Identifying First-person Camera Wearers in Third-person Videos

Chenyou Fan[1], Jangwon Lee[1], Mingze Xu[1], Krishna Kumar Singh[2], Yong Jae Lee[2],
David J. Crandall[1] and Michael S. Ryoo[1]
[1]Indiana University Bloomington
[2]University of California, Davis
{fan6,mryoo}@indiana.edu

Abstract

We consider scenarios in which we wish to perform joint scene understanding, object tracking, activity recognition, and other tasks in environments in which multiple people are wearing body-worn cameras while a third-person static camera also captures the scene. To do this, we need to establish person-level correspondences across first- and third-person videos, which is challenging because the camera wearer is not visible from his/her own egocentric video, preventing the use of direct feature matching. In this paper, we propose a new semi-Siamese Convolutional Neural Network architecture to address this novel challenge. We formulate the problem as learning a joint embedding space for first- and third-person videos that considers both spatial- and motion-domain cues. A new triplet loss function is designed to minimize the distance between correct first- and third-person matches while maximizing the distance between incorrect ones. This end-to-end approach performs significantly better than several baselines, in part by learning the first- and third-person features optimized for matching jointly with the distance measure itself.

1. Introduction

Wearable cameras are becoming mainstream: GoPro and other first-person cameras are used by consumers to record extreme sports and other activities, for example, while body-worn cameras are now standard equipment for many police and military personnel [8]. These cameras capture unique perspectives that complement video data from traditional third-person cameras. For instance, in a complex and highly dynamic environment like a busy city street or a battlefield, third-person cameras give a global view of the high-level appearance and events in a scene, while first-person cameras capture ground-level evidence about objects and people at a much finer level of granularity. The combination of video from these highly complementary views could be used to perform a variety of vision tasks – scene understanding, object tracking, activity recognition, etc. – with greater fidelity and detail than either could alone.

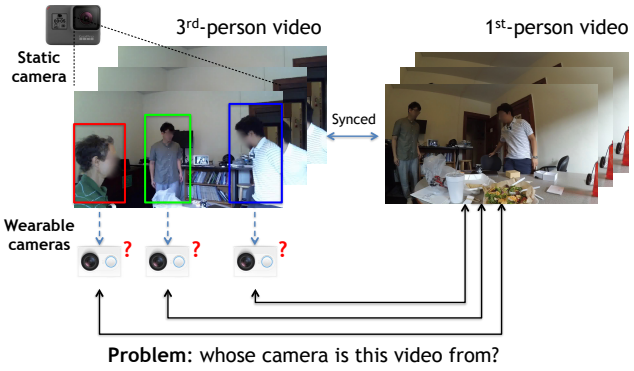


Figure 1. One or more people wear first-person cameras in a scene that is also recorded by a third-person camera. We wish to identify which person in the third-person view (left) was wearing the camera that captured a first-person video (right). This is challenging because the camera fields of view are very different and the camera wearer almost never appears in their own first-person view.

In these scenarios, multiple people may be in a scene at any given time, with people regularly entering and exiting the view of the third-person camera. Some subset of these people may be wearing first-person cameras, each of which is also capturing part of the scene but from a highly dynamic point of view that changes as the wearer moves. Thus at any moment in time, some (possibly empty) subset of people appear in any given camera's view, and each person appears in some (possibly empty) subset of the first- and third-person cameras (and that person themselves may be wearing one of the first-person cameras). Compared to static cameras, first-person video data is significantly more challenging because of camera motion, poor scene composition, challenging illumination, etc.

Jointly solving computer vision problems across multiple first- and third-person cameras requires the crucial first step of establishing correspondences between the people and the cameras, including (1) identifying the same person

appearing in different views, as well as (2) matching a camera wearer in one view with their corresponding first-person video. The former problem is similar to person identification and re-identification problems that have been studied for third-person cameras [10]. These approaches typically rely on matching visual and motion features of a person across different views; the first-person camera version is similar in principle but significantly more difficult due to the difference in perspectives and characteristics of first- and third-person video.

The second problem is even more challenging, since a person's *appearance* in one video may share few (if any) visual features with his or her *first-person visual field* of the same scene. For instance, a surveillance camera might capture a camera wearer walking down the street, including her physical appearance, the cars parked on the street beside her, and the friends walking next to her, while her front-facing first-person camera may capture *none* of these because its field of view is down the street. Finding correspondences thus cannot rely on direct appearance feature matching. Instead, we must rely on indirect sources of evidence for finding correspondences: (i) matching first-person videos against an estimate of what a person's field of view *would look like* based on the third-person view of the scene; (ii) matching estimates of a person's body movements based on the camera motion of their first-person video to the movements observed by the third-person static camera; and (iii) matching the (rare) moments when part of a camera wearer's body or actions are directly visible in the scene (e.g. when reaching for an object and both first- and third-person cameras see the hand). See Figure 1.

Despite its importance, we are aware of very little work that tries to address this problem. Several recent papers propose using multiple cameras for joint first-person recognition [3, 5, 26, 29], but make simplistic assumptions like that only one person appears in the scene. Using visual SLAM to infer first-person camera trajectory and map to third-person cameras (e.g., [17, 19]) works well in some settings, but can fail for crowded environments when long-term precise localizations are needed and when first-person video has significant motion blur. Ardeshir and Borji [2] match a set of egocentric videos to people appearing in a top-view video using graph matching, but assume that there are multiple first-person cameras sharing the same field of view at any given time, and only consider purely overhead third-person cameras (not oblique or ground-level views). We require a more general approach that matches each individual first-person video with the corresponding person appearing in an arbitrarily-positioned third-person camera.

In this paper, we present a new semi-Siamese Convolutional Neural Network (CNN) framework to learn the distance metric between first- and third-person videos. The idea is to learn a joint embedding space between first- and third-person perspectives, enabling us to compute the similarity between *any given first-person video* and *an individual human appearing in a third-person video.* Our new semi-Siamese design allows for learning low-level features specialized for first-person videos and for third-person videos separately, while sharing higher-level representations and an embedding space to permit a distance measure. Evidence from both scene appearance and motion information is jointly considered in a novel two-stream semi-Siamese CNN. Finally, we introduce a new "triplet" loss function for our semi-Siamese network, and confirm its advantages in our experiments on a realistic dataset.

2. Related work

While many of the core problems and challenges of recognition in first-person (egocentric) videos are shared with traditional third-person tasks, first-person video tends to be much more challenging, with highly dynamic camera motion and difficult imaging conditions. Research has focused on extracting features customized for first-person video, including hand [14], gaze [16], and ego-motion cues [20]. Other work has studied object-based understanding for activity recognition [18], video summarization [15, 30], and recognition of ego-actions [13] and interactions [22], but in single first-person videos.

Several recent papers have shown the potential for combining first-person video analysis with evidence from other types of synchronized video, including from other first-person cameras [3, 29], multiple third-person cameras [26], or even hand-mounted cameras [5]. However, these papers assume that a single person appears in each video, avoiding the person-level correspondence problem. Our work is complementary, and could help generalize these approaches to scenarios in which multiple people appear in a scene.

A conventional approach to our person correspondence problem might use visual odometry and other camera localization techniques [7, 23] to estimate the 3-d trajectory of the wearable camera, which could then be projected onto the static camera's coordinate system to identify the camera wearer [17]. However, this is problematic in crowded or indoor environments where accurate localization is difficult and people are standing close together. Precise online visual localization in indoor environments with few landmarks is itself challenging, and not applicable when cameras are not calibrated or move too quickly and cause motion blur.

Perhaps the work most related to ours is that of Ardeshir and Borji [2], which matches a set of egocentric videos to a set of individuals in a top-view video using graph-based analysis. This technique works well but makes two significant assumptions that limit its real-world applicability. First, it requires the static camera to have a strictly top-down (directly overhead) view, which is relatively uncommon in the real world (e.g. wall-mounted surveillance cameras cap-

ture oblique views). Second, it assumes that multiple egocentric videos sharing the same field-of-view are available. This assumption is strong even if there are multiple people wearing cameras: the cameras may not share any field of view due to relative pose or occlusions, and even if multiple first-person videos with overlapping fields of view are recorded, some users may choose not to share them due to privacy concerns, for example. In contrast, we consider the more challenging problem of matching each of multiple first-person cameras having arbitrary fields of view with a static, arbitrarily-mounted third-person camera.

We believe this is the first paper to formulate first- and third-person video correspondence as an embedding space learning problem and to present an end-to-end learning approach. Unlike previous work [19, 28] which uses hand-coded trajectory features to match videos without any embedding learning, our method is applicable in more complex environments (e.g. with arbitrarily placed first- and third-person cameras and arbitrary numbers of people).

3. Our approach

Given one or more first-person videos, our goal is to decide if each of the people appearing in a third-person video is the wearer of one of the first-person cameras. The key idea is that despite having very different characteristics, synchronized first- and third-person videos are different perspectives on the same general environment, and thus capture some of the same people, objects, and background (albeit from two very different perspectives). This overlap may allow us to find similarities in *spatial-domain (visual) features*, while hopefully ignoring differences due to perspective. Meanwhile, corresponding first- and third-person videos are also two reflections of the same person performing the same activity, which may allow us to find *motion-domain feature* correspondences between video types.

We formulate this problem in terms of learning embedding spaces shared by first- and third-person videos. Ideally, these embeddings minimize the distance between the first-person video features observed by a camera wearer and the visual features of the same person observed by a static third-person camera at the same moment, while maximizing the distances between incorrect matches. We propose a new semi-Siamese network architecture, detailed in the next section, to learn this embedding space. To handle the two modalities (motion and spatial-domain), we design a new two-stream Siamese CNN architecture where one stream captures temporal information using optical flow (i.e., motion) and the other captures spatial information (i.e., surrounding scene appearance), which we detail in Section 3.1.3. We also consider two loss functions: a traditional contrastive loss that considers pairs of samples, and a new triplet loss that takes advantage of the fact that both positive and negative first-to-third-person pairings exist in the same scene. We describe these losses in Section 3.2.

3.1. Semi-siamese networks

Our approach is based on Siamese networks with contrastive loss functions, which enable end-to-end learning of both low-level visual features and an embedding space (jointly optimizing them based on training data). The original Siamese formulation [11] interprets the network as a function $f(I;\theta)$ that maps each input video I into an embedded point using parameters θ, which are typically trained based on contrastive loss between embedding of the positive and negative examples [4, 24]. If we applied this approach to our problem, I would be either the first-person or third-person video, such that the network (i.e., function f and parameters θ) would be shared by both types of videos.

However, first- and third-person videos are very different, even when recording the same event by the same person in the same location. We hypothesize that although the higher-level representations that capture object- and action-level information in first- and third-person videos might be shared, the optimal low-level features (i.e., early convolutional filters) may not be identical.

We thus propose a *semi-Siamese* architecture to learn the first- to third-person distance metric. We find separate parameters for first- and third-person videos, which we call θ_1 and θ_2, respectively, while forcing them to share a subset of parameters θ. Given a set E of egocentric cameras and a set P of detected people in a third-person camera view, we can easily estimate the person corresponding to a given egocentric camera $e \in E$ using this embedding space,

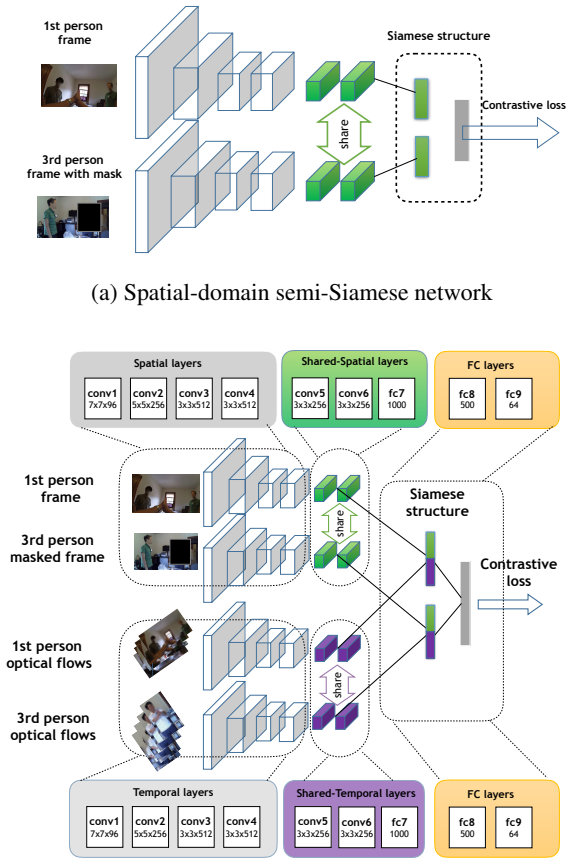
$$p_e^* = \arg\min_{p \in P} ||f(I_e;\theta_1,\theta) - f(I_p;\theta_2,\theta)||. \quad (1)$$

We now propose specific network architectures, first considering the two feature modalities (spatial-domain and motion-domain) independently, and then showing how to combine them into integrated two-stream networks.

3.1.1 Spatial-domain network

To learn the spatial-domain correspondences between first- and third-person cameras, our network receives a single frame of first-person video and the corresponding frame of third-person video (Figure 2(a)). For the third-person video, we force the network to consider one specific person by masking him or her out from the rest of the frame, replacing their bounding box with black pixels. This is important because a camera wearer does not appear in their own first-person video (with the occasional exception of arms or hands). We thus encourage the network to learn a relationship between a first- and third-person video frame, with that person *removed from* the third-person scene.

As shown in Figure 2(a), each of the first- and third-person branches maintains its own four early convolution



(a) Spatial-domain semi-Siamese network

(b) Motion-domain semi-Siamese network

(c) Two-stream semi-Siamese network

(d) Two-stream semi-triplet network

Figure 2. *Overview of our networks.* All networks receive features from time-synchronized first- and third-person video frames, which during training consist of correct correspondences as positive examples and incorrect correspondences as negative examples. **(a) Spatial-domain network** is a semi-Siamese network with separate early convolutional layers (gray) and shared later layers (green). Corresponding input pairs consist of a first-person video frame and third-person frame with the corresponding person (camera wearer) masked out, since he or she is not visible in his or her own first-person camera. **(b) Motion-domain network** is also semi-Siamese with a similar structure, except that it inputs stacked optical flow fields instead of images frames, and the third-person flow field consists of a crop of a single person. **(c) Two-stream semi-Siamese network** combines both networks with a fully-connected layer that produces the final feature vector. **(d) Two-stream semi-Siamese network trained with triplet loss** receives three inputs during training: a first-person frame, the corresponding third-person frame with the correct person masked out, and the same third-person frame with a random incorrect person masked out.

layers while sharing the last two convolution layers and fully connected layers. The intuition here is that while we need to capture the same high-level semantic information from each video, the low-level features corresponding to those semantics may differ significantly. The last fully-connected layer abstracts spatial-domain information from the two perspectives as two D-dimensional feature vectors. To train the network, we present known true and false correspondences, and use a contrastive loss function that minimizes sum-of-squares between feature vectors of true pairs and a hinge loss that examines if the distance is greater than a margin for negative pairs (detailed below).

3.1.2 Motion-domain network

Figure 2(b) shows the motion-domain network, which learns correspondences between motion in a first-person video *induced by* the camera wearer's movements and their *directly visible* movements in a third-person video. The idea is that (1) body movements of the camera wearer (e.g., walking) will be reflected in both first- and third-person videos and that (2) hand motion may also be captured in both cameras during gestures or actions (e.g., drinking coffee). We first compute optical flow for each video, and then stack the flow fields for sets of five consecutive frames as input to the network. The first-person input is the entire op-

tical flow field, whereas the third-person input is the flow field *cropped around a single person.* This differs from the input to the spatial-domain network: here we encourage correspondences between the motion of a camera wearer and their motion as seen by a third-person camera, whereas with the spatial network we encouraged correspondences between the first-person scene and the third-person scene *except for* the camera wearer.

3.1.3 Two-stream networks

To combine evidence from both spatial- and motion-domain features, we use a two-stream network, as shown in Figure 2(c). Like the spatial-domain network described above, the spatial stream receives pairs of corresponding first-person and masked third-person frames, while the temporal stream receives pairs of corresponding first-person and cropped third-person stacked flow fields. Within each stream, the final two convolution layers and fully-connected layers are shared, and then two final fully-connected layers and a contrastive loss combine the two streams. This design was inspired by Simonyan and Zisserman [25], although that network was proposed for a completely different problem (activity recognition with a single static camera) and so was significantly simpler, taking a single frame and corresponding stack of optical flow fields. In contrast, our network features two semi-Siamese streams, two shared fully connected layers, and a final contrastive loss.

3.2. Loss functions

We propose two loss functions for learning the distance metric: a standard contrastive loss, and a new "triplet" loss that considers pairs of correct and incorrect matches.

Contrastive loss: For the Siamese or semi-Siamese networks, we want first- and third-person frame representations generated by the CNN to be close only if they correspond to the same person. For a batch of B training exemplars, let x_e^i be the first-person visual feature corresponding to the i-th exemplar, x_p^i refer to the third-person visual feature of the i-th exemplar, and y^i be an indicator that is 1 if the exemplar is a correct correspondence and 0 otherwise. We define the contrastive loss to be a Euclidean distance for positive exemplars and a hinge loss for negative ones,

$$L_{siam}(\theta) = \sum_{i}^{B} y_i ||x_e^i - x_p^i||^2 + (1 - y_i) \max(m - ||x_e^i - x_p^i||, 0)^2 \quad (2)$$

where m is a predefined constant margin.

Triplet loss: At training time, given a third-person video with multiple people and a first-person video, we know which pairing is correct and which pairings are not. As an alternative to treating pairs independently as with the contrastive loss, we propose to form triplet exemplars consisting of both a positive and negative match. The triplet loss encourages a metric such that the distance from the first-person frame to the correct third-person frame is low, but to the incorrect third-person frame is high. More precisely, for a batch of B training examples, the i-th exemplar is a triple (x_e^i, x_1^i, x_0^i) corresponding to the features of the first-person frame, the correct third-person frame, and the incorrectly-masked third-person frame, respectively. Each exemplar thus has a positive pair (x_e^i, x_1^i) and a negative pair (x_e^i, x_0^i), and we want to minimize the distance between the true pair while ensuring the distance between the false pair is larger. We use a hinge loss to penalize if this condition is violated,

$$L_{trip} = \sum_{i}^{B} ||x_e^i - x_1^i||^2 + \max(0, m^2 - (||x_e^i - x_0^i||^2 - ||x_e^i - x_1^i||^2)) \quad (3)$$

where m is a constant. This loss is similar to the Siamese contrastive loss function, but explicitly enforces the distance difference to be larger than a margin. Our loss can be viewed as a hybrid between Schroff *et al*. [24] and Bell and Bala [4]: like [4], we explicitly minimize the distance between the positive pair, and like [24], we maximize the *difference in distance* between the negative and positive pairs.

Figure 2(d) shows the two-stream semi-Siamese network with a triplet loss function. During training, the spatial stream of the network expects a first-person frame, a corresponding masked third-person frame, and an incorrect masked third-person frame, while the temporal stream expects a first- and two third-person cropped optical flow stacks, with the third-person inputs sharing all layers and the first- and third-person layers separate.

4. Experiments

We evaluated our proposed technique to identify people appearing in third-person video and their corresponding first-person videos, comparing our various network architectures, feature types, and loss functions against baselines.

4.1. Data

Groups of three to four participants were asked to perform everyday activities in six indoor environments while two wore first-person video cameras. Each environment was also equipped with a static camera that captured third-person video of the room, typically from a perspective a bit above the participants' heads. We did not give specific

instructions but simply asked participants to perform everyday, unstructured activities and interactions, such as shaking hands, writing on a whiteboard, drinking, chatting, eating, etc. The first-person videos thus captured not only objects, participants, and background, but also motion of other people in the scene and ego-motion from hand and body movements. Participants were free to walk around the room and so regularly entered and exited the cameras' fields-of-view.

We collected seven sets of three synchronized videos (two first- and one third-person) ranging between 5-10 minutes. Three sets had three participants and four included four. All videos were recorded at HD resolution at 30fps, using Xiaoyi Yi Action Cameras [1] for the first-person video and a Macbook Pro webcam for the third-person video. After collecting the videos, we subsampled to 5fps to yield 11,225 frames in total. We created ground truth by manually drawing bounding boxes around each person in each frame and giving each box a unique person ID, generating a total of 14,394 bounding boxes across 4,680 frames.

Because contiguous frames are typically highly correlated, we split training and test sets at the video level, with five videos for training (3,622 frames) and two for testing (1,058 frames). Since there are usually multiple people per third-person frame, most frames generate multiple examples of correct and incorrect person pairs (totaling 3,489 positive and 7,399 negative pairs for training, and 1,051 positive and 2,455 negative pairs for testing). Training and test sets have videos of different scenes and actors.

4.2. Evaluation and training setting

We use two different metrics for measuring accuracy on the person correspondence task. In the first measure, we formulate the problem in terms of binary classification, asking whether a given person in a third-person frame corresponds with a given first-person frame or not, and then applying this classifier on all possible pairs in each frame. In this setting, a given first-person video may not correspond to any of the people in the third-person frame (if the person is out of the camera's field of view), in which case the system should reject all candidate pairs. In the second measure, we formulate the task as the multi-class classification problem of assigning a given first-person video to a corresponding person in the third-person scene. For instance, if there are four people appearing in the third-person camera, the goal is to choose the one corresponding to the first-person video, making this a four-way classification task.

We implemented our networks in Caffe [12] with stochastic gradient descent with fixed learning rate 10^{-5}, momentum 0.9 and weight decay 0.0005 for 50,000 iterations, using three NVidia Titan X GPUs. This required about six hours for the spatial network and one day for the temporal and two-stream networks. We have released data and code online.[1] As described above, during training we feed our networks with first-person frames and flow fields, and corresponding positive and negative cropped flow fields (for the motion networks) and masked images (for the spatial networks). During testing, we use our ground-truth bounding boxes to "highlight" a person of interest in the third-person view by masking them out for the spatial network and cropping them out for the motion networks.

4.3. Baselines

We implemented multiple baselines to confirm the effectiveness of our approach. These included mapping of optical flow features from first-person to third-person view, direct matching of pre-trained CNN features, and learning an embedding space with traditional HOOF features.

Flow magnitude to magnitude calculates the mean magnitude of the optical flow vectors on each corresponding first- and third-person frame, and then learns a linear regressor relating the two. Intuitively, at any moment in time there should be a correlation between the "quantity" of motion in a person's first-person view and that of their corresponding appearance in a third-person view. *HOOF to HOOF* divides the flow field of an image into a 3×3 grid, and then computes 5-bin Histogram of Optical Flow (HOOF) features [6] for each cell. We stack these 9 histograms to give a 45-d histogram per frame, and then average the histograms over a 10-frame temporal window to give a final 45-d feature vector. We then learn a linear regressor relating the corresponding first-person and third-person HOOFs. *Odometry to HOOF* estimates camera trajectories through visual odometry for each first-person video. We use LibVISO2 [9] to estimate a 13-d pose and velocity vector encoding 3-d position, 4-d orientation as a quaternion, and angular and linear velocity in each axis for each first-person frame, and then learn a regressor to predict the HOOF features in the third-person video. *Velocity to flow magnitude* learns a regressor between just the 3-d XYZ velocity vector computed by LibVISO2 for the third-person frame and the mean flow magnitude in the first-person frame.

In addition to the above basic baselines, we tested two types of stronger baselines: (1) directly comparing standard video CNN features (*two-stream* [25] and *C3D* [27]) from first- and third-person videos, and (2) learning an *embedding* space with traditional HOOF (or motion magnitude). In particular, the latter baselines have exactly the same loss function as ours by using fully connected layers. Finally, we implemented Poleg *et al.*'s *head motion signatures* [19], which track bounding boxes of people in third-person frames and correlates them with average XY optical flows in first-person frames.

[1] http://vision.soic.indiana.edu/identifying-1st-3rd

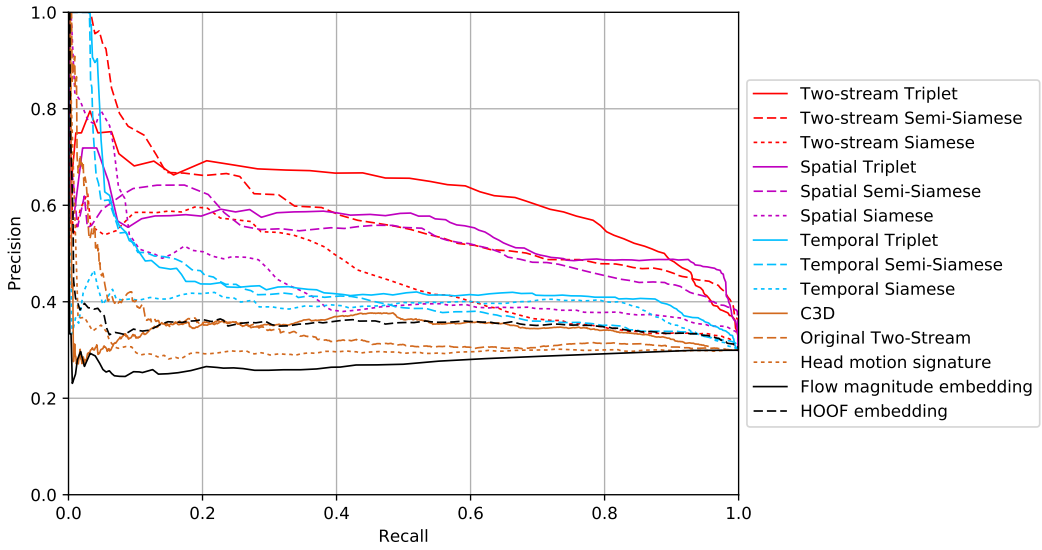


Figure 3. *Precision-recall curves* for baselines and variants of our proposed approach.

4.4. Results

Figure 3 presents precision-recall curves for variants of our technique and the baselines, and Table 1 summarizes in terms of Average Precision (AP). The figure views our task in terms of retrieval, which is our first measure: for each frame, we generate the set of all possible candidate pairings consisting of a person in the third-person view and one of the first-person views, and ask the system to return the correct matches (potentially none). The figure shows that for all feature types, our proposed semi-Siamese architecture outperforms Siamese networks, suggesting that first- and third-person perspectives are different enough that early layers of the CNNs should be allowed to create specialized low-level features. Switching to the triplet loss yields a further performance boost compared to the traditional contrastive loss; for the two-stream network, for example, it increases from an average precision of 0.585 to 0.621.

Across the different feature types, we find that the spatial-domain networks perform significantly better than the temporal (motion)-domain network (e.g., average precisions of 0.549 vs 0.456 for triplet semi-Siamese). The temporal networks still significantly outperform a random baseline (about 0.452 vs 0.354), indicating that motion features contain useful information for matching between views. The two-stream network that incorporates both types of features yields a further significant improvement (0.621).

Table 1 clearly indicates that our approach of learning the shared embedding space for first- and third-person videos significantly outperforms the baselines. Unlike previous work relying on classic hand-crafted features like head trajectories (e.g., [19]), our method learns the optimal embedding representation from training data in an end-to-end fashion, yielding a major increase in accuracy. We also compared our Siamese and semi-Siamese architectures against the model of not sharing any layers between first-person and third-person branches (*Not-Siamese* in Table 1), showing that semi-Siamese yields better accuracy.

Multi-class classification: Table 1 also presents accuracies under our second evaluation metric, which views the problem as multi-way classification (with the goal to assign a given first-person video to the correct person in the third-person scene; e.g., if there are four people in the third-person video, the goal is to choose the one corresponding to the first-person video). We see the same pattern as with average precision: semi-Siamese works better than Siamese, triplet loss outperforms contrastive, the two-stream networks outperform the single-feature networks, and all of the baselines underperform. Our proposed two-stream semi-Siamese network trained with a triplet loss yields the best accuracy, at about 69.3% correct classification.

Multiple wearable cameras: Although we have focused on static third-person cameras, our approach is applicable to any setting where there are at least two cameras, one from an actor's viewpoint and another observing the actor (including multiple wearable cameras). To test this, we also tested a scenario in which video from one wearable camera is treated as first-person while video from the other (wearable) camera is treated as third-person. These videos seldom have any spatial overlap in their views, and we made our approaches and the baselines to rely only on temporal information for the matching. Table 2 illustrates the results, showing that our approach outperforms baselines.

Network setting		*Evaluation*	
Type	Method	Binary AP	Multi Accuracy
Baselines	Flow magnitude to magnitude	0.285	0.250
	HOOF to HOOF	0.316	0.336
	Odometry to HOOF	0.302	0.493
	Velocity to flow magnitude	0.279	0.216
	HOOF embedding	0.354	0.388
	Magnitude embedding	0.276	0.216
	Head Motion Signature [19]	0.300	0.290
	Original Two-stream [25]	0.350	0.460
	C3D [27]	0.334	0.505
Spatial	Siamese	0.481	0.536
	Semi-Siamese	0.528	0.585
	Triplet	0.549	0.588
Temporal	Siamese	0.337	0.372
	Semi-Siamese	0.389	0.445
	Triplet	0.452	0.490
Two-Stream	Siamese	0.453	0.491
	Not-Siamese	0.476	0.554
	Semi-Siamese	0.585	0.639
	Triplet	**0.621**	**0.693**

Table 1. Evaluation in terms of average precision and multi-way classification for baselines and variants of our approach.

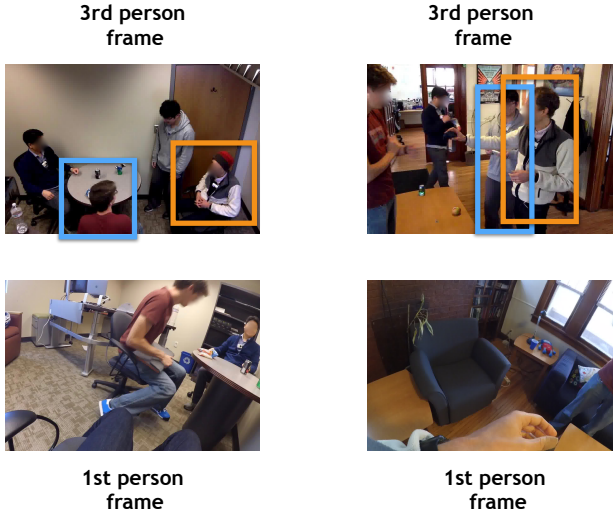
Network setting		*Evaluation*	
Type	Method	Binary AP	Multi Accuracy
Baselines	Flow magnitude to magnitude	0.389	0.442
	HOOF to HOOF	0.382	0.365
	Odometry to HOOF	0.181	0.077
	Velocity to flow magnitude	0.310	0.327
	HOOF embedding	0.405	0.365
	Magnitude embedding	0.406	0.442
	Head Motion Signature [19]	0.359	0.462
	C3D [27]	0.380	0.327
	Two-stream [25] (temporal part)	0.336	0.365
Ours	Temporal Semi-Siamese	**0.412**	**0.500**
	Temporal Triplet	0.386	**0.500**

Table 2. Results for multiple wearable camera experiments.

4.5. Discussion

Generality: Our approach is designed not to rely on long-term tracking and is thus suitable for crowded scenes. Our matching is applicable as long as we have a short tracklet of the corresponding person detected in the third-person video (e.g., only 1 frame in our spatial network), to check whether the match score is above the threshold.

Failure cases: We observed two typical failure cases. The first arises when the actual first-person camera wearer happens to have very similar motion to another person in the third-person video. Figure 4(a) shows such a situation. Our analysis of optical flows of the people suggests that the person in blue was in the process of sitting down, while the camera wearer in orange was nodding his head, creating confusingly similar flow fields (strong magnitudes in the vertical direction). Another common failure occurs when the camera wearer is heavily occluded by another person in the third-person video, such as in Figure 4(b).



(a) Motion failure case (b) Spatial failure case

Figure 4. *Sample failures,* with the person whose camera took the bottom frame in orange and our incorrect estimate in blue.

Gaze: In addition to our approach of presenting the spatial-domain network with person regions masked out, we also tried explicitly estimating gaze of people appearing in third-person videos. The idea was to encourage the spatial network to focus on the region a person is looking at, and then match it with first-person videos. We tried Recasens *et al.* [21] for gaze estimation, but this provided noisy estimates which harmed the matching ability of our network.

5. Conclusion

We presented a new Convolutional Neural Network framework to learn distance metrics between first- and third-person videos. We found that a combination of three innovations achieved the best results: (1) a semi-Siamese structure, which takes into account different features of first- and third-person videos (as opposed to full Siamese), (2) a two-stream CNN structure which combines spatial and motion cues (as opposed to a single stream), and (3) a triplet loss which explicitly enlarges the margin between first- and third-person videos (as opposed to Siamese contrastive loss). We hope this paper inspires more work in this important problem of finding correspondences between multiple first- and third-person cameras.

Acknowledgements: This work was supported in part by NSF (CAREER IIS-1253549) and the IU Office of the Vice Provost for Research, the College of Arts and Sciences, and the School of Informatics and Computing through the Emerging Areas of Research Project "Learning: Brains, Machines, and Children." CF was supported by a Paul Purdom Fellowship.

References

[1] Xiaoyi Yi Action Camera. http://www.xiaoyi.com/en/specs_en.html.

[2] S. Ardeshir and A. Borji. Ego2top: Matching viewers in egocentric and top-view cameras. In *European Conference on Computer Vision (ECCV)*, 2016.

[3] S. Bambach, D. Crandall, and C. Yu. Viewpoint integration for hand-based recognition of social interactions from a first-person view. In *ACM International Conference on Multimodal Interaction (ICMI)*, 2015.

[4] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. In *ACM Transactions on Graphics (SIGGRAPH)*, 2015.

[5] C.-S. Chan, S.-Z. Chen, P.-X. Xie, C.-C. Chang, and M. Sun. Recognition from hand cameras: A revisit with deep learning. In *European Conference on Computer Vision (ECCV)*, 2016.

[6] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human activities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM). *IEEE Robotics & Automation Magazine*, 2006.

[8] M. Funk. Should we see everything a cop sees? *The New York Times*, Oct 18, 2016.

[9] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[10] S. Gong, M. Cristani, S. Yan, and C. C. Loy. *Person Re-Identification*. Springer, 2014.

[11] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia (MM)*, 2014.

[13] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[14] S. Lee, S. Bambach, D. J. Crandall, J. M. Franchak, and C. Yu. This hand is my hand: A probabilistic approach to hand disambiguation in egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.

[15] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[17] H. S. Park, E. Jain, and Y. Sheikh. Predicting primary gaze behavior using social saliency fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[18] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Y. Poleg, C. Arora, and S. Peleg. Head motion signatures from egocentric videos. In *Asian Conference on Computer Vision (ACCV)*, 2014.

[20] Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[21] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[22] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[23] D. Scaramuzza and F. Fraundorfer. Visual odometry. *IEEE Robotics & Automation Magazine*, 2011.

[24] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[25] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014.

[26] B. Soran, A. Farhadi, and L. Shapiro. Action recognition in the presence of one egocentric and multiple static cameras. In *Asian Conference on Computer Vision (ACCV)*, 2015.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[28] R. Yonetani, K. M. Kitani, and Y. Sato. Ego-surfing first-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[29] R. Yonetani, K. M. Kitani, and Y. Sato. Recognizing micro-actions and reactions from paired egocentric videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] J. Zheng, Z. Jiang, P. J. Phillips, and R. Chellappa. Cross-view action recognition via a transferable dictionary pair. In *British Machine Vision Conference (BMVC)*, 2012.