric videos present in each set as the input to our method. In this experiment, we compare the accuracy of assignment and ranking as a function of the *completeness ratio* ($\frac{n_{Ego}}{n_{Top}}$) of our egocentric set. Each of our sets contains $3 < N^t < 11$ viewers in the top-view camera, and $2 < N^e < 8$ egocentric videos. Depending on the included subset of egocentric videos, we can generate up to 2,862 instances of our problem. We evaluate the accuracy of our method and baselines using different subsets of the egocentric videos. A total of $2^{N^e} - 1$ non-empty subsets of egocentric videos is possible depending on which egocentric video, out of N^e, is included (i.e., all possible non-empty subsets).

Figure 4.5 illustrates the assignment and ranking accuracies using the graph matching method [2] versus the ratio of the available egocentric videos to the number of visible people in the top-view camera. It shows that as the completeness ratio increases, the assignment accuracy improves drastically. Intuitively, having more egocentric cameras gives more information regarding the structure of the graph (by providing more pairwise terms) which leads to improvement in the spectral graph matching and assignment accuracy.



(a) (b) (c) (d)

Figure 4.5: Effect of the relative number of egocentric cameras referred to as the completeness ratio ($\frac{n_{Ego}}{n_{Top}}$). (a) shows the ranking accuracy vs $\frac{n_{Ego}}{n_{Top}}$, only using the unary features, (b) shows the same evaluation using the graph matching output, (c) shows the accuracy of the hard assignment computed based on Hungarian bipartite matching on top of the unary features, and (d) shows the hard-assignment computed based on the spectral graph matching.

4.1.2.8 Effect of Video Length in Assignment Accuracy

Here, we analyze the effect of video length in assignment accuracy [2]. We use smaller portions of the videos and measure how the assignment accuracy changes as we increase the clip length. Shown in Figure 4.8, as the video length increases, the assignment accuracy increases. Intuitively, longer videos provide more discriminative unary and pairwise features, and therefore lead to better performance.

4.1.3 Results on Synthetic Data for Testing Scalability

Since our dataset contains limited number of people in each test case, we evaluated the scalability of our proposed algorithms in terms of computation time using synthetic data. We generated a set of trajectories in 2D space by randomly generating 2D initial positions, speed and acceleration. This set of trajectories is then transformed to another set using a similarity transform (i.e., rotation and translation). Random noise, with different standard deviations, is added to the trajectory points. A random temporal offset is also added to each trajectory in the transformed set (examples shown in Figure 4.6). We then form graphs similar to the ones in Section 3.1.1 on each set of trajectories. We compute unary and binary features similar to the features that were extracted from top-view trajectories. These graphs and their features (extracted similar to the top-view features in our method; Section 3.1.1) are then fed to the proposed algorithms. Computation time is then evaluated as functions of graph size.

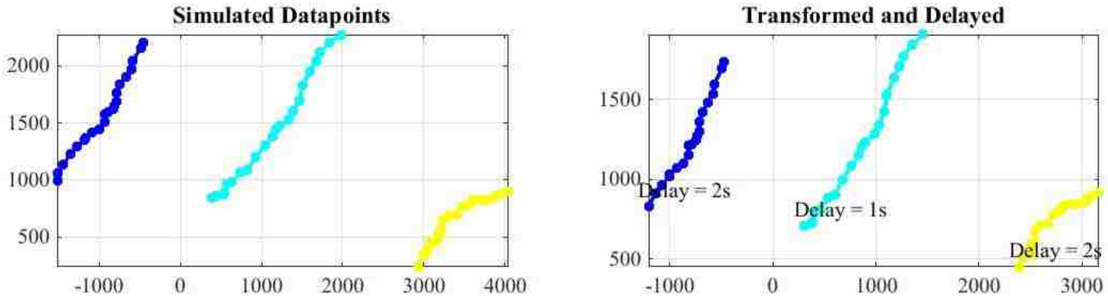


Figure 4.6: An example of the synthetic data generated for scalability testing. Left shows the set of generated trajectories. Right shows the transformed and temporally delayed trajectory set.

4.1.4 *Run-time Analysis*

In this section, we analytically and empirically evaluate the run-time and scalability of our proposed approach and compare it to that of [2]. We show that the computational bottleneck is computing the cross-correlations across the features, which is in common between the two frameworks. Our proposed algorithms do not add significant additional time-complexity to that.

4.1.4.1 *Analytical*

Let n be the number of top-view identities, m be number of egocentric viewers, and t be the video length. Egocentric and top-view videos have roughly the same order or number of frames. The top-view graph then has $\mathcal{O}(n^2)$ features (nodes and edges), and the ego graph has $\mathcal{O}(m^2)$ features. As a result, we compute $\mathcal{O}(m^2n^2)$ cross-correlations for forming the affinity matrix and initializing the time-delay vector. Each 2D cross-correlation operation takes $\mathcal{O}(t^4)$ time. Thus, the whole initialization procedure takes $\mathcal{O}(m^2n^2t^4)$. Notice that this part is the same in [2] and our proposed framework. In [2], after the initialization, we perform eigen decomposition and hard-to-soft assignment using the Munkres algorithm only once. In the proposed algorithms, however, we perform this computation several times. The estimation of the leading eigenvector of the affinity matrix (whose size is $mn \times mn$) takes $\mathcal{O}(m^2n^2)$ as it is estimated using the power iteration method. The Munkres algorithm also takes $\mathcal{O}(m^2n^2)$ with our implementation. At each iteration of our iterative algorithms, we modify the delay vector by performing a local grid search over the time delay of each egocentric video. This process takes $\mathcal{O}(m^2n^2)$ at each iteration as the cross-correlation is pre-computed and we only need to look up the new values in the affinity matrix using the candidate time delays. More specifically, at each iteration, we iterate over m different egocentric videos to update the elements in the affinity matrix related to that specific node and its connected edges (mn^2 elements), which would lead to $\mathcal{O}(m^2n^2)$. Thus, in the spectral level optimization algorithm, each iteration takes $\mathcal{O}(m^2n^2)$. For the graph matching optimization, it takes additional $\mathcal{O}(m^2n^2)$ for the Munkres algorithm, resulting in the same order in theory ($\mathcal{O}(m^2n^2)$), but in practice slightly longer per iteration.

In total the framework presented in [2] has time complexity of $\mathcal{O}(m^2n^2t^4) + \mathcal{O}(m^2n^2) + \mathcal{O}(m^2n^2) = \mathcal{O}(m^2n^2t^4)$. Given that our number of iterations k_{itr} is bounded by max_{itr}, which we set to 30, the proposed algorithms have the total time complexity of $\mathcal{O}(m^2n^2t^4) + k_{itr}\mathcal{O}(m^2n^2) = \mathcal{O}(m^2n^2t^4)$. Therefore, the overall time-complexity is of the same order as in [2], yet we achieve higher accuracy in all of the tasks.

4.1.4.2 Empirical

Each cross correlation (video lengths of around 300 frames) takes around 0.91 second, leading to 568.75 seconds for the initialization of two 5 by 5 graphs. Each iteration of the spectral level optimization takes approximately 0.21 second and the graph matching score based method approximately 0.28 second. In our experiments, on average the total time of all of the iterations of each spectral based instance takes 1.05 seconds, while the graph matching based method takes 1.51 seconds. Given the 568.75 seconds needed for the initialization, the run-time of both of our proposed algorithms are negligible. All experiments were performed in MATLAB using a 2.4 GHz CPU.

4.1.4.3 Scalability of Temporal-alignment Algorithms

We performed an experiment using our synthetic data (as discussed in Section 4.1.3), where we analyzed the effect of graph size. As shown in Figure 4.7 and discussed in section 4.1.4.1, the computation time increases polynomially with graph size $\mathcal{O}(m^2n^2)$. Given that the computation complexity of spectral graph matching is $\mathcal{O}(m^2n^2)$, any framework using spectral graph matching would have at least the same order of computational complexity. Please note that the measured time is the time of our iterative algorithms excluding the cross-correlation and initialization step which is in common between our method and [2]. As discussed in section 4.1.4.1, the time complexity of these steps is negligible in comparison to the cross-correlation step which is on $\mathcal{O}(m^2n^2t^4)$ as long as $t >> m,n$.

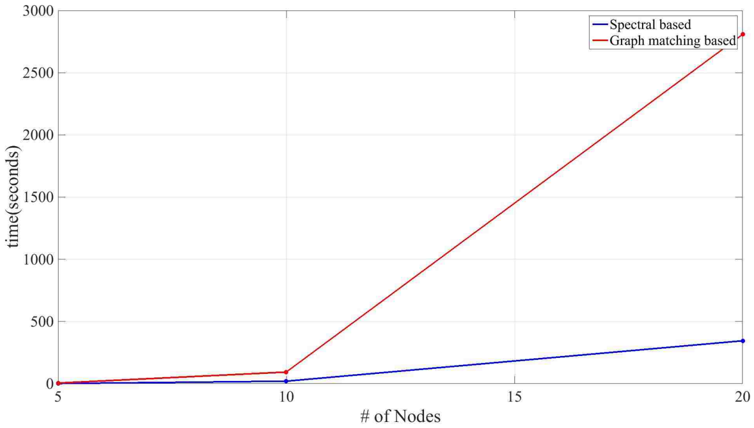


Figure 4.7: Testing scalability of the proposed algorithms in terms of computation time. It can be observed that the computation time increases polynomially with the size of the graphs for both of our proposed algorithms

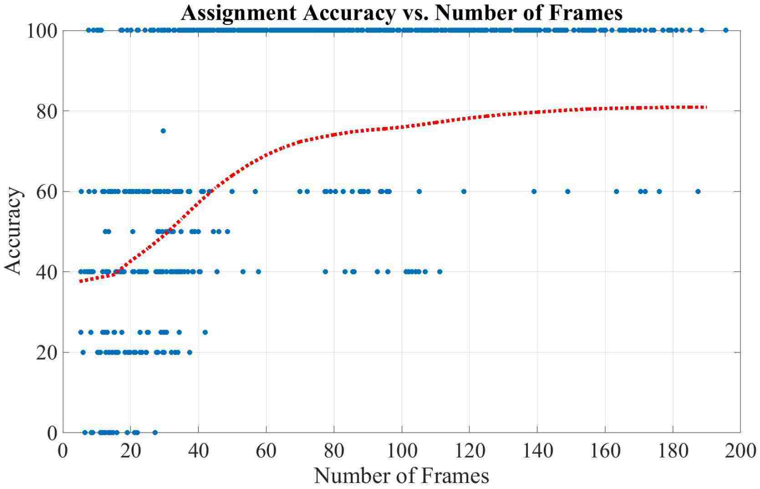


Figure 4.8: The effect of video length on the assignment accuracy. As the video length increases, the assignment accuracy improves. Please note that each blue dot represents one test sample, and the dashed red curve represents the cumulative mean assignment accuracy (mean of all accuracies with video length smaller than t)

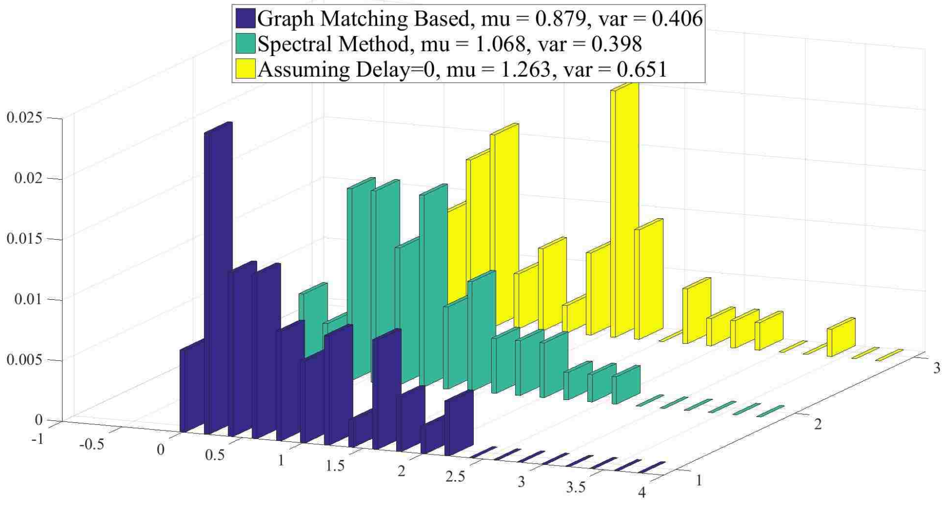


Figure 4.9: Distribution of temporal misalignments before (yellow) and after our spectral based iterative algorithm (green), and after our graph matching based method (blue). It can be observed that the distribution of the misalignments have been shifted to lower values. The average of misalignments have been reduced from 1.2 to 1.06 and .87 seconds.

4.2 Conclusion

Here we addressed two main questions regarding relating multiple egocentric videos to a single top-view video. First, can we tell if a set of egocentric videos belong to a set of humans present in a top-view video? And second, given that they do, can we identify the egocentric viewers in the top-view video? We proposed a unified framework that can properly answer these questions. Our experiments suggest that the pattern of change in the content of the egocentric videos, along with the relationships among them helps identify the viewers in top-view. To capture such patterns, we utilized a spectral graph matching technique and showed that the graph matching score is a meaningful criterion for narrowing down the search space in a set of top-view videos. Further, the assignment obtained by our framework is capable of associating egocentric videos to the viewers in the top-view camera. We conclude that meaningful features can be extracted from single, and pairs of egocentric camera(s), simply based on global scene gist of the content of the camera and incorporating the temporal information of the video(s). Empirical investigation shows that the assignment accuracy drops significantly if we do not include the binary features. This means that capturing the relationship among the viewers in top and egocentric views is a crucial factor. Also, enforcing consistency among the time-delays improved the accuracy in terms of assignment and ranking, as it prevents the system from producing invalid answers with contradictory implicit time-delay assignments. We also demonstrate that the completeness of the egocentric set is a key factor in the performance of our proposed algorithms. Generally, the more completed the egocentric set, the higher assignment and ranking accuracy of the graph matching method. Video length is another significant factor. Longer videos result in more discriminative patterns in 1D and 2D feature descriptors, and thus a more accurate assignment. Our work helps relate two domain that so far have been studied in isolation and can also be used to infer new insights regarding the visual world from different perspectives. As one example, we studied human identification but the same methods can be used for understanding behavior of other entities such as animals or cars.

CHAPTER 5: SIMULTANEOUS SELF-IDENTIFICATION, RE-IDENTIFICATION AND TEMPORAL ALIGNMENT OF EGOCENTRIC AND TOP-VIEW VIDEOS

Here we address two main shortcomings of the methods introduced in the previous chapters. First, identification is addressed in chapter 3 and 4 in terms of self-identification. The egocenric camera holder is identified but not the people visible in the content of the egocentric video. Here we aim to address this shortcoming and formulate re-identification in addition to self-identification. Second shortcoming of the previous approaches is their high dependency to the relationship between different egocentric videos. Here we propose a more robust framework capable of dealing with scenarios where such information is not possible. Provided with only one egocentric, and one top-view video, we address the three following problems:

Self-identification: The goal here is to identify the camera holder of an egocentric video in another reference video (here a top-view video). The main challenge is that the egocentric camera holder is not visible in his/her egocentric video. Thus, there is often no information about the visual appearance of the camera holder (example in Fig. 5.1).

Human re-identification: The goal here is to identify the humans seen in one video (here an egocentric video) in another reference video (here a top-view video). This problem has been studied extensively in the past. It is considered a challenging problem due to variability in lighting, view-point, and occlusion. Yet, existing approaches assume a high structural similarity between captured frames by the two cameras, as they usually capture humans from oblique or side views. This allows a rough spatial reasoning regarding parts (e.g., relating locations of head, torso and legs in the bounding boxes). In contrast, when performing human re-identification across egocentric and top-view videos, such reasoning is not possible (examples are shown in Figs. 5.1 and 1.5).

Temporal alignment: Performing temporal alignment between the two videos directly is non-trivial as the top-view video contains a lot of content that is not visible in the egocentric video. We leverage the other two tasks (self identification and re-identification) to reason about temporal alignment and estimate the time-delay between them.

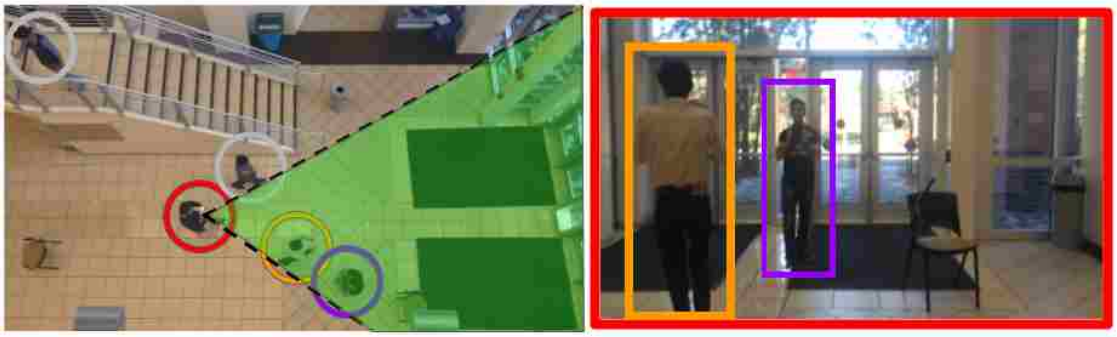


Figure 5.1: A pair of top- (left) and egocentric (right) views. Self identification is to identify the egocentric camera holder (shown in red). Human re-identification is to identify people visible in the egocentric video, in the content of the top-view video (orange and purple).

The interdependency of the three tasks mentioned above encourages designing a unified framework to address all simultaneously. To be able to determine the camera holder's identity within the content of the top-view video (task 1), it is necessary to know the temporal correspondence between the two videos (task 3). Identifying the people visible in the egocentric video in the content of the top-view video (task 2), would be easier if we already knew where the camera holder is in the top-view video at the corresponding time (tasks 1 and 3), since we can reason about who the camera holder is expected to see at any given moment. Further, knowing the correspondence between the people in ego and top views, and temporal alignment between two videos (tasks 2 and 3), could hint towards the identity of the camera holder (task 1). Finally, knowing who the camera holder is (task 1) and who he is seeing at each moment (task 2) can be an important cue to perform temporal alignment (task 3). The chicken-and-egg nature of these problems, encourage us to address them jointly. Thus, we formulate the problem as jointly minimizing the total cost $C_{tot}(l_s, L_r, \tau)$, where l_s is the identity of the camera holder (task 1), L_r is the set of identities of people visible in the egocentric video (task 2), and τ is the time offset between the two videos (task 3).

5.1 Framework

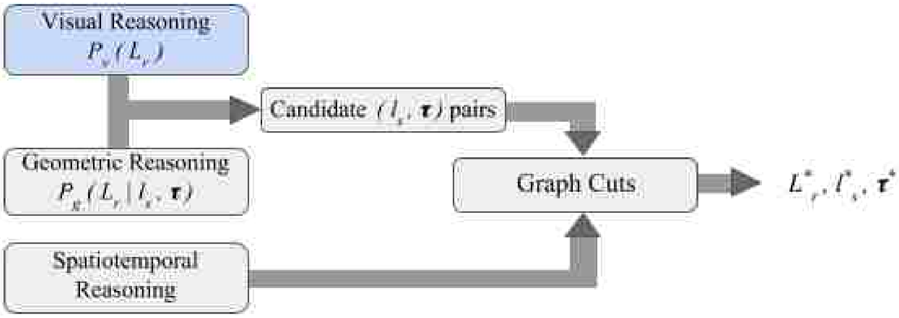


Figure 5.2: The block diagram of our proposed method. We use three main cues: visual, geometrical, and spatiotemporal. Visual reasoning is used for initializing re-identification correspondences. Combining geometric and visual reasoning, we generate a set of candidate (l_s, τ) pairs. Finally, we evaluate the candidates using graph cuts while enforcing spatiotemporal consistency and find the optimum combination of labels and values.

We aim to address three different tasks jointly. To find the optimal values for all of the variables in a unified framework, we seek to optimize the following objective:

$$l_s^*, L_r^*, \tau^* = \underset{l_s, L_r, \tau}{argmin}\ C_{tot}(l_s, L_r, \tau) \tag{5.1}$$

Assuming a set of identities visible in the top-view video as $I^t = \{1, 2, ..., |I^t|\}$, our goal in task 1 is to identify the camera holder (assign a self-identity l_s to the camera holder). We assume that the camera holder is visible in the content of the top-view video, therefore $l_s \in I^t$. In task 2, we aim to perform human re-identification for the visible humans in the egocentric video. Let $D^e = \{d_1^e, d_2^e, ..., d_{|D^e|}^e\}$ be the set of all human detection bounding boxes across all the frames of the egocentric video. Task 2 aims to find labeling $L_r = \{l_1^e, l_2^e, ..., l_{|D^e|}^e\} \in |I^t|^{|D^e|}$, which is the set of re-identification labels for human detection bounding boxes. Finally, τ is the time offset between the egocentric and the top-view video, meaning that the frame τ_0 in the top-view video corresponds to frame $\tau_0 + \tau$ in the egocentric video. We aim to estimate τ in task 3. In our notation, we use superscripts to encode the view (t: top, e: ego).

The block diagram of our proposed method is shown in Fig. 5.2. Our method is based on three types of reasonings across the two views. First, we perform visual reasoning across the two videos by comparing the visual appearance of the humans visible in the top video to the people visible in the egocentric video. This reasoning will provide us some initial probabilities for assigning the human detection bounding boxes in the ego-view to the identities in the top-view video. It gives an initial re-identification prior $P_v(L_r)$ based on the likelihood of the human detections matching to top-view identities (Sec 5.1.1). The second cue is designed to geometrically reason about the presence of different identities in each other's field of view in top-view over time (Sec 5.1.2), providing us cues for re-identification relative to different self-identification labels. We then define two spatiotemporal constraints to enforce consistency among our re-identification labels (Section 5.1.3) in ego view. In the fusion step (Sec 5.1.4), we combine visual and geometrical reasoning to narrow down the search space and generate a set of candidate (l_s, τ) pairs. Finally, we enforce spatiotemporal constraints and evaluate the candidates using graph cuts [88].

5.1.1 Visual Reasoning

The first clue for performing re-identification across the two views is to compare appearances of the two bounding-boxes. Since in traditional re-identification works both cameras are static, and they have similar poses (oblique or ground level), there is an assumption of rough spatial correspondence between two human detection bounding boxes (i.e. the rough alignment in location of head-torso-leg between two bounding boxes). Since the viewpoints are drastically different in our problem, the rough spatial alignment assumption does not hold. A few examples are shown in Fig. 1.5. We perform this task in unsupervised and supervised settings. In the unsupervised setting, we extract some generic features from the two views and directly compare their features. In the supervised setting, we design a two stream network capable of measuring similarity across the two views.

5.1.1.1 Unsupervised Approach

For each bounding box d_i^e in the ego-view, we extract VGG-19 deep neural network features [89] f_i^e (last fully connected layer, 4096 dimensional features). We perform L2 normalization on the features. As mentioned before, top-view bounding boxes are tracked and identities have been assigned to each track (set of bounding boxes belonging to each person). Therefore, for identity j in the top-view video, we extract VGG features from all of its bounding-boxes and represent identity j with the average of its feature vectors f_j^t.

To enforce the notion of probability, we measure the probability of ego-view bounding box d_i^e being assigned to label j in top-view ($l_i^e = j$) as:

$$P(l_i^e = j) = \frac{e^{-||f_i^e - f_j^t||}}{\sum_{m=1}^{|I^t|} e^{-||f_i^e - f_m^t||}}. \tag{5.2}$$

5.1.1.2 Supervised Approach:

Training: We train a cross-view two stream convolutional neural network to match humans across the two views. As illustrated in Fig. 5.3, each stream consists of convolution and pooling layers, ending in fully connected layers. The output is defined as the Euclidean distance of the output of the last fully connected layers of each stream passed through a sigmoid activation. If the two bounding boxes belong to the same identity, the output is set to zero (and one, otherwise). This forces the network to find a distance measure across the two views.

Testing: We feed bounding box d_i^e to the ego stream of the network and extract f_i^e (We perform L2 normalization). In top-view, for identity j we feed all of its bounding-boxes to the top-view stream and represent identity j with the average of its feature vectors f_j^t. Similar to the unsupervised approach, we measure the probability of ego-view bounding box d_i^e being assigned to label j in top-view ($l_i^e = j$) according to Eqn. 5.2.

Implementation details of the CNN: We resize each of the top-view bounding boxes to 40×40, and each ego-view bounding box to 300×100 in the RGB format (3 channels). Each stream consists of 3 convolutional blocks, each having two convolutional layers and a pooling layer with 2×2 pooling. The number of filters for the convolutional layers in order are 16, 16, 32, 32, 64, and 64. Finally, each stream projects to two fully connected layers (top stream: 512, 128; ego-stream: 1024, 128). The inner product of the output of the two streams is then passed through a sigmoid activation in order to enforce the notion of probability. We use Adam optimizer with learning rate of 0.001 and binary cross entropy loss, and train the network end-to-end. The hyper-parameters were fine-tuned on the validation set using grid search in logarithmic scale.

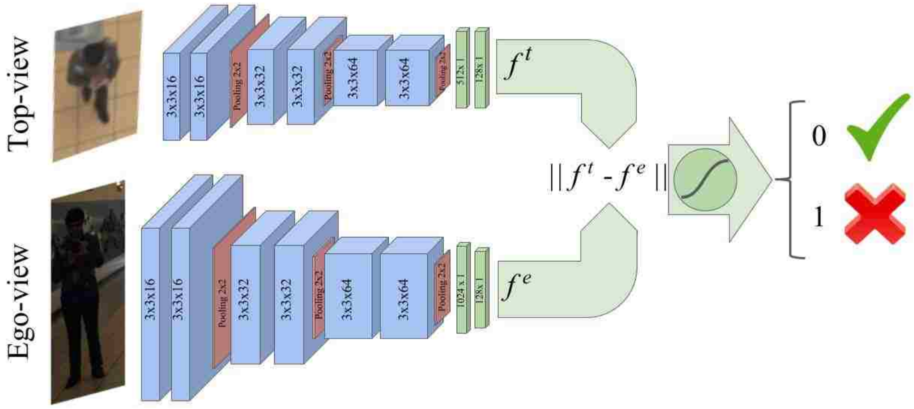


Figure 5.3: The architecture of our two stream convolutional neural network trained on pairs of bounding boxes. The Euclidean distance between the output of the last fully connected layers (i.e., top and ego) passed through sigmoid activation is set to 0 when the pair belongs to the same person and 1, otherwise.

5.1.2 Geometric reasoning

Here, we leverage the geometric arrangement of people with respect to each other in top-view and reason about their presence in each other's field of view. We iterate over different identities in top-view, and perform geometric reasoning assuming the identity is the camera-holder. In Fig.5.4, we illustrate reasoning about the presence of the identities highlighted with blue and orange bounding boxes, assuming the person highlighted in the red bounding box is the camera holder.

Given the identity of the camera holder (l_s), we compute how likely it is for each person i to be present in l_s's field of view (FOV) at any given time. Following [2], we perform multiple object tracking [12] on the provided top-view bounding boxes (provided by the dataset). Knowing the direction of motion of each trajectory at each moment, we employ the same assumptions used in [2]. We estimate the head direction of each of the top-view camera holders by assuming that people often tend to look straight ahead while walking. Since the intrinsic parameters of the egocentric camera (e.g., focal length and sensor size) are unknown, we consider a lower and upper bound for the angle of the camera holder's FOV (θ_1 and θ_2 in Fig.5.4) to estimate boundaries on l_s's field of view. As a result, we can determine the probability of each identity being present in the field of view of l_s (i.e., camera holder) at any given time ζ (Fig. 5.4, right side). We define the probability of identity i being present in the field of view of the camera holder (l_s) at time ζ as:

$$P_{g_\zeta}(i|l_s) = \begin{cases} 1, & \theta_i < \theta_1 \\ \frac{(\theta_2 - \theta_i)}{(\theta_2 - \theta_1)}, & \theta_1 < \theta_i < \theta_2 \\ 0, & \theta_2 < \theta_i \end{cases} \tag{5.3}$$

Intuitively, if the bounding box is within the lower bound of the FOV, we assign its presence probability to 1. If its orientation with respect to l_s is outside the upper-bound of the FOV range, we assign its presence probability to 0. For values in between the two bounds (e.g., the person at the bottom-left of Fig. 5.4), we assign its probability proportional to its orientation with respect to the camera holder. In our experiments, we empirically set θ_1 and θ_2 to 30° and 60°, respectively.

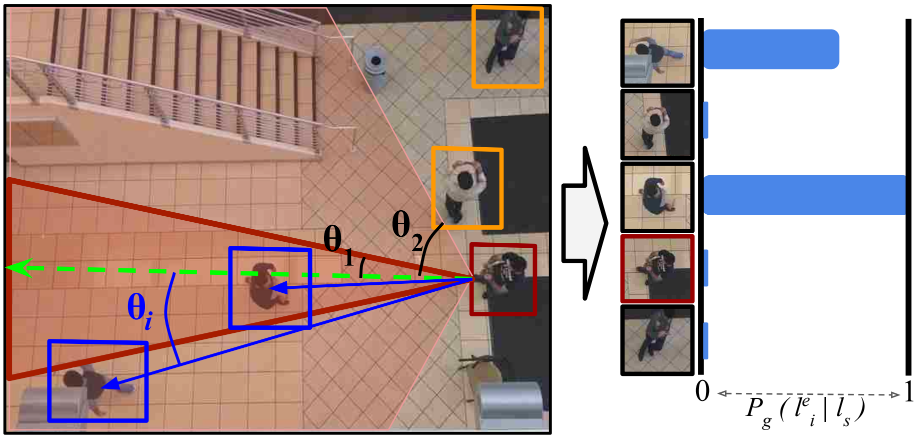


Figure 5.4: Geometric reasoning in the top-view video. In this example (left), two identities are present in the field of view of the egocentric camera holder (the two red cones showing the lower and upper bound of field of view). Using their orientation (shown by blue arrows) with respect to the camera holder's direction of movement in the top-view (dashed green arrow), we estimate the probability of their presence in the content of the egocentric video. Right bar graph shows the probability of each person being present in the FOV of the camera holder.

5.1.3 Spatiotemporal Reasoning

The third component of our approach enforces spatiotemporal constraints on the re-identification labels within the egocentric video. We define a cost for assigning the same identity label to a pair of human detection bounding boxes. We later incorporate this cost in our graph cuts formulation. Two constraints are defined as follows:

Constraint 1: Two different bounding boxes present in the same frame cannot belong to the same person. Note that non-maximum suppression is performed in the human detection process. Therefore the binary cost between any pair of co-occurring bounding boxes is set to infinity.

Constraint 2: If two bounding boxes have a high overlap in temporally nearby frames, their binary cost should be reduced, as they probably belong to the same identity. We incorporate two constraints in C_{st} cost as follows:

$$C_{st}(d_i^e, d_j^e) = \begin{cases} \infty, \text{ if } \zeta_{d_i^e} = \zeta_{d_j^e} \\ -1, \text{ if } 0 < |\zeta_{d_i^e} - \zeta_{d_j^e}| < \varepsilon \text{ and } \frac{A_{d_i^e} \cap A_{d_j^e}}{A_{d_i^e} \cup A_{d_j^e}} > \sigma \\ 0, \; Otherwise \end{cases} \tag{5.4}$$

where $A_{d_i^e}$ and $A_{d_j^e}$ correspond to the image area covered by human detection bounding boxes d_i^e and d_j^e, and $\zeta_{d_i^e}$ and $\zeta_{d_j^e}$ encode the time in which bounding boxes d_i^e and d_i^e are present. If d_i^e and d_j^e have been visible in the same frame, $C_{st}(d_i^e, d_j^e)$ will be set to infinity in order to prevent graph-cuts from assigning them to the same label (constraint 1). The negative cost of $C_{st}(i, j)$ in case of temporal neighborhood ($0 < |\zeta_{d_i} - \zeta_{d_j}| < \varepsilon$) and high spatial overlap ($\frac{A_{d_i^e} \cap A_{d_j^e}}{A_{d_i^e} \cup A_{d_j^e}} > \sigma$) will encourage the graph cuts algorithm to assign them to the same label (constraint 2), as they may correspond to the same identity if they have a high overlap. Here, we empirically set ε to 5 frames and σ to 0.8.

5.1.4 Fusion

In this section we describe how visual, geometrical, and spatiotemporal reasonings are combined. First, we combine the visual and geometrical reasoning to find a set of candidate (l_s, τ) pairs. We then examine each candidate pair using graph cuts to measure the cost of its resulting (L_r, l_s, τ) labeling and select the one with the minimum cost.

5.1.4.1 Candidate Selection

In section 5.1.1, we described how an initial human re-identification prior can be obtained using visual reasoning. In section 5.1.2, we described how an independent source of information (geometric reasoning) provides yet another set of human identification priors given each possible self identity. In this section, we search over different self identities and time delays, and choose the one whose geometric re-identification patterns is consistent with the visual re-identification patterns.

Temporal representation: In section 5.1.1, we described how we can compute $P_v(l_i^e = j)$ for any given egocentric human detection bounding box d_i^e and top-view identity j. We can form a $T^e \times |I^t|$ matrix R^v, where T^e is the number of frames in the egocentric video and $|I^t|$ is the number of identities visible in the top-view video. Intuitively, $R^v(\zeta, j)$ captures the probability of visibility of top-view identity j in the field of view of egocentric camera holder at time ζ. Let $D^e_\zeta = \{d^e_{\zeta_1}, d^e_{\zeta_2}, ..., d^e_{\zeta_{|D^e_\zeta|}}\}$ be the set of human detection bounding boxes visible in frame ζ of the egocentric video. We define $R^v(\zeta, j) = \sum_{i=1}^{|D^e_\zeta|} P_v(l_i^e = j)$. Since the sum of the probabilities might lead to a value higher than 1, we truncate the value at 1. In other words $R^v(\zeta, j) \leftarrow min(1, R(\zeta, j))$. An example R^v matrix is shown in Fig.5.5 (center panel).

We can form a similar matrix based on the geometric reasoning for each self-identity. As described in section 5.1.2, given the self identity of the camera holder (l_s), we can compute $P_g(l_i^e = j|l_s)$. Similar to R^v, we can form $T^t \times |I^t|$ matrix R^g where T^t is the number of frames in the top-view video, and $R^g(\zeta, j)|_{l_s} = P_{g_\zeta}(i|l_s)$, which is computed according to Eqn. 5.3. Intuitively $R^g(\zeta, j)|_{l_s}$ is the probability of visibility of identity j in the field of view of self-identity l_s at time ζ of the top-vie video, geometrically (an example shown in Fig. 5.5-left). Forming R^v and $R^g|_{l_s}$ for different self-identities (l_s), we expect them to have similar patterns for the correct l_s. For each top-view identity l_s, we compute the cross correlation of its $R^g|_{l_s}$ matrix with R^v across the time dimension in order to evaluate their similarities across different time delays (τ). This cross correlation results in a 1D signal encoding the similarity score of the two matrices given different time offsets. As shown in Fig. 5.5, we estimate the time offset between the two videos (assuming self-identity l_s) by finding the maximum of that score. We search across all self identities and sort them based on their maximum cross correlation score.

$$l_s^*, \tau^* = \underset{l_s,\tau}{argmax}\ R^v \odot R^g|_{l_s} \tag{5.5}$$

where $\odot$ denotes element-wise multiplication. Please note that all the videos in our dataset are captured with the same frame rate. Thus, we can perform all of these computations in a frame-based manner. Otherwise, a pre-processing and quantization on the temporal domain would be necessary to match the two matrices in terms of temporal comparability.

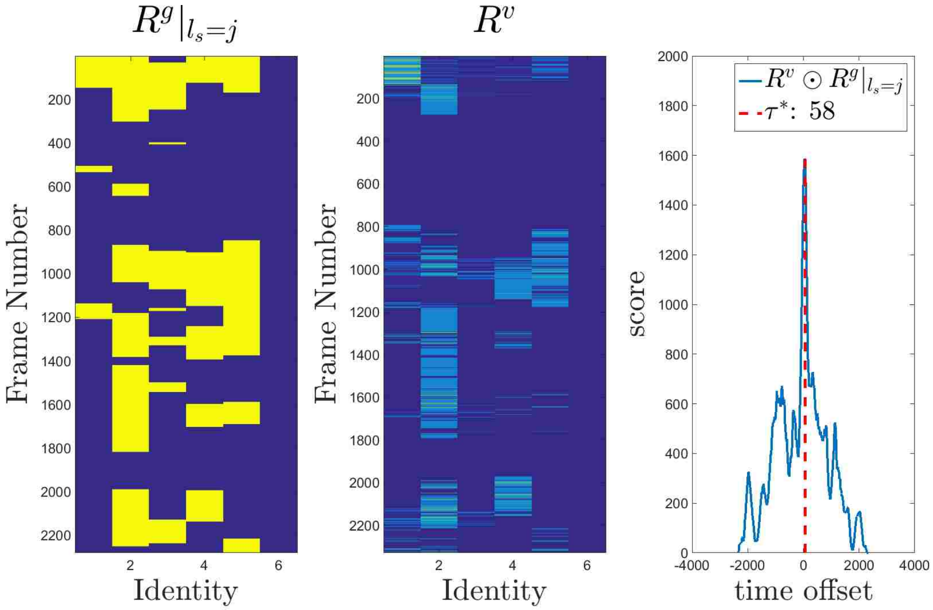


Figure 5.5: An example of estimating the self-identity and temporal offset. For a certain self-identity (l_s), the geometric reasoning is performed and the suggested re-identification priors are stored in matrix $R^g|_{l_s}$ (values color-coded). The matrix acquired by visual reasoning (in this case the supervised CNN based method) is shown in the middle (R^v). The similarity between the patterns in two matrices suggests that the self identity (l_s) is a good candidate. By correlating the two matrices across the time domain (the rightmost panel), we can observe a peak at $\tau = 58$. This suggests that if the camera holder has in fact identity l_s, the time-offset of his egocentric video with respect to the top-view video is 58 frames. Also, the score of self-identity l_s is the maximum value of the cross correlation which is 1587 in this case. By computing this value for all of the possible self-identities, we can pick the most likely self identity as the one with the highest score. More examples of this step are included in the supplementary material.

5.1.4.2 Graph Cuts:

Given a set of suggested (l_s, τ) pairs from the previous section, we evaluate the overall labeling cost as the cost of assigning l_s to the self identity, τ to the time delay, and L_r to the re-identification labels. Graph cuts allows the re-identification labels to adjust to the spatiotemporal constraint.

We form a graph $G(V,E)$ in which nodes are the human detection bounding boxes in ego-view $V = \{d^e_1, d^e_2, d^e_3, ..., d^e_{|D^e|}\}$ (See Fig. 5.6 for an illustration.). The goal is to assign each node to one of the top-view labels. Edges of the graph encode the spatiotemporal constraints between the nodes (as described in Section 5.1.3). Given the self identification label and time delay, we can perform graph cuts with its cost defined as:

$$C_{tot}(l_s, \tau) = \sum_{i=1}^{|D^e|} \left[C_u(l^e_i | \tau, l_s) + \sum_{j=1, j \neq i}^{|D^e|} C_{st}(l^e_i, l^e_j) \right] \tag{5.6}$$

The first term in rhs of Eqn. 5.6 encodes the unary cost for assigning d^e_i to its label l^e_i, given self-identity l_s and relative temporal offset (τ) between the two videos. We set C_u as:

$$C_u(l^e_i = j | \tau, l_s) = 1 - R^v(\zeta^e_i, j) R^g(\zeta^e_i - \tau, j) | l_s \tag{5.7}$$

where ζ^e_i is the time in which human detection bounding box d^e_i appears in the ego-view. Intuitively Eqn. 5.7 means that the probability of bounding box d^e_i (appearing at time ζ^e_i in the ego-view) being identity j in top-view, is the probability of the visibility of identity j at the field of view of l_s at time $\zeta^e_i - \tau$ in the top-view, multiplied by its likelihood of being identity j visually. The binary terms determine the costs of the edges and encode the spatiotemporal cost described in section 5.1.3. The output of this method provides us with a cost for each (l_s, τ) pair, alongside with a set of labellings for the human detection bounding boxes L_r. The pair with the minimum cost and its corresponding L_r is the final solution of our method (i.e., l^*_s, L^*_r, τ^*).

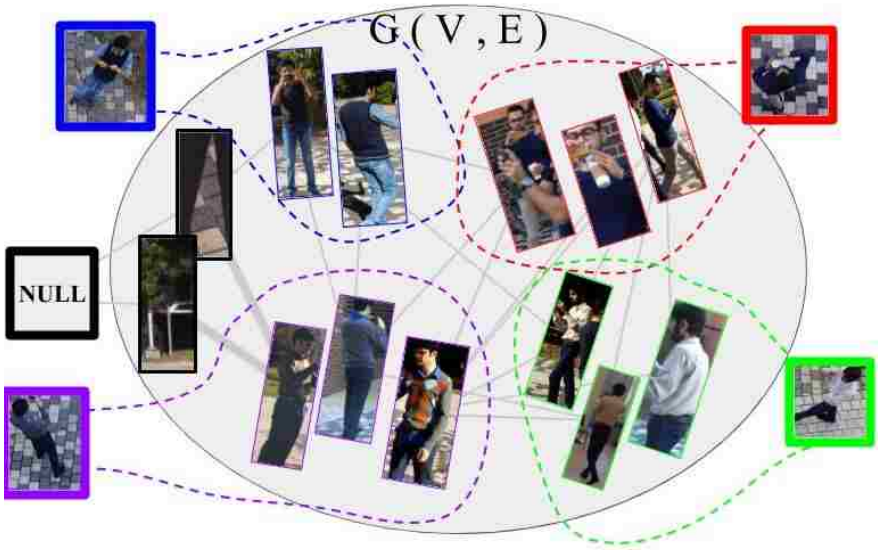


Figure 5.6: An illustration of the graph formation. The silver oval contains the graph G(V,E) in which each node is one of the ego-view human detection bounding boxes. The squared bounding boxes highlight different top-view labels in different colors. The graph cuts are visualized using the dashed colored curves. We always consider an extra NULL class for all of the human detection bounding boxes that do not match any of the classes.

5.2 Experimental Results

Here we describe our experimental setup and the dataset used for evaluating the proposed approach. We analyze the contribution of each component, and evaluate baselines, upper-bounds, and state of the art in order to assess the efficacy of our approach.

5.2.1 Dataset

We use the publicly available dataset [2]. It contains sets of videos shot in different indoor and outdoor environments. Each set contains one top-view and several egocentric videos captured by the people visible in top-view. Each ego-top pair is used as an input to our method. We used three sets for training our two stream neural network and the rest for testing. There are 47 ego-top test pairs and therefore 47 cases of self-identification and temporal alignment. The total number of human detection bounding boxes, and therefore human re-identification instances is 28,250. We annotated the labels for all the 28,250 human detection bounding boxes and evaluated the accuracy for re-identification and self-identification. The number of people visible in top-view videos vary from 3 to 6, and lengths of the videos vary from 1,019 frames (33.9 seconds) up to 3,132 frames (104.4 seconds).

5.2.2 Evaluation

We evaluate our proposed method in terms of addressing each objective and compare its performance in different settings. Moreover, we analyze the contribution of each component of our approach in the final results.

4.2.1. Self-identification: We evaluate our proposed method in terms of identifying the camera holder in the content of the top-view video. Since we perform self-identification based on initial re-identification probabilities (visual reasoning), we evaluate self-identification based on supervised and unsupervised re-identification results, alongside with state-of-the-art baselines. We also evaluate the performance in each setting before and after the final graph cuts step to assess the contribution of the spatiotemporal reasoning. Upper-bounds of the proposed method are also evaluated by providing the ground-truth re-identification and temporal alignment. The cumulative matching curves are shown in Fig.**??**. The solid yellow curve is the performance of [2]. As explained before, [2] highly relies on the relationship among multiple egocentric videos and does not perform well when it is provided with only one egocentric video. The dashed yellow curve shows the performance of [20]. The network provided by the authors was used. We tried fine-tuning the network on our dataset, but no improvement was achieved. As explained in the related work section, this framework is not designed for scenarios such as ours. The cyan and blue curves show our self-identification accuracy in the unsupervised setting before and after the graph cuts step, respectively.

The magenta and red curves show the performance in supervised setting, before and after the graph cuts step, respectively. The dashed black curve shows random ranking (performance of chance). The advantage of graph cuts and the spatiotemporal constraints can be observed by comparing before and after graph cuts curves. The contribution of our two stream visual reasoning is evident by comparing the unsupervised curves with their corresponding supervised settings. The effect of the geometrical reasoning could be seen by comparing visual reasoning results, and the before GC curves. The numbers in the figure legend show the area under each curve for quantitative comparison. The margin between the supervised and unsupervised approaches shows the effect of re-identification quality on self-identification performance, confirming the interconnectedness of the two tasks. The solid green and solid black curves show the upper-bounds of the proposed method. We evaluate self-identification, when providing ground-truth re-identification labels and the time-delay to the proposed approach.

4.2.2. Cross-view human re-identification: We compute the human re-identification performance in supervised and unsupervised settings, before and after graph cuts (shown in Fig. **??**). In order to better assess the performance, we compute the performance of our proposed method given the ground truth self identification label ($l_{s_{gt}}$), and ground truth time delay τ_{gt} (Fig. **??**), which results in upper-bounds for re-identification performance. In both figures (a and b), the dashed black line shows the chance level performance. The dashed cyan and magenta curves show the performance of direct visual matching across the two views using our unsupervised and supervised visual reasonings, respectively. Solid cyan and magenta curves show the performance of our unsupervised and supervised visual cues combined with geometric reasoning. Which is re-identification solely based on unary confidences in Eqn. 5.7 and before applying graph cuts. Finally, blue and red curves show performance of the unsupervised and supervised methods (in order) after the graph cuts step, which enforces the spatio-temporal constraints. Black solid curve in Fig **??** shows the performance of the proposed method, given the ground truth time delay between the two videos in addition to the ground truth self-identity. Comparing the red curves of Fig. **??** and **??** shows the effect of knowing the correct self identity on re-identification performance and thus confirming the inter dependency of the two tasks. Comparing the red and black solid curves in Fig. **??** shows that once the self-identity is known, correct time-delay does not lead to a high boost in re-identification performance which is consistent with our results on self-identification and time delay estimation. Comparing Fig. **??** a and b shows that knowing the correct self identity improves re-identification. As explained before, any re-identification method capable of producing a visual similarity measure could be plugged into our visual reasoning component. We evaluate the performance of two state of the art re-identification methods in Table 5.1. Before Fusion is the performance of each method in terms of Area under curve of cumulative matching curve (similar to Fig. **??**). After fusion is the overall performance after combining the re-identification method with our geometrical and spatiotemporal reasoning.

Table 5.1: Performance of different re-identification methods. Before Fusion is the performance of the re-identification method directly applied to the bounding boxes (only visual reasoning). After fusion shows the performance of our method if we replace our two stream network with the methods mentioned above.

Method	Before Fusion	After Fusion
Ours (Unsupervised)	0.537	0.612
Ahmed [40]	0.563	0.621
Cheng[41]	0.581	0.634
Ours (supervised)	**0.668**	**0.716**

4.2.3. Time-delay estimation: Defining τ_{gt} as the ground truth time offset between the egocentric and top-view videos, we compute the time-offset estimation error ($|\tau^{*} - \tau_{gt}|$) and compare its distribution with that of baselines and upper bounds. Fig. **??** shows the distribution of time-offset estimation error. In order to measure the effectiveness of our time-delay estimation process, we measure the absolute value of the original time-offset. In other words, assuming $\tau^* = 0$ as a baseline, we compute the offset estimation error (shown in the dark blue histogram). The mean error is also added to the figure legend for quantitative comparisons. Please note that the time delay error is measured in terms of the number of frames (all the videos have been recorded at 30fps). The baseline $\tau = 0$ leads to 186.5 frames error (6.21s). Our estimated τ^* in the unsupervised setting, reduces this figure to 138.9 frames (4.63s). Adding visual supervision reduces this number to an average of 120.6 frames (4.02s). To have upper bounds and evaluate the performance of this task alone, we isolate it from the other two by providing the ground-truth self identification ($l_{s_{gt}}$) and human re-identification labels ($L_{r_{gt}}$). Providing $l_{s_{gt}}$ will lead to 97.39 frames error (3.24), and providing both $l_{s_{gt}}$ and $L_{r_{gt}}$ reduces the mean error to 90.32 (3.01s). Similar to our re-identification upper-bounds, knowing the self-identity improves performance significantly. Once self-identity is known, the ground truth re-identification labels will improve the results by a small margin.

5.2.3 Run-time Analysis

Analytical: The computational bottleneck of our method is the graph cuts step performed for each self-identity. The run-time of our approach (excluding the graph cuts step) is $\mathscr{O}(|I^t|^3 T^t)$ which grows polynomially with the number of identities and linearly with video length. More details are presented in supplementary materials.

Empirical: Being provided with the human detection results, here we report the empirical run time of our algorithm. As an example, forming the visual reasoning matrix takes 5.12 seconds for 5 top-view identities. Computing the spatiotemporal costs for the egocentric video takes 1.1 seconds. The geometrical reasoning and CRF for each top-view identity takes 1.51 seconds. The whole process for evaluating all 5 identities takes 7.55 seconds. All the implementations are in MATLAB using a 2.4 G.Hz CPU.

5.3 Discussion and Conclusion

Limitations: We employ certain assumptions which even though are accurate in majority of scenarios, may not hold in some cases. Future work could explore alternative approaches capable of dealing with such cases. We assume that top-view human detections are given. We also formulate the problem as assigning an egocentric video to a full trajectory in top-view. In case of imperfect tracks, a generalized version of our problem allowing assigning more than one ego video to a trajectory could be explored.

Conclusion: We explored three interconnected problems in relating egocentric and top-view videos namely human re-identification, camera holder self-identification, and temporal alignment. We perform visual reasoning across the two domains, geometric reasoning in top-view domain and spatiotemporal reasoning in egocentric domain. Our experiments show that solving these problems jointly improves the performance in each individual task, as the knowledge about each task can assist solving the other two.

CHAPTER 6: ACTION RECOGNITION ACROSS FIRST AND THIRD PERSON VIDEOS

The results of this work has been published in the following paper:
”An Exocentric Look at Egocentric Actions, and Vice Versa”, Shervin Ardeshir, Ali Borji, Computer Vision and Image Understanding (CVIU) 2018.

In this section we explore relating first and third person videos in the context of action classification, retrieval, and matching. This work has received a minor revision to be accepted in the following:
An Exocentric Look at Egocentric Actions, and Vice versa. Shervin Ardeshir, and Ali Borji. Computer Vision and Image Understanding (CVIU).
In the following, we go through the proposed approach, the dataset collected for evaluating our model, the experimental results, and conclusion respectively.

6.1 Framework

In the following, we first explain our proposed network alongside with our baseline network architectures. We then go over the training and testing procedures, and the details of our approach for addressing each of the three tasks of action classification, matching and retrieval. We then describe our dataset and experimental setup.

6.1.1 Network Architectures

We design a two stream convolutional neural network (CNN) [90] to address the above-mentioned tasks. Ideally, we would like the network not only to be able to predict the action class of an input video but also generate a view-invariant feature vector that can be used for retrieval and matching. Given a video, egocentric or exocentric, the view-invariant feature will be used to perform video retrieval from the other view. Given a pair of videos, one from each view, their view invariant features can be used to perform matching and to tell whether they belong to the same action class. In the following, we describe our proposed architecture and baselines.

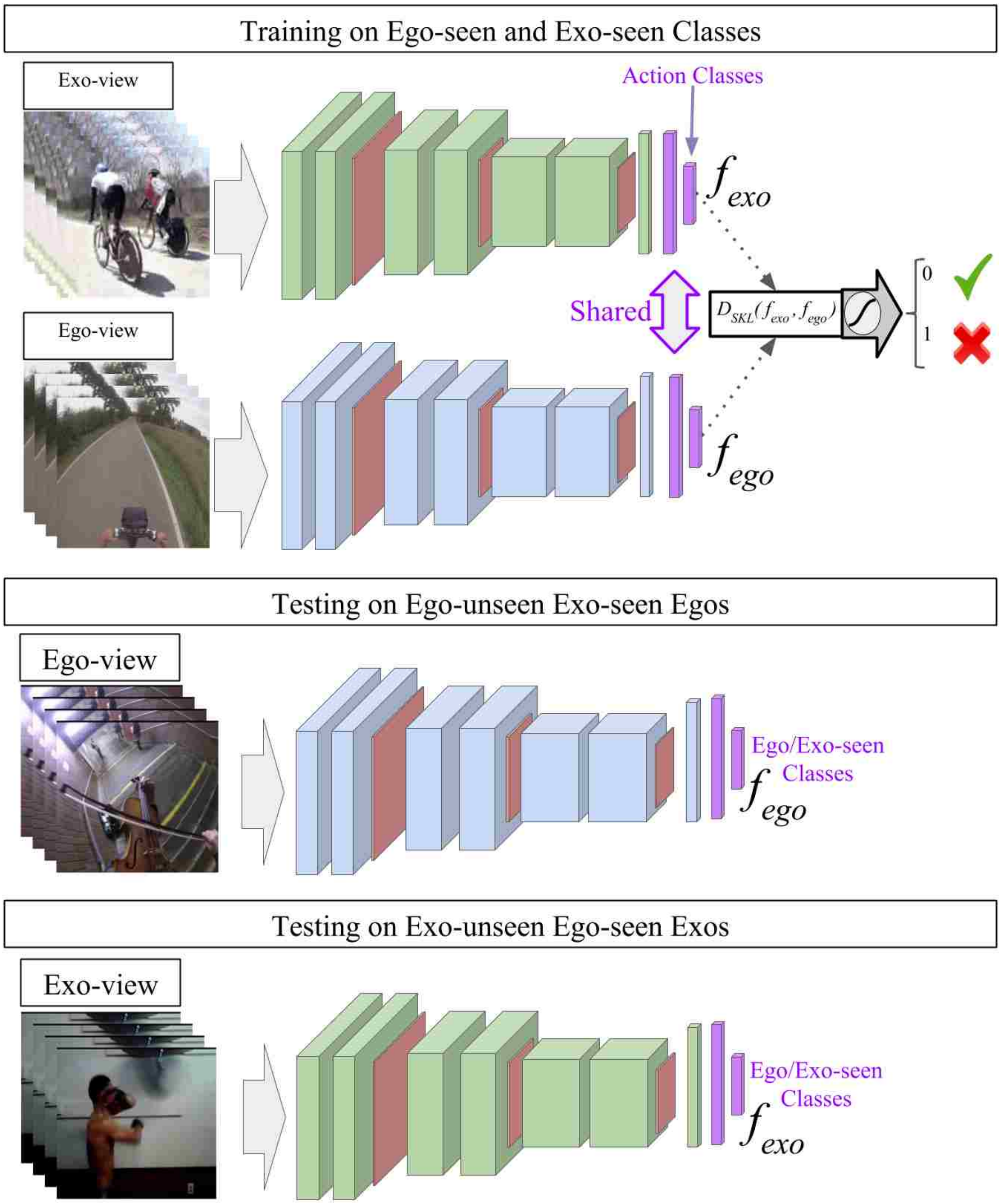


Figure 6.1: Our proposed architecture with view-specific feature extractors and shared classifiers. We train a 2-stream network in which one stream aims to extract features suitable for action recognition in ego domain, and the other stream extracts features in exo domain. We enforce the network to produce similar features across the two views if their action classes match by minimizing their KL-divergence. At test time, we perform action classification in both domains and across all the classes for which at least one of the streams has been trained (i.e., seen). We show that our network is capable of performing matching and retrieval across the two views based on the action class.

6.1.1.1 Proposed Architecture

As depicted in Figure 6.1, our network consists of two streams of convolutional neural networks. One stream is fed with the third person videos (exo-view; highlighted in green) while the other stream is fed with the first person videos (ego-view; highlighted in blue).

Each stream starts with convolutional and pooling layers and ends in shared fully connected layers. The two streams share a fully connected layer and a softmax classification layer. Since the spatial correspondence between arbitrary egocentric and exocentric videos are non-trivial, we do not share any of the convolutional layers across the streams and only share the fully connected layer which contains global information. Given a pair of videos, our network produces three outputs: 1) an action classification label for the egocentric video, 2) an action classification label for the exocentric video, and 3) a distance metric which computes the difference between the two videos. The retrieval measure encourages the output of the softmax layer (f_{ego} and f_{exo}) to have similar embeddings across the two views if they belong to the same action class. We enforce this similarity using the symmetric Kullback-Leibler (KL) divergence and incorporate it in the loss function. As we show later, the retrieval measure allows the network to effectively transfer knowledge from the seen to the unseen classes. Keeping the initial layers separate for the two views encourages the network to learn view-specific features. The details of each feature extraction stream (highlighted in green and blue in 6.1) and the shared classification block (highlighted in purple in 6.1) can be found in Table 6.1.

Table 6.1: Configuration details of our network. All the convolution kernels are set to 3 by 3 and their strides are set to 1. The top part contains the specifications of each stream in our proposed architecture, and both baselines. The bottom part contains the shared layers in our two-stream network and the two-stream baseline. The one stream baseline (baseline 1) also contains these layers after one stream of the feature extraction block. f_{exo} and f_{ego} are the output of the softmax layers of the exo and ego streams, respectively.

Each Exo/Ego-stream (Feature extraction block)		
Layer	Operations	Output Size
Conv$_{11}$	Convolution, BN, ReLU	$16 \times 128 \times 128$
Conv$_{12}$	Convolution, BN, ReLU	$16 \times 128 \times 128$
Pool$_1$	Max Pooling	$16 \times 64 \times 64$
Conv$_{21}$	Convolution, BN, ReLU	$32 \times 64 \times 64$
Conv$_{22}$	Convolution, BN, ReLU	$32 \times 64 \times 64$
Pool$_2$	Max Pooling	$32 \times 32 \times 32$
Conv$_{31}$	Convolution, BN, ReLU	$64 \times 32 \times 32$
Conv$_{32}$	Convolution, BN, ReLU	$64 \times 32 \times 32$
Pool$_3$	Max Pooling	$64 \times 32 \times 32$
AveragePool$_1$	Avg Pooling	$64 \times 16 \times 16$
Flatten$_1$	Flatten	64×1
Shared (Classification block)		
Layer	Operations	Output Size
FC$_1$	Fully Connected, ReLU	32×1
FC$_2$	Fully Connected, ReLU	32×1
Softmax$_1$	Softmax	14×1
Symmetric KL-divergence	$D_{KL}(f_{ego}\|\|f_{exo}) + D_{KL}(f_{exo}\|\|f_{ego})$	1×1
Sigmoid	$\frac{1}{1+e^{-x+5}}$	1×1

6.1.2 Training

First, we divide each video into 16 rgb-frame temporal windows and feed a pair of $128 \times 128 \times 3 \times 16$ blocks of data to corresponding streams in the network. The output labels are the action classes and the retrieval term (0 if they belong to the same class, and 1, otherwise). We augment the data by adding the horizontally flipped versions of the videos. We set batch size to 32, number of sample pairs to 640,000, and number of validation sample pairs to 32,000. Each training sample consists of two 16-frame clips of a random pair of videos (v_{ego} and v_{exo}) and their corresponding class labels (l_{ego} and l_{exo}). Class labels are used for the classification branches, and $1-(l_{ego} = l_{exo})$ is fed to the retrieval branch. We trained the network from scratch, instead of using pre-trained networks, in order to make sure it is completely uninformed about the unseen classes. Using a pre-trained model would prevent us from measuring the capability of our network in terms of generalizing to unseen classes, as it might have been exposed to the relationship between seen and unseen classes.

6.1.2.1 Optimization

The total loss of the network consists of three terms. The first two terms account for the classification loss in each stream, using the categorical cross-entropy. The third term is defined as the logical expression verifying if the two videos are from the same action class. We calculate the KL divergence between the outputs of the softmax layer from the two views and pass it through a sigmoid function. If the two videos belong to the same action class, this value is set to 0. If they belong to different classes, this value is set to 1. Given an egocentric video v_{ego} with action label l_{ego}, and exocentric video v_{exo} with action class l_{exo}, the total loss is

$$L(v_{ego},\ v_{exo},\ l_{ego},\ l_{exo}) = L^C_{ego} + L^C_{exo} + L^R, \tag{6.1}$$

where L^{C}_{ego} and L^{C}_{exo} are the classification losses for the egocentric and exocentric videos, respectively. The last term is the retrieval loss and is defined as

$$L^{R} = |1 - (l_{ego} = l_{exo}) - \sigma(D_{SKL}(f_{ego}, f_{exo}))|, \tag{6.2}$$

where D_{SKL} encodes the symmetric KL divergence and is defined as

$$D_{SKL}(f_{ego}, f_{exo}) = D_{KL}(f_{ego}||f_{exo}) + D_{KL}(f_{exo}||f_{ego}). \tag{6.3}$$

Including the symmetric KL divergence in the objective function enforces the two streams to produce similar output vectors for similar classes.

One may argue that the classification loss might be enough for generating similar output vectors. However, this loss only considers the class with the maximum score, and ignores other classes. The KL divergence term, on the other hand, takes into account the distribution of the two output vectors rather than simply the class with the maximum score. This will enforce the network to embed same (different) action classes in the two views to have similar (different) representations, thus allowing the network to discover features that are suitable for tasks such as action matching and retrieval.

The reason behind using a shifted sigmoid ($s(x) = \frac{1}{1+e^{-x+a}}$) in Eqn. 6.2 is to squash the divergence values to the [0,1] range. Here, a is set to 5. Since symmetric KL divergence always yields a positive value, we used a to shift the sigmoid function towards the positive values, in a way that a large portion of its active range is in the positive range.

6.1.3 Testing

In what follows, we describe in detail how the network is used to perform action classification and matching/retrieval. Given an input video (ego or exo), we first split it to 16-frame blocks, with an 8-frame overlap between blocks. Each block is then passed to its corresponding stream and its feature vector (f_{ego} or f_{exo}) is extracted. The mean of the extracted features represents the video and is used for action classification and retrieval.

6.1.3.1 Action classification

At test time, we evaluate the capability of each stream in terms of performing action classification in its own view. Given a video, we extract its view-specific features as described in section 6.1.3 and perform action recognition.

6.1.3.2 Matching and Retrieval

We perform matching and retrieval across the two views. We evaluate the capability of the network in terms of comparing two videos from the two views. Given a pair of videos (egocentric and exocentric), the goal is to determine if the two videos belong to the same action class or not and also learning a proper distance measure between the two. We use the retrieval branch of the network shown in Figure 6.1 for this purpose. At test time, we generate random pairs of egocentric-exocentric test videos from the dataset and feed them to the network. The output of the retrieval branch then indicates whether they match or not. Action matching is defined as a binary classification problem. Performance is measured as the fraction of cases for which the network has made the right decision. We threshold the output of the retrieval branch (symmetric KL-divergence between the features) after the sigmoid activation on 0.5 to achieve the matching binary label.

In order to evaluate the performance of our proposed method in terms of retrieval, given an egocentric or exocentric video, we extract its global feature as described in section 6.1.3. As an example, for the egocentric video i (v^i_{ego}), we compute its global feature f^i_{ego}. We then compare f^i_{ego} to all the global features computed for the exocentric test videos $F_{exo} = \{f^1_{exo}, f^2_{exo}, ..., f^n_{exo}\}$. Finally, we sort all the elements in F_{exo} in terms of their symmetric KL-divergence with respect to f^i_{ego}. In this task, we would ideally want the top-ranks with minimum divergence to be exocentric videos from the same class.

6.1.4 Networks

In order to fully assess the performance of our model, we design several variations of it described as follows. In addition, we evaluate two well-known domain adaptation approaches on our data.

6.1.4.1 One-stream single view baseline (1S1V)

We evaluate the performance of a one-stream network, trained on one view and tested on the other view. For the sake of brevity, we refer to this baseline as 1S1V. We evaluate this network in action classification.

6.1.4.2 One-stream single view fine-tuned baseline (1S1V-F)

We train a one-stream network on one view and fine-tune the last fully connected layers of the trained model on action classification in the other view. We evaluate this network in action classification.

6.1.4.3 One-stream both-views baseline (1S2V)

We evaluate a single-stream network trained for action classification and retrieval on videos from both views. We evaluate this baseline in terms of classification and retrieval/matching.

6.1.4.4 Two-stream classification baseline (2S2V-C)

Our second baseline is similar to our proposed architecture with the only difference that here the retrieval term is not used in the loss function. We evaluate this baseline in terms of classification and retrieval/matching.

6.1.4.5 Two-stream retrieval baseline (2S2V-R)

This baseline is the proposed architecture, excluding the classification branches. This baseline is only used for retrieval as it lacks classifiers.

6.1.4.6 Proposed method: Two-stream classification and retrieval model (2S2V-CR)

We evaluate the performance of the proposed network containing both the classification and retrieval branch.

For the two stream baselines, the loss is similar to Eqn. 6.1, except that it does not include the retrieval or classification terms in 2S2V-C and 2S2V-R, respectively. For the one stream two view (1S2V) architecture, we have a view agnostic action classification loss (i.e., only one of the first two terms in Eqn. 6.1).

6.1.4.7 Unsupervised domain adaptation by back-propagation

Here, we evaluate the performance of [91] for action classification. We use a one-stream network (with the same architecture as ours) for classification and two (32 dimensional) fully connected layers ending in a one dimensional followed by a sigmoid activation for view classification.

6.1.4.8 Adversarial domain adaptation

We evaluate and report the performance of [85] on our dataset for action classification. First, a classifier is trained on the source domain (for which we use our one-stream architecture). Next, a discriminator is trained to differentiate between source and target domains (two 32-dimensional fully connected layers ending in a 1 dimensional output with sigmoid activation). A second one-stream network is then trained on the target domain data with the aim of fooling the discriminator. After training, we append the target network with the classifier of the source network and perform action classification in the target domain. Exocentric and egocentric sets are alternatively assigned as the source and target domains.

6.1.4.9 Combining unsupervised methods with our method

We combine the results of the aforementioned two methods with our best performing networks (i.e., 2S2V-C and 2S2V-CR) by computing the average of the output of the classification softmax layer from the unsupervised approaches [85, 91] and our models.

We evaluate [85], [91], 1S1V, and 1S1V-F in terms of classification, as they lack the retrieval branch. Also 2S2V-R is only evaluated in terms of retrieval, as it lacks the classification branch.

6.2 Experiments and Results

In this section, we evaluate the performance of our proposed method in each task alongside with baselines.

6.2.1 Action Classification

Table 6.2 contains the performance of different networks in terms of action classification. We are particularly interested in cases containing unseen classes (last two columns) which show the performance of the models in terms of transfer learning. Second and third column contains the action classification accuracy of different networks on ego-seen exo-seen classes. The superiority of the proposed method (2S2V-CR) and the two-stream baseline (2S2V-C) compared to the one-stream network (1S2V) shows the effectiveness of the view-specific feature extractors. We attribute the slight advantage of 2S2V-C over 2S2V-CR method in seen classes to the fact that it does not have the retrieval term in its objective function and therefore it solely optimizes its parameters for classification. Please note that the difference is marginal and within margin of error.

Column 4 and 5 of Table 6.2 show results of the transfer learning scenarios. Having ego-unseen exo-seen egocentric actions and ego-seen exo-unseen exocentric actions, the 2S2V-CR network (proposed) achieves the highest accuracy. 2S2V-C does not perform as well, which shows the effectiveness of the retrieval term. Also, 1S2V is not able to perform well which further justifies using view specific feature extractors. 1S1V (one stream network trained on the opposite view) performs close to the chance level, which justifies the necessity of domain adaptation. 1S1V-F fails to perform well on the unseen classes. This suggests that features trained on one view are not suitable for the other view.

Adversarial domain adaptation [85] does not achieve favorable results in our setup. Since unsupervised methods often assume that data distribution over different classes are similar in the two domains. This is clearly not the case in our scenario. For example, let the exocentric set be the source domain and egocentric set be the target domain. As shown in Figure 1.6, the training data in the source (exocentric) domain is a combination of ego-seen exo-seen, and ego-unseen exo-seen classes. The distribution of the unlabeled target (egocentric) data consists of ego-seen exo-seen, and ego-seen exo-unseen classes. Given that the two sets of ego-seen exo-unseen and ego-unseen exo-seen are mutually exclusive, distributions of different classes in the source and target domain are not similar. This will lead to low accuracy in the ego-seen exo-unseen, and ego-unseen exo-seen classes. As a result, using the labels in the ego-seen and exo-seen subset is necessary in order to learn a valid mapping in our setup.

The method in [91] achieves higher accuracy compared to [85], since it does contain a classification branch which is not blind to the target domain labels. It can be observed that [91] does perform better than our 1S1V and 1S1V-F baseline, but not as favorable as the two stream baselines. We also evaluate the combination of [91] and [85] with our two stream networks (2S2V-C,and 2S2V-CR). We combine the approaches by simply computing the average of the output of their classification probabilities.

Table 6.2: Action classification Accuracy (mean and standard deviations over 10 different runs). In the seen-seen scenarios, the proposed approach (2S2V-CR) and the two-stream baseline (2S2V-C) reach better classification accuracies. We are interested in the third and fourth columns where action classification has been successfully performed with no direct training and solely based on transferring knowledge from the other domain, where 2S2V-CR performs more favorable.

Class Type	Ego-seen Exo-seen		Ego-unseen Exo-seen	Ego-seen Exo-unseen
Method \View	Ego	Exo	Ego	Exo
Chance	7.14%	7.14%	7.14%	7.14%
1S1V	8.1 ±2.3%	7.3 ±1.9%	8.0 ±1.2%	7.2 ±2.1%
1S1V-F	21.1 ±1.1%	29.2 ±2.1%	7.8 ±1.6%	8.1 ±1.3%
1S2V	21.8 ±1.1%	31.2 ±2.4%	8.9 ±1.2%	9.1 ±1.0%
2S2V-C	38.2 ±1.4%	41.1 ±2.3%	10.4 ±0.9%	9.8 ±1.3%
Proposed: 2S2V-CR	37.9 ±0.7%	40.9 ±1.1%	**19.5** ±1.0**%**	**12.7** ±1.2**%**
Tzeng et al. [85]	10.3 ±2.5%	9.2 ±2.7%	7.0 ±2.3%	6.8 ±2.1%
[85] + 2S2V-C	18.11 ±2.0%	17.11 ± 1.9%	8.2 ± 2.1%	8.1 ± 2.2%
[85] + 2S2V-CR	19.23 ±2.6%	17.03 ± 2.2%	12.7 ± 1.9%	10.3 ± 2.4%
Ganin et al. [91]	21.1 ±2.1%	30.1 ±2.3%	10.1 ±1.9%	10.7 ±2.0%
[91] + 2S2V-C	33.13 ±2.3%	38.10 ± 2.2%	12.9 ± 2.2%	11.2 ± 1.8%
[91] + 2S2V-CR	33.92 ±2.2%	39.03 ± 2.6%	19.2 ± 2.1%	12.1 ± 1.6%

6.2.1.1 All-seen Scenario

As mentioned above, comparing the unsupervised adversarial approach with ours is unfair due to the fact that the main underlying assumption of source and target data having similar distributions over classes, does not hold in our setup. Thus, we evaluate the performance of different models in an additional experiment where all classes are ego-seen exo-seen. Results are reported in Table 6.3. It can be seen that 2S2V-C, 2S2V-CR, and its combination with [91] outperform the other approaches. Since all the classes are seen in both domains, 2S2V-C and 2S2V-CR perform roughly similar. This is expected as the difference between the two was significant only in unseen classes.

Table 6.3: Action classification Accuracy in an all-seen scenario.

Class Type	Ego-seen Exo-seen	
Method \View	Ego	Exo
Chance	7.14%	7.14%
1S1V	18.1%	17.3%
1S1V-F	41.1%	49.2%
1S2V	41.8%	51.2%
2S2V-C	68.2%	61.1%
Proposed: 2S2V-CR	67.9%	61.9%
Tzeng et al. [85]	20.3%	19.2%
[85] + 2S2V-C	38.11%	37.11%
[85] + 2S2V-CR	40.09%	39.08%
Ganin et al. [91]	45.2%	58.5%
[91] + 2S2V-C	68.6%	**63.1**%
[91] + 2S2V-CR	**68.8**%	62.8%

6.2.2 Matching and Retrieval

Here we evaluate the models in terms of matching and retrieval across the two views.

6.2.2.1 Matching

Table 6.4 shows the action matching results. We provide the networks with 1,000 random instances of video pairs and evaluate the matching accuracy in terms of the ratio of instances for which the network made the correct decision (i.e., whether the two videos belong to the same action class). We measured the performance on ego-seen exo-seen, ego-unseen exo-seen, ego-seen exo-unseen, and ego-unseen exo-unseen scenarios.

We find that the minimization of the KL-divergence in positive pairs (ego/exo pairs of videos with the same action class) and maximizing it for negative pairs (ego/exo pairs of videos from different action classes) helps the proposed network perform better compared to the two-stream baselines (2S2V-C and 2S2V-R). Even though 2S2V-C performed similar to our proposed method (2S2V-CR) in action classification of seen classes, it still significantly underperforms 2S2V-CR in action matching. The reason is that 2S2V-C only pays attention to the peak of the output of the softmax layer, whereas 2S2V-CR considers the probability distribution over all classes. This allows the network to deal with unseen classes better through learning their relationship with other classes. We also evaluated 2S2V-R (two stream network without the classification branch and solely with the retrieval branch). Our proposed method outperforms 2S2V-R indicating that using class labels as extra supervision boosts the matching capabilities of the network.

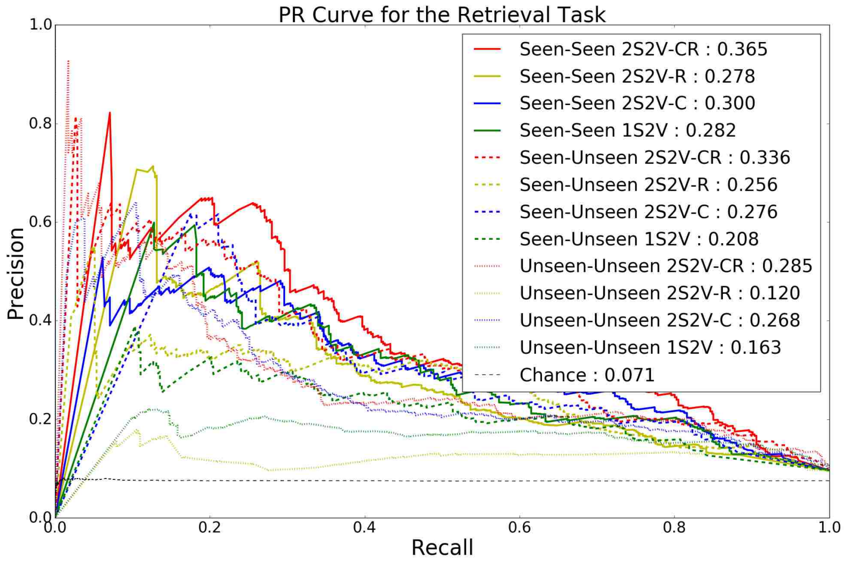
Table 6.4: Action matching accuracies for the proposed network and baselines. The proposed approach outperforms the baselines in all scenarios.

Test Scenario	Chance	1S2V	2S2V-C	2S2V-R	2S2V-CR
Ego-seen Exo-seen	50%	54.2%	61.2%	53.3%	**70.1%**
Ego-unseen Exo-seen	50%	51.1%	53.0%	52.1%	**58.7%**
Ego-seen Exo-unseen	50%	51.2%	51.4%	50.7%	**56.6%**
Ego-unseen Exo-unseen	50%	50.9%	51.1%	50.3%	**52.8%**

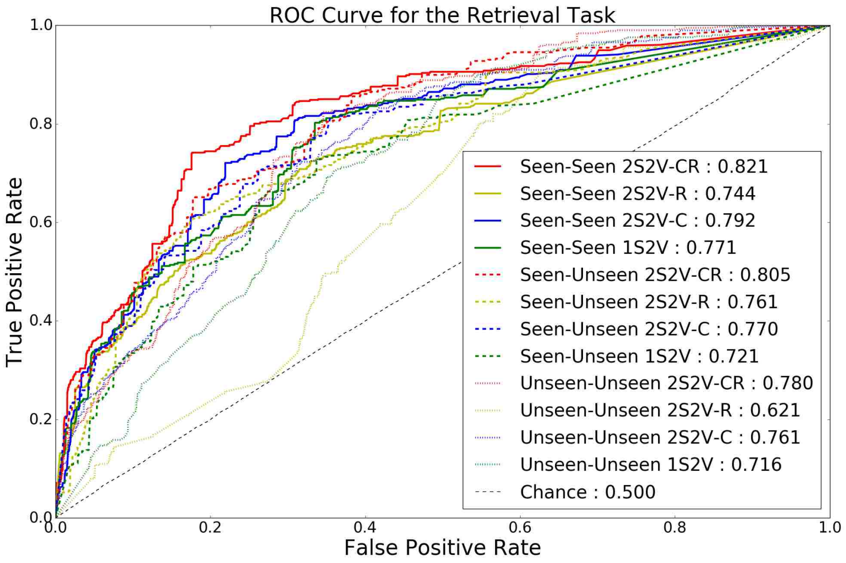
6.2.2.2 Retrieval

Given a query video, egocentric or exocentric, all the videos in the other view are sorted (in ascending order) based on their symmetric KL-divergence to the query video (using the global feature vectors as explained in 6.1.3). The goal is to have the videos with the same action label in the other view, within the top ranks. Retrieval performance is measured in terms of precision-recall and ROC curves shown in figures 6.2a and 6.2b, respectively. Interestingly, even for ego-unseen exo-unseen classes (the dotted curves), the retrieval performance is better than the chance level. This indicates that the network is capable of learning a global mapping between the two views, not bounded by the semantic space of its previous observations.

The quantitative results of the retrieval task are provided in Table 6.5 in terms of average precision and the area under the ROC curve. It can be observed that in all settings our proposed method (2S2V-CR) achieves the best retrieval performance.



(a)



(b)

Figure 6.2: Top: Precision-recall curve for the retrieval task. The average precision values for these curves can be found in Table 6.5. The best performance was achieved in the red solid curve in which the proposed architecture was used for training and the experiment was on ego-seen exo-seen instances. Bottom: ROC curve for the retrieval task. The quantitative results measuring the area under curve can be found in Table 6.5. It can be observed that the best performance was achieved in case of ego-seen exo-seen scenario.

Table 6.5: Quantitative results in the retrieval task. The retrieval capability of the two architectures reaches its peak on ego-seen exo-seen classes. However, even on the most difficult scenario of ego-unseen exo-unseen, the performance is still meaningful and above chance.

Test Scenario	Method	AUC	Avg. Precision
Unseen-Unseen	Chance	0.500	0.071
	1S2V	0.716	0.163
	2S2V-C	0.761	0.268
	2S2V-R	0.621	0.120
	2S2V-CR	**0.780**	**0.285**
Unseen-Seen	Chance	0.500	0.071
	1S2V	0.721	0.208
	2S2V-C	0.770	0.276
	2S2V-R	0.761	0.256
	2S2V-CR	**0.805**	**0.336**
Seen-Seen	Chance	0.500	0.071
	1S2V	0.771	0.282
	2S2V-C	0.792	0.300
	2S2V-R	0.744	0.278
	2S2V-CR	**0.821**	**0.365**

6.3 Conclusions and Future Work

Inspired by the mirror neuron concept, here we explored the possibility of transferring action information across two drastically different views: egocentric (or first-person) and exocentric (or third-person). We show that it is possible to transfer knowledge from the third person perspective to the first person perspective and vice versa. Our experiments indicate that encouraging a common embedding across the two views leads to efficient action recognition over unseen classes, for which no examples have been seen during training. We also observe that action-based video retrieval is possible across the two views, even for ego-unseen exo-unseen classes. This is promising as it suggests that the learned mapping unifies the two views globally and is not limited to specific action classes. This suggests the possibility of transferring motion information across the two views.

Considering the large amount of existing data and models in the exocentric domain, we believe that following a knowledge transfer approach can be very rewarding. Future investigations could consider more sophisticated domain adaptation methods for solving the tasks investigated here. Further, learning the relationship between first and third person vision can be expanded to other problems such as human identification, action quality assessment, and tracking. Having a unified embedding can potentially lead to higher accuracy in both domains (i.e., in analogy with multi-task learning) if trained on larger scale datasets. Another interesting research direction regards exploring how low-level features such as motion (optical flow), mid and high level information such as semantic parsing, object recognition, and human pose estimation relate in the two domains.

We believe that our model and results open new directions to study first person vision and in particular its relation to third person vision. To encourage future research in this direction, we plan to share our collected dataset and code with the computer vision community.

CHAPTER 7: CONCLUSION AND FUTURE WORK

Here we highlight the concluding remarks on this thesis, and expand on potential future works in this direction of research.

7.1 Conclusion

We make the first attempt in exploring the relationship between first-person (egocentric) and third-person (exocentric) vision. We design and evaluate algorithms for addressing problems such as self-identification, re-identification, temporal alignment, action classification, action based retrieval and matching. We show that establishing the relationship between these two views is possible once each view is modeled properly and systematic comparisons is performed. In chapter 3 and 4 we explore a more analytic and hand-crafted modeling of each view. We explore the relationship across the two views using structural comparison of the two sets of identities using graph matching. We use hand crafted features extracted from the two views, represented the two views using graphs, and the comparisons were done based on spectral graph matching. In section 6 we explore a more learning based approach, allowing a convolutional neural network to learn the relationship between the two views in a supervised manner. In section 5 we explore a combination of the two approaches, utilizing both supervised CNN based visual similarity measures alongside with handcrafted features relating the two views geometrically.

In chapter 3, 4 and 5, we explore identification across egocentric and top-view videos. Each identification instance is categorized as self-identification or re-identification based on the visibility of the visual appearance of the person of interest. In chapter 5, we show that these two tasks are highly interconnected to each other and to the temporal alignment of the videos.

Inspired by the mirror neuron concept, in chapter 6 we explored the possibility of transferring action information across two views. We show that it is possible to transfer knowledge from third person perspective to first person perspective and vice versa. Our experiments indicate that encouraging a common embedding across the two views could lead to efficient action recognition over unseen classes, for which no examples have been observed by the model during training. We also observe that action-based video retrieval is possible across the two views, even for ego-unseen exo-unseen classes. This is promising as it suggests that the learned mapping unifies the two views globally and is not limited to specific action classes. Considering the large amount of existing data and models in the exocentric domain, we believe that following a knowledge transfer approach can be rewarding.

7.2 Future Work

Here we highlight some potential extensions and areas of future research that could be explored to further study the relationship between first-person and third-person images and videos. These extensions could potentially address different problems across the two views, or could address the shortcomings of the current approaches and relax the assumptions employed by our study.

7.2.1 Assumptions and Shortcomings:

In section 3, 4, and 5, we employed certain assumptions which even though are accurate in a majority of scenarios, may not hold in some cases. Future work could explore alternative representations capable of dealing with such cases. For example, we assumed that the top-view video has a top-down (90°) orientation with respect to the ground plane. In case of oblique view videos, a rectification preprocessing step would be necessary. In addition, here the egocentric cameras are assumed to have normals parallel to the ground plane. That is, our model only handles egocentric camera roll, therefore yaw and pitch of the camera are not considered. A more accurate model should take these into account when computing top-view FOVs.

To focus on the details of the cross-view assignment problem, we assumed that top-view person detections are given. We formulate the problem as assigning each egocentric video to one full trajectory in the top-view video. In case of imperfect tracks generated by an automatic tracker, our method will assign each egocentric video to one of the imperfect tracks. Performance evaluation in such a scenario is non-trivial. A potential more general definition of our problem could allow assigning more than one egocentric to one trajectory.

One of our assumptions is that the viewer's direction of motion indicates his head direction. In normal walking, head and body orientations are often aligned. This does not hold in stationary situations where a person rotates in place. A possible remedy to this is to train a head pose estimation model and use it for top-view FOV estimation.

7.2.2 Alternative approaches, problems, and setups:

A more general version of the addressed problem in chapetr 3,4, and 5 could include multiple egocentric viewers, multiple viewers without wearable cameras and multiple surveillance cameras recording overlapping or non-overlapping parts of the scene could be considered. A unified framework capable of relating egocentric and top (or oblique) view videos can potentially perform geometric reasoning for inferring more accurate identification of the egocentric viewers, and also reason about possible blind spots. Further, non-visual cues such as audio could be used to temporally align videos.

Other computer vision techniques such as visual odometry can be considered for relating the two views. Other computer vision tasks such as pose estimation, 3D reconstruction, and multi-view video summarization could also be explored across exocentric and egocentric domains using supervised and unsupervised domain adaptation techniques.

Overall, we believe bridging the gap between these two views could be further explored, and would be of benefit to many computer vision applications.

LIST OF REFERENCES

[1] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402.

[2] S. Ardeshir, A. Borji, Ego2top: Matching viewers in egocentric and top-view videos, arXiv preprint arXiv:1607.06986.

[3] R. J. Fathi A, Farhadi A, Understanding egocentric activities., Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE.

[4] R. J. Fathi A, Li Y, Learning to recognize daily actions using gaze., Computer VisionECCV.

[5] I. E. Bettadapura, Vinay, C. Pantofaru., Egocentric field-of-view localization using first-person point-of-view devices., Applications of Computer Vision (WACV), IEEE Winter Conference on.

[6] G. Rizzolatti, L. Craighero, The mirror neuron system, Annu. Rev. Neurosci. 27 (2004) 169–192.

[7] R. B. Girshick, P. F. Felzenszwalb, D. McAllester, Discriminatively trained deformable part models, release 5, http://people.cs.uchicago.edu/ rbg/latent-release5/.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[9] B. F. T. M. Bak S, Corvee E, Multiple-shot human re-identification by mean riemannian covariance grid., In Advanced Video and Signal-Based Surveillance (AVSS), 8th IEEE International Conference on.

[10] A. R. Zamir, A. Dehghan, M. Shah, GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs, in: European Conference on Computer Vision (ECCV), 2012.

[11] Y. K. Egozi, Amir, H. Guterman., A probabilistic approach to spectral graph matching., Pattern Analysis and Machine Intelligence, IEEE Transactions on.

[12] O. C. Dicle, Caglayan, M. Sznaier., The way they move: Tracking multiple targets with similar appearance., Proceedings of the IEEE International Conference on Computer Vision.

[13] T. Kanade, M. Hebert., First-person vision., Proceedings of the IEEE 100.8.

[14] R. C. R. M. Betancourt A, Morerio P, The evolution of first person vision methods: A survey., Circuits and Systems for Video Technology, IEEE Transactions on.

[15] X. R. Fathi, Alireza, J. M. Rehg., Learning to recognize objects in egocentric activities., Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On.

[16] Z. Lu, K. Grauman., Story-driven summarization for egocentric video., Computer Vision and Pattern Recognition (CVPR), IEEE Conference On.

[17] Y. Li, A. Fathi, J. Rehg, Learning to predict gaze in egocentric video, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3216–3223.

[18] P. Polatsek, W. Benesova, L. Paletta, R. Perko, Novelty-based spatiotemporal saliency detection for prediction of gaze in egocentric video.

[19] A. Borji, D. N. Sihite, L. Itti, What/where to look next? modeling top-down visual attention in complex interactive environments, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 44 (5) (2014) 523–538.

[20] C. Fan, J. Lee, M. Xu, K. Kumar Singh, Y. Jae Lee, D. J. Crandall, M. S. Ryoo, Identifying first-person camera wearers in third-person videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5125–5133.

[21] M. B. Alahi, Alexandre, M. Kunt., Object detection and matching with mobile cameras collaborating with fixed cameras., Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2.

[22] B. M. K. M. Alahi A, Marimon D, A master-slave approach for object detection and matching with fixed and mobile cameras., InImage Processing, 2008. ICIP 2008. 15th IEEE International Conference.

[23] L. D. C. M. F. Ferland F, Pomerleau F, Egocentric and exocentric teleoperation interface using real-time, 3d video projection., InHuman-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on.

[24] E. J. Park, Hyun, Y. Sheikh., Predicting primary gaze behavior using social saliency fields., Proceedings of the IEEE International Conference on Computer Vision.

[25] B. Soran, A. Farhadi, L. Shapiro, Action recognition in the presence of one egocentric and multiple static cameras, in: Asian Conference on Computer Vision, Springer, 2014, pp. 178–193.

[26] Y. Poleg, C. Arora, S. Peleg, Head motion signatures from egocentric videos, in: Asian Conference on Computer Vision, Springer, 2014, pp. 315–329.

[27] K. M. K. Yonetani, Ryo, Y. Sato., Ego-surfing first person videos., Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE,.

[28] I. G. Kiefer, Peter, M. Raubal., Where am i? investigating map matching during selflocalization with mobile eye tracking in an urban environment., Transactions in GIS 18.5.

[29] R. Yonetani, K. M. Kitani, Y. Sato, Recognizing micro-actions and reactions from paired egocentric videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2629–2638.

[30] D. Chen, Z. Yuan, B. Chen, N. Zheng, Similarity learning with spatial constraints for person re-identification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[31] S. M. B. L. M. V. Cheng DS, Cristani M, Custom pictorial structures for re-identification., BMVC.

[32] M. V. Bazzani L, Cristani M, Symmetry-driven accumulation of local features for human characterization and re-identification., omputer Vision and Image Understanding.

[33] R. Zhao, W. Oyang, X. Wang, Person re-identification by saliency learning, IEEE transactions on pattern analysis and machine intelligence 39 (2) (2017) 356–370.

[34] N. Martinel, G. L. Foresti, C. Micheloni, Person reidentification in a distributed camera network framework, IEEE transactions on cybernetics.

[35] J. García, N. Martinel, A. Gardel, I. Bravo, G. L. Foresti, C. Micheloni, Discriminant context information analysis for post-ranking person re-identification, IEEE Transactions on Image Processing 26 (4) (2017) 1650–1665.

[36] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: Deep filter pairing neural network for person re-identification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 152–159. `doi:10.1109/CVPR.2014.27`.

[37] R. R. Varior, M. Haloi, G. Wang, Gated siamese convolutional neural network architecture for human re-identification, CoRR abs/1607.08378.
URL `http://arxiv.org/abs/1607.08378`

[38] R. R. Varior, B. Shuai, J. Lu, D. Xu, G. Wang, A siamese long short-term memory architecture for human re-identification, CoRR abs/1607.08381.
URL `http://arxiv.org/abs/1607.08381`

[39] D. Yi, Z. Lei, S. Z. Li, Deep metric learning for practical person re-identification, CoRR abs/1407.4979.
URL `http://arxiv.org/abs/1407.4979`

[40] E. Ahmed, M. Jones, T. K. Marks, An improved deep learning architecture for person re-identification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[41] D. Cheng, Y. Gong, S. Zhou, J. Wang, N. Zheng, Person re-identification by multi-channel parts-based cnn with improved triplet loss function, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[42] Y.-J. Cho, K.-J. Yoon, Improving person re-identification via pose-aware multi-shot matching, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[43] T. Matsukawa, T. Okabe, E. Suzuki, Y. Sato, Hierarchical gaussian descriptor for person re-identification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[44] A. Chakraborty, B. Mandal, H. K. Galoogahi, Person re-identification using multiple first-person-views on wearable devices, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–8.

[45] K. Zheng, H. Guo, X. Fan, H. Yu, S. Wang, Identifying same persons from temporally synchronized videos taken by multiple wearable cameras.

[46] G. B.-A. Hoshen, Yedid, S. Peleg., Wisdom of the crowd in egocentric video curation., Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014.

[47] J. K. H. Fathi, Alireza, J. M. Rehg., Social interactions: A first-person perspective., Computer Vision and Pattern Recognition (CVPR), IEEE Conference on.

[48] e. a. Yan, Yan, Egocentric daily activity recognition via multitask clustering., Image Processing, IEEE Transactions on.

[49] H. O. C. A. M.-C. W. Damen D, Leelasawassuk T, You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video., BMVC.

[50] Y. Lin, K. Abdelfatah, Y. Zhou, X. Fan, H. Yu, H. Qian, S. Wang, Co-interest person detection from multiple wearable camera videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4426–4434.

[51] J. K. Aggarwal, M. S. Ryoo, Human activity analysis: A review, ACM Computing Surveys (CSUR) 43 (3) (2011) 16.

[52] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer vision and image understanding 115 (2) (2011) 224–241.

[53] C. Thurau, V. Hlavác, Pose primitive based human action recognition in videos or still images, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[54] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3169–3176.

[55] Y. Li, Z. Ye, J. M. Rehg, Delving into egocentric actions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[56] M. Ma, H. Fan, K. M. Kitani, Going deeper into first-person activity recognition, CoRR abs/1605.03688.
URL `http://arxiv.org/abs/1605.03688`

[57] K. Ogaki, K. M. Kitani, Y. Sugano, Y. Sato, Coupling eye-motion and ego-motion features for first-person activity recognition, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 1–7. `doi:10.1109/CVPRW.2012.6239188`.

[58] S. Singh, C. Arora, C. V. Jawahar, First person action recognition using deep learned descriptors, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[59] K. Matsuo, K. Yamada, S. Ueno, S. Naito, An attention-based activity recognition for egocentric video, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014.

[60] M. S. Ryoo, L. Matthies, First-person activity recognition: What are they doing to me?, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 2730–2737.

[61] Y. Poleg, A. Ephrat, S. Peleg, C. Arora, Compact CNN for indexing egocentric videos, CoRR abs/1504.07469.
URL `http://arxiv.org/abs/1504.07469`

[62] K. M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3241–3248.

[63] N. Rhinehart, K. M. Kitani, Learning action maps of large environments via first-person vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 580–588.

[64] S. Su, J. P. Hong, J. Shi, H. S. Park, Social behavior prediction from first person videos, arXiv preprint arXiv:1611.09464.

[65] N. Rhinehart, K. M. Kitani, First-person activity forecasting with online inverse reinforcement learning.

[66] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: An unsupervised approach, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 999–1006.

[67] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International Conference on Machine Learning, 2015, pp. 97–105.

[68] E. Tzeng, J. Hoffman, T. Darrell, K. Saenko, Simultaneous deep transfer across domains and tasks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4068–4076.

[69] Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash, A. Torralba, Cross-modal scene networks, arXiv preprint arXiv:1610.09003.

[70] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based image retrieval: Benchmark and bag-of-features descriptors, IEEE transactions on visualization and computer graphics 17 (11) (2011) 1624–1636.

[71] F. Wang, L. Kang, Y. Li, Sketch-based 3d shape retrieval using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1875–1883.

[72] L. Chen, W. Li, D. Xu, Recognizing rgb images by learning from rgb-d data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1418–1425.

[73] Y.-Y. Lin, J.-H. Hua, N. C. Tang, M.-H. Chen, H.-Y. Mark Liao, Depth and skeleton associated action recognition without online accessible rgb-d cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2617–2624.

[74] I. N. Junejo, E. Dexter, I. Laptev, P. Pérez, Cross-view action recognition from temporal self-similarities, in: European Conference on Computer Vision, Springer, 2008, pp. 293–306.

[75] J. Liu, M. Shah, B. Kuipers, S. Savarese, Cross-view action recognition via view knowledge transfer, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3209–3216.

[76] R. Li, T. Zickler, Discriminative virtual views for cross-view action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2855–2862.

[77] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, W. Burgard, Multimodal deep learning for robust rgb-d object recognition, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, 2015, pp. 681–687.

[78] M. Geng, Y. Wang, T. Xiang, Y. Tian, Deep transfer learning for person re-identification, arXiv preprint arXiv:1611.05244.

[79] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2066–2073.

[80] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, Unsupervised visual domain adaptation using subspace alignment, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 2960–2967.

[81] K. Muandet, D. Balduzzi, B. Schölkopf, Domain generalization via invariant feature representation, in: Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 10–18.

[82] D. Zhang, J. He, Y. Liu, L. Si, R. Lawrence, Multi-view transfer learning with a large margin approach, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 1208–1216.

[83] M.-Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Advances in neural information processing systems, 2016, pp. 469–477.

[84] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, Journal of Machine Learning Research 17 (59) (2016) 1–35.

[85] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, arXiv preprint arXiv:1702.05464.

[86] A. Torralba, Contextual priming for object detection, in: International Journal of Computer Vision, Vol. 53(2), 169-191, 2003.

[87] H. W. Kuhn, The hungarian method for the assignment problem, Naval research logistics quarterly 2 (1-2) (1955) 83–97.

[88] B. Fulkerson, A. Vedaldi, S. Soatto, Class segmentation and object localization with super-pixel neighborhoods, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 670–677.

[89] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[90] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in neural information processing systems, 1990, pp. 396–404.

[91] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, arXiv preprint arXiv:1409.7495.