Towards Automated Spatio-Temporal Trajectory Recovery in Wide-Area Camera Networks

Meng Zheng, *Student Member, IEEE,* Srikrishna Karanam, *Member, IEEE,*
and Richard J. Radke, *Senior Member, IEEE*

Abstract—Much recent research in person re-identification has focused on improving the accuracy of matching query images from one camera view to candidates from another camera view. However, in a practical scenario, real-world surveillance system operators often must continually re-identify a person of interest through multiple views in a wide-area camera network, spatially and temporally "following" the person's trajectory. This aspect is substantially different from a traditional re-id algorithm that can only tell whether two images belong to the same person. To address this gap, we present a new algorithm to automatically reconstruct the time-stamped spatial trajectory of a person of interest moving in a camera network. With this output, a surveillance system user can easily tell where in the camera network the person of interest was located at any specific time. Since existing datasets lack the kind of annotated data needed to address this problem, we present a new dataset, RPIfield, which includes extensive trajectory annotations. We then present a novel algorithm with topology-informed transition time modeling and candidate space pruning strategies that lead to efficient trajectory reconstruction. To evaluate our method, we introduce three new evaluation metrics directly informed by practical system-level considerations and conduct extensive experiments on RPIfield.

Index Terms—spatio-temporal trajectory recovery, surveillance systems, camera network, video analytics.

1 Introduction

PERSON re-identification (re-id) has attracted much recent interest due to its broad application in forensic analysis and surveillance systems [1–6]. The re-id problem is defined as follows: given a query image from one camera view (the *probe*), find the correct match(es) from candidate images captured from another camera view (the *gallery*). The output of existing re-id algorithms is typically a list of candidate images ranked by the corresponding similarity scores computed by the algorithm. However, for real-world applications of re-id, this is not enough. For example, a suspect typically will need to be consistently tracked throughout a wide-area camera network, not just a single camera, so that the user of the system (e.g., a security officer) can monitor and recover the spatial and temporal information about his/her travel path. To achieve this with conventional re-id algorithms, system users would have to manually, incrementally select correct matches from the camera-to-camera ranked lists produced by the algorithm and then reconstruct the probe's path based on the selected appearances. With the list of candidate images likely to be large in dense mass-transit environments (e.g., airports) where such systems are typically installed, the burden on the system operator to manually go through even a part of the list and recover the travel path can be immense. To address this key practical necessity, we introduce spatio-temporal trajectory recovery, the problem of automatically reconstructing the time-stamped spatial trajectory of a person of interest in a camera network. We illustrate the problem and its difference to re-id in Figure 1. Given a probe, a traditional re-id algorithm will produce ranked candidate lists for each

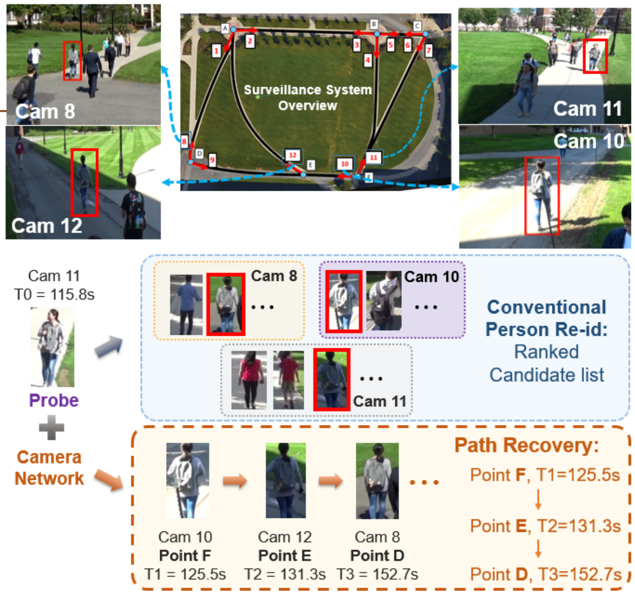


Fig. 1: An illustration of the trajectory recovery problem in a wide-area multi-camera system, compared to conventional person re-identification.

- *M. Zheng and S. Karanam are with United Imaging Intelligence, Cambridge, MA 02140 USA (e-mail: mengzhengrpi@gmail.com, srikrishna@ieee.org).*
- *R.J. Radke is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (rjradke@ecse.rpi.edu).*

- *Corresponding author: M. Zheng. This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.*

camera in the network. On the other hand, our proposed trajectory recovery involves a complete spatial and temporal path reconstruction of the probe as s/he moves through the camera network. Specifically, as shown in Figure 1, the output is a time-stamped sequential list, including camera identity and image sequence, that shows where the person of interest was in the camera network and at what time. End-users of a surveillance system can then use this data to easily retrieve the desired forensic information for the person of interest.

To address the trajectory recovery problem under the considerations discussed above, we need datasets that have explicit spatio-temporal annotations, i.e., timestamps and spatial locations for each camera of each person's appearances. Standard benchmark re-id datasets such as VIPeR [8], iLIDSVID [9], and MARS [10] lack this information, rendering them inappropriate for our use here. While the timestamp information in DukeMTMC-reID [11, 12] can possibly be re-purposed for this problem, manually associating massive amounts of reappearances is suboptimal. Furthermore, it is difficult, and even impossible in some cases, to construct continuous trajectories for many people in the dataset since they may enter/leave the camera network at any point in time, breaking the trajectory. To address these issues, we introduce a new publicly available multi-camera dataset, named RPIfield [7], that includes spatio-temporally annotated image sequences of reappearances of 112 "actors" who were explicitly asked to walk along designated paths in the camera network. We note that to our knowledge, this is the only re-id dataset collected with planned pedestrian trajectories vs. after-the-fact probes created from existing surveillance footage.

Given an appropriate dataset, we next address the algorithmic problem of trajectory recovery. Starting from the probe's first appearance and the known camera topology information, the incremental determination of the reappearances of the probe, and hence the spatio-temporal path, is essentially a search problem. We present an illustration in Fig. 2, we see an exponential growth in the search tree as the reconstruction step increases. For the purposes of computational efficiency and practical use, we need an appropriate search space pruning strategy to recover the probe's most likely path. To this end, we present a dynamic pruning strategy based on each tree node's associated confidence score. While the easiest approach would be to simply take the highest-scoring match in the collection of neighbor cameras at each time step, correct matches to the probe rarely appear at rank 1 due to distractors and/or occlusions. To address this issue, we propose to learn inter-camera temporal transition models (i.e., the time taken to move from one camera view to the other) to efficiently prune out unlikely candidates in our search space. Specifically, we learn transition time statistics and patterns between pairs of adjacent cameras based on reappearances of the same identities and the corresponding (given) timestamp information. Our RPIfield dataset provides exactly the kind of timestamp annotation data needed to learn these models, mimicking real-world camera placement/topology/temporal aspects unlike other benchmark datasets [9, 10, 13] that lack synchronized timestamp information.

Since we extend the standard re-id problem to one of recovering full spatio-temporal trajectories, we need to also carefully consider the problem of performance evaluation. Standard re-id metrics such as cumulative match characteristic (CMC) curves [14] only capture the likelihood that a re-id algorithm matches a probe image to the correct image in the gallery, and are not designed for full trajectory recovery (see Fig. 1 for differences). To bridge this key gap, we propose three new evaluation metrics, each of which covers various practical aspects as we transition algorithms for this problem to real-world systems. Specifically, we consider three key aspects in the design of these evaluation metrics: (a) how accurate is the recovered trajectory?, (b) what duration of the total time represented by the recovered trajectory is accurate?, (c) what was the last time the algorithm correctly re-identified a probe before s/he went missing (i.e., the system stopped tracking)?. We evaluate our proposed algorithm on RPIfield using these metrics, establishing a first baseline which we hope will spur further research in this area.

To summarize, our key contributions include:

- We present a new problem, spatio-temporal trajectory recovery, as an extension of the conventional re-id paradigm for finding persons of interest in wide-area camera networks.
- We present RPIfield, a new large-scale multi-camera dataset that provides the first testbed to study trajectory recovery by means of explicit spatio-temporal trajectory annotations.
- We present a novel algorithm informed by both re-id-based appearance similarity and topology-aware temporal transition models to automatically recover the trajectory of a person of interest, establishing a first baseline for this new problem.
- We propose three new evaluation metrics to evaluate algorithms for trajectory recovery. These metrics address the limitations of standard protocols such as CMC curves and take a more practical systems-based approach.

2 Related Work

While there has been extensive research in re-id over the past decade (see Karanam *et al.* [14] and Zheng *et al.* [15] for surveys), here we overview recent developments most relevant to the two overarching themes we consider in this paper: temporally annotated re-id datasets and multi-camera re-id/tracking. We note that our proposed approach is compatible with any camera-to-camera re-id algorithm, and that the design of such algorithms is not our focus here.

2.1 Temporally Rich Re-id Datasets

Existing re-id algorithms are typically evaluated on academic re-id datasets such as Market1501 [16], CUHK03 [17], DukeMTMC4ReID [12], PRID [13], 3DPeS [18], VIPeR [8], WARD [19], iLIDSVID [9] and RAiD [20]. These datasets are hand-curated to only contain sets of bounding boxes for the probes and the corresponding matching candidates in the gallery. On the other hand, re-id is typically only one module of a larger end-to-end surveillance system, as noted in the systems paper of Camps *et al.* [21]. Consequently,

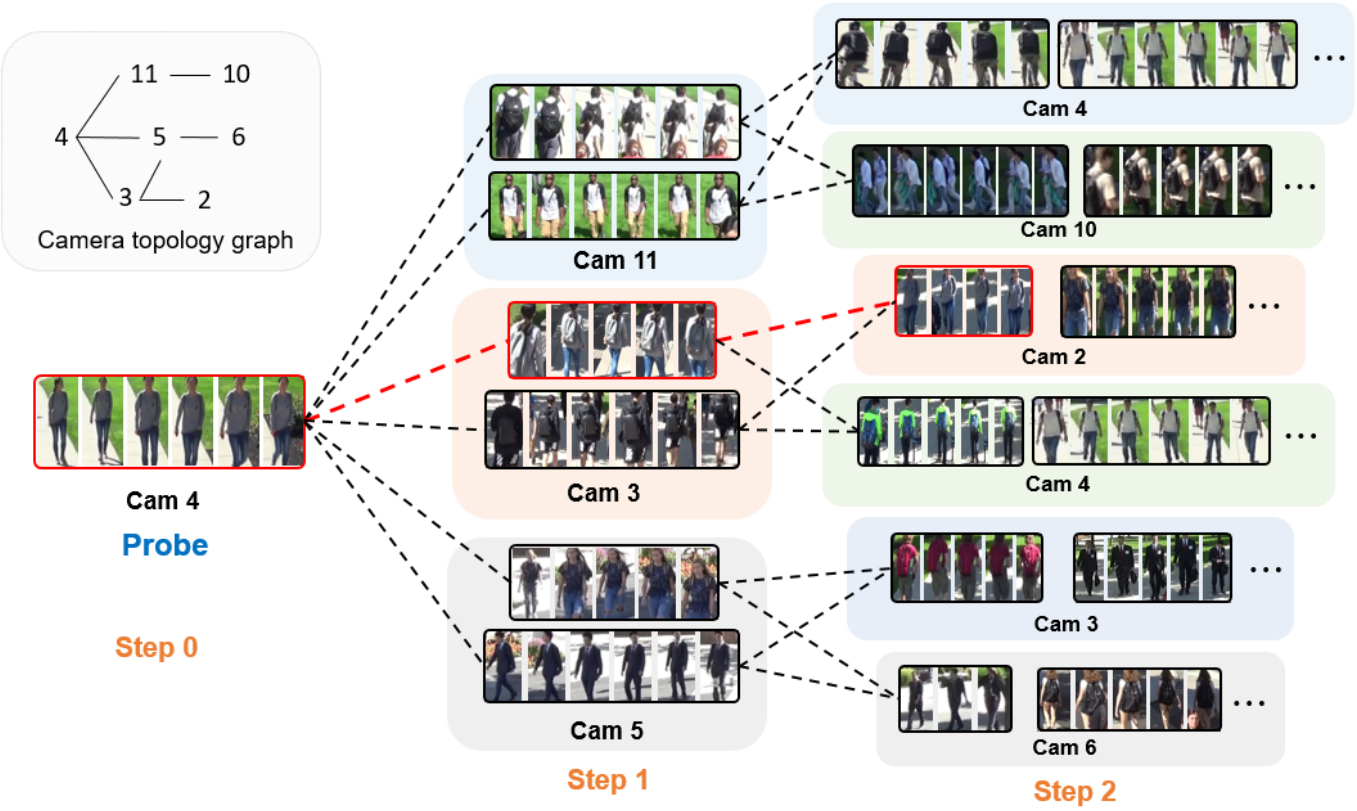


Fig. 2: Illustrating the search process for trajectory recovery. At each step, the parent nodes are expanded with new children nodes, which are candidates appearing at adjacent cameras to the parent node within a time window. Images are selected from the RPIfield [7] dataset.

it is important to construct datasets that help study this critical module from a real-world operational perspective. As described in the previous section, the spatio-temporal trajectory recovery problem, with a particular focus on temporal aspects, is at the forefront.

Figueira *et al.* proposed the HDA+ [22] dataset comprised of 30-minute long videos from 13 disjoint cameras as a testbed for evaluating an automatic re-id system. Gou *et al.* presented a much larger-scale multi-camera dataset, called DukeMTMC4ReID [12], based on the DukeMTMC dataset [11], where images corresponding to 1852 unique people were captured from a disjoint 8-camera network. Wei *et al.* proposed the large-scale person re-id dataset MSMT17 [23], which contains 12 outdoor and 3 indoor cameras using the Faster RCNN detector to collect person images. While these and other relevant datasets (described further in [14]), are multi-shot and multi-camera by design, i.e., including consecutive image frames of each person's appearance in multiple camera views, they lack the crucial timestamp information for person appearances needed for constructing spatio-temporal trajectory annotations. As opposed to all the datasets discussed above, our proposed RPIfield dataset (Section 3) provides consecutive sequences of each probe's reappearances with corresponding spatial and temporal information, thus enabling a careful study of re-id and trajectory recovery from the temporal perspectives discussed in Section 1.

2.2 Trajectory Recovery in Multi-Camera Networks

There have been recent attempts to integrate real-world systems constraints, such as temporal information and camera topology, in re-id algorithm development [24, 25], where appearance timestamps are used to infer camera topology and improve re-id matching accuracy in a bootstrapped fashion. One example is the work of Lv et al. [26], in which spatial-temporal transfer statistics were integrated as an intermediate module to learn a re-id model using a transfer learning approach. However, we note our work is different in several aspects. First, in a real-world surveillance application, the camera topology information is easily available, since the camera network is typically carefully designed and planned prior to physical installation, thereby reducing the likelihood that we may need on-the-fly topology inference. We therefore directly use the available network topology information (Figure 3) for temporal transition modeling. Second, as we discuss in Section 4.2, we utilize predictions from our temporal transition model along with the re-id matching score to construct a comprehensive confidence score for reducing the search space, as opposed to inferring the camera topology in prior work. Finally, while our focus is trajectory recovery instead of pairwise re-id as in [24–26], we note that our approach can be regarded as a plug-in module for extending and enabling any existing re-id algorithm to recover trajectories of persons of interest. Furthermore, while methods like those of Lv et al. [26] require dataset/application-specific retraining of the re-id model, our proposed temporal strategy works independently of any appearance feature learning/matching module as part of a real-world deployment.

More generally, the problem of trajectory recovery is related to multi-camera tracking (MCT) [27–33]. While most of these MCT approaches attempted to improve the overall retrieval accuracy with either improved feature representations or metric learning strategies, our focus is to infer the complete (camera spatial) motion trajectories of the person of interest. In the aforementioned MCT problem, the output is a ranked list, similar to standard re-id, whereas our output is a time-stamped multi-camera trajectory (see Figure 1).

An overview of even earlier work in multi-camera tracking and trajectory association can be found in [34]. While most of these methods focused on person/object tracking across overlapping camera views [35–38], there were some attempts to associate trajectories under the non-overlapping-views scenario [39, 40]. The performance of these methods depends on accurate camera calibration and synchronization and is largely limited by the representational ability of early and classical feature description methods [41]. Consequently, it may be challenging to scale these methods to modern surveillance systems involving large, wide-area camera networks with substantial viewpoint changes. Our method, comprised of appearance description and inter-camera transition time modules, is agnostic about the choice of the re-id representation scheme, lending itself to be flexibly used as a plug-in to the most recent advances in re-id algorithm development. Furthermore, our algorithm is solely based on the known camera network connectivity and requires neither accurately calibrated cameras nor known viewpoint changes in the camera network during the search process. Note that while camera calibration information might be needed to infer the camera network topology, we assume this topology information is known a priori, and hence do not need the network to be calibrated.

3 The RPIfield Dataset

As discussed in Section 1, in real-world forensic applications, system users usually desire to continuously track a person of interest appearing in different camera views within a widely-spread camera network. In order to simulate such a real-world scenario, we introduce a new multi-shot multi-camera re-id dataset, called RPIfield [7], containing explicit spatio-temporal trajectory annotations for all persons of interest. The dataset and additional path annotations are available at https://github.com/MZhengRPI/RPIfield.

RPIfield is constructed from 12 manually synchronized cameras placed around an outdoor field on the campus of Rensselaer Polytechnic Institute. Figure 3 shows the camera network layout. 6 poles were positioned around the field, one each at points A through F. Each red arrow represents a camera with its corresponding viewing direction.

In the RPIfield dataset, all 12 surveillance cameras were placed on top of 3.3m high poles, angled slightly downwards to simulate real-world surveillance cameras. The dataset was collected in a continuous period between 11:30 AM and 2 PM on a weekday to capture heavy foot traffic. Each of the 112 known participants (actors) was asked to walk along one unique, pre-defined path (provided by us) between different points (A through F) around the field shown in Figure 3. Adjacent cameras may have overlapping views depending on the distance between them, e.g., cameras 5 and 6. To make our algorithm widely applicable to surveillance systems with non-overlapping camera views, we do not use this overlap information in our design in any capacity.

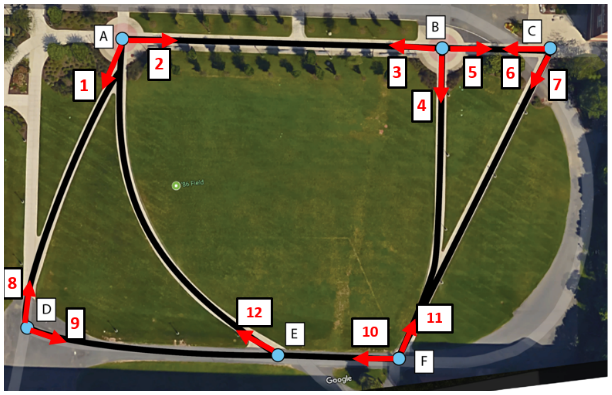


Fig. 3: Overhead view of the RPIfield camera locations and orientations, superimposed on a map of the RPI ’86 field.

In order to ensure re-appearances of the actors in all camera views, each assigned path for a participant contains at least 3 different points, and each participant was assigned a different path. A statistical summary of RPIfield and its comparison to two related and popular benchmark datasets is presented in Table 1, where each row from top to down shows the number of cropped bounding boxes (#Bounding Box), unique participants (#Identities), reappearances of participants (#Reappearaces), distracting pedestrians (#Pedestrians), image sequences (#Sequences), video length (Len.) in minutes for each camera view, average appearance duration (Dur.) in minutes over all participants, and whether the dataset contains actors. From Table 1, we can see that RPIfield has a large distractor to probe ratio (≈ 36:1), which is crucial to simulate a realistic scenario, and a substantial appearance duration (approximately half an hour) for each participant, making it well-suitable for generic temporal study of person re-id. To collect image sequences for all pedestrians appearing in the system, we used an off-the-shelf person detector, based on the aggregated channel features (ACF) algorithm of Dollar et al. [42, 43] to crop individual images in each frame, and then manually associate same-person images within and across different camera views. Please see Zheng *et al.* [7] for a more detailed comparison to existing benchmark re-id datasets.

We emphasize that the 112 probes in RPIfield correspond to **known** actors, who were provided specific walking instructions to aid in the kind of trajectory recovery research we discuss in this work. Each participant was assigned a different path to enrich the variety and difficulty of the problem. We manually associated the appearances of each participant across different camera views and then collected the appearances into a time-stamped sequence of spatial locations according to camera identities and the network

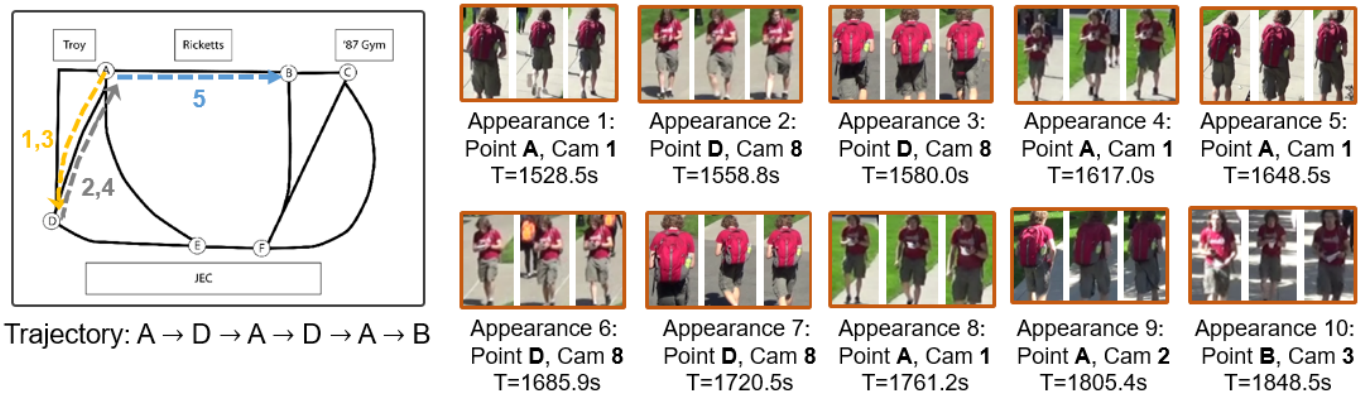

Troy
Ricketts
'87 Gym
A
B
C
D
E
F
5
1,3
2,4
JEC
Trajectory: A → D → A → D → A → B
Appearance 1: Point **A**, Cam **1** T=1528.5s
Appearance 2: Point **D**, Cam **8** T=1558.8s
Appearance 3: Point **D**, Cam **8** T=1580.0s
Appearance 4: Point **A**, Cam **1** T=1617.0s
Appearance 5: Point **A**, Cam **1** T=1648.5s
Appearance 6: Point **D**, Cam **8** T=1685.9s
Appearance 7: Point **D**, Cam **8** T=1720.5s
Appearance 8: Point **A**, Cam **1** T=1761.2s
Appearance 9: Point **A**, Cam **2** T=1805.4s
Appearance 10: Point **B**, Cam **3** T=1848.5s

Fig. 4: Left: Overview of annotated trajectory of one participant in the RPIfield dataset. The underlying map is a sketch abstracted from Figure 3. Right: Corresponding reappearances collected by the system.

TABLE 1: Statistics of the RPIfield dataset and comparison to two popular benchmark datasets.

Datasets	RPIfield	MARS [10]	Duke [11]
#Bounding Box	601,581	1,067,516	46,261
#Identities	112	1,261	1,852
#Reappearances	802	-	-
#Pedestrians	3,996	3,248	21,551
#Sequences	6,577	20,715	-
Video Len. (m)	1,826	-	680
Appearance Dur. (m)	30.5	-	-
Actors	**Yes**	No	No

topology. An illustrative example of the trajectory for a participant and corresponding collected image sequences is shown in Figure 4. The participant in Figure 4 was asked to walk along the path $A \rightarrow D \rightarrow A \rightarrow D \rightarrow A \rightarrow B$, he appeared in camera views 1, 8, 1, 8, 1, 2 and 3 accordingly. The cropped image sequences of the participant under different camera views are shown on the right.

4 Approach

As we briefly discussed above, there are two key challenges to automatically recover the trajectory of the person of interest. First, it is computationally infeasible to evaluate the likelihood of each possible path in the search tree during the reconstruction process, as shown in Figure 2. Next, the pairwise-appearance performance of off-the-shelf re-id algorithms is limited due to challenges such as occlusions or distractors, negatively impacting the downstream accuracy of the reconstructed trajectory. In this section, we describe the details of our approach that is particularly designed to address these issues. In Section 4.1, we show how inter-camera temporal transition statistics, based on pairwise re-id of the same identities and the corresponding synchronized timestamps, can help address the challenge of limited re-id matching accuracy. In Section 4.2, we show how such temporal statistical modeling and re-id matching can help us develop an effective search space pruning strategy, leading to efficient recovery of the trajectory of the person of interest.

4.1 Temporal Modeling

In this section, we propose to use the time-stamped reappearance information from RPIfield to learn an inter-camera temporal transition model that predicts the probe's transition time between adjacent cameras. We show this can efficiently shrink the size of the candidate pool and filter false alarms, thereby improving pairwise appearance-based re-id performance.

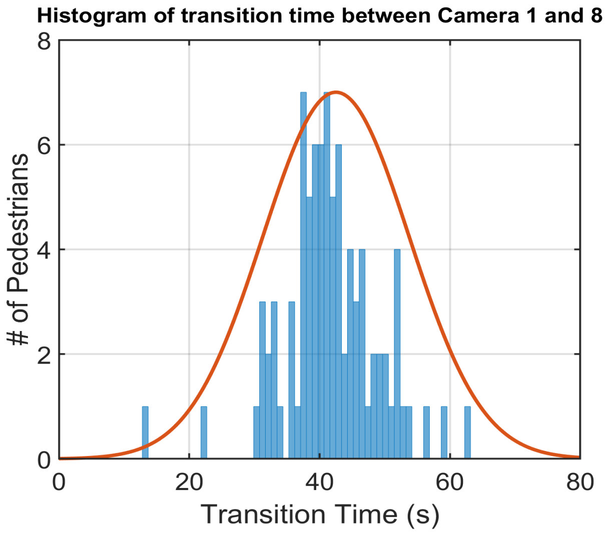


Fig. 5: Transition time modeling.

In the camera network topology of RPIfield shown in Figure 3, we estimate a transition time model given the timestamp statistics from the training data for all adjacent pairs of cameras (e.g., cameras 1 and 8, cameras 11 and 4, etc.). We define the transition time of a person as the difference between his/her time of first entry in the two adjacent cameras. Specifically, given the adjacent camera pair (c_m, c_n), we estimate the parameters of a Gaussian model $N_{mn}(\mu_{mn}, \sigma_{mn}^2)$ from the (known) timestamps of same identities walking along the (c_m, c_n) path. An illustration of a histogram of these transition times between cameras 1 and 8 and the learned Gaussian distribution is shown in Figure 5 (right). During inference, given the arrival time t_m of the probe in camera c_m, we calculate the temporal confidence score for candidates appearing in c_m's adjacent camera c_n (at time t_n) based on this learned Gaussian model $N_{mn}(\mu_{mn}, \sigma^2_{mn})$ as:

$$\tau = \exp\left(-\frac{[(t_n - t_m) - \mu_{mn}]^2}{2\sigma^2_{mn}}\right) \tag{1}$$

This way, candidates with inter-camera transition times more closely following the learned Gaussian distribution will have a higher confidence scores, and hence will be more likely to be selected as the correct match to the probe. By calculating this temporal confidence score, we can easily filter out distractor candidates that may produce high appearance similarity scores, e.g., people who look very much like the person of interest but are ruled out as matches due to the temporal model.

4.2 Pruning Strategy

As shown in Figure 2, in the trajectory reconstruction process, the number of nodes in the search tree grows exponentially as the step increases. Assuming the system on average collects N pedestrians at each step, without pruning, the total number of leaf nodes in the search tree (candidates included in the camera system) at step i would be N^i. In real-world surveillance systems where cameras are typically spatially widely spread, the transition time of the probe between different cameras can be long. Consequently, N may be quite large (hundreds or even thousands) due to the long duration between the search steps and (likely) large crowd flow density. In the following, we illustrate how our proposed pruning strategy facilitates effective and efficient trajectory reconstruction. For clarity of exposition, in Table 2, we first present a list of all the notations we use subsequently.

$\mathcal{T}$	The search tree visualized as in Figure 2.
$\mathcal{N}$	The set of nodes in the search tree $\mathcal{T}$.
$Q = (P, S, t, c)$	A node in the search tree $\mathcal{T}$, representing one appearance (image sequence) of a certain pedestrian appearing in the system, where P denotes the feature vector representation of the image sequence, obtained using some combination of re-id algorithm and feature aggregation scheme (*e.g.*, average pooling), S is the confidence score representing the likelihood of the node to be the correct match to its ancestor nodes, and c and t are the camera ID and arrival time of the pedestrian image appearance.
$\mathcal{W}^i$	The set of nodes collected at step i before pruning (usually called depth in graph theory), i.e., $\mathcal{W}^i = \{Q \in \mathcal{N} \mid step(Q) = i\}$.
Q^i_m	The mth node at step i, $Q^i_m \in \mathcal{W}^i, m = 1, \ldots, \lvert\mathcal{W}^i\rvert$. $Q^i_m = (P^i_m, S^i_m, t^i_m, c^i_m)$
$\mathcal{P}a(Q^i_m)$	Parent node of Q^i_m: $\mathcal{P}a(Q^i) = (P_{\mathcal{P}a(Q^i_m)}, S_{\mathcal{P}a(Q^i_m)}, c_{\mathcal{P}a(Q^i_m)}, t_{\mathcal{P}a(Q^i_m)}) \in \mathcal{W}^{i-1}$. Sometimes the superscript may added to denote the depth of $\mathcal{P}a(Q^i_m)$, i.e., $\mathcal{P}a^{i-1}(Q^i_m)$.
$\mathcal{A}n^j(Q^i_m)$	A sequence of ancestor nodes of Q^i_m: $\mathcal{A}n^j(Q^i_m), j \in \{1, \ldots, i-1\}$. $\mathcal{A}n^j(Q^i_m) = (P_{\mathcal{A}n^j(Q^i_m)}, S_{\mathcal{A}n^j(Q^i_m)}, c_{\mathcal{A}n^j(Q^i_m)}, t_{\mathcal{A}n^j(Q^i_m)})$
$\mathcal{C}hi(Q^i_m)$	Set of children nodes of node Q^i_m, $\mathcal{C}hi(Q^i_m) \in \mathcal{W}^{i+1}$. Given $Q^i_m = (P^i_m, S^i_m, t^i_m, c^i_m)$, $\mathcal{C}hi(Q^i_m)$ represents a collection of pedestrians appearing in cameras adjacent to c^i_m within a pre-specified time window T, i.e., from t^i_m to $t^i_m + T$.

TABLE 2: List of notations used.

To reconstruct the trajectory of the probe, we start from the probe's first appearance at step 0, where the time t^0 and camera ID c^0 are given. Thus the root note can be denoted as $Q^0 = (P^0, S^0, t^0, c^0)$. S^0 is initialized to zero for the root node and updated to S^i at each step i as discussed next. Then, given Q^0 in step 1, we collect all appearances (children nodes of Q^0) $\mathcal{W}^1 = \mathcal{C}hi(Q^0)$ of all pedestrians in cameras adjacent to c^0 in the time frame t^0 to $t^0 + T$ as all the possible candidates.

To compute the confidence score S^1_m for each node $Q^1_m, m = 1, \ldots, |\mathcal{W}^1|$, we first compute the appearance similarity score a^1_m for each node Q^1_m using an off-the-shelf re-id algorithm given the probe and candidate appearances P^0 and P^1_m, i.e., $a^1_m = \text{sim}(P^0, P^1_m)$, where $\text{sim}(x, y) \in [0, 1]$ is an appropriate similarity function between feature vectors x and y. We then compute the temporal confidence score from the learned transition model using Equation 1 given t_0 and the timestamps t^1_m. Based on these two scores, we compute the overall confidence score for node Q^1_m at step 1 as:

$$S^1_m = a^1_m + \alpha\tau^1_m \tag{2}$$

where α is a parameter that controls the relative importance of the temporal τ^1_m and appearance a^1_m confidence scores. Note that the range of both a^1_m and τ^1_m is $[0, 1]$.

Since $|\mathcal{W}^1|$ is large, with S^1_m for each node, we can now easily prune out nodes in the search tree with low appearance similarity and poor transition model fits to effectively reduce the search space. Specifically, we first generate a sorted node list according to the confidence scores of the nodes in $\mathcal{W}^1$. As discussed earlier, the true/correct match to the probe is frequently not at the very top of the list due to re-id algorithm imperfections. We then retain the top R candidates to be our possible matches at step 1 for further computation, i.e., $\hat{\mathcal{W}}^1$ denotes the set of nodes at step 1 after pruning and $|\hat{\mathcal{W}}^1| = R$. We report results using $R = 10$ in the experiments, and analyze the impact of this choice with an ablation study in Section 5.3.5. Additionally, in order to get better matches at subsequent steps (e.g., we may see the front of the person in step 0 and his/her back in step 1), we augment the original probe feature representation with the representations from putative matches along the trajectory as the algorithm proceeds. Specifically, when computing S^i_m for node Q^i_m, we preserve feature representations from all previous steps $\{\mathcal{P}_{\mathcal{A}n^j(Q^i_m)}, j = 1, \ldots, i-1\}$ to compute a more robust and representative appearance confidence score. This is especially important for steps $i > 2$, which we discuss next.

Given $\hat{\mathcal{W}}^1$, at step 2, for $\forall Q^1 \in \hat{\mathcal{W}}^1$ from step 1, we repeat the same search process, collecting their children nodes as $\mathcal{W}^2 = \{\mathcal{C}hi(Q^1) | Q^1 \in \hat{\mathcal{W}}^1\}$. For $\forall Q^2 \in \mathcal{W}^2$, the confidence scores are calculated based on appearance and temporal confidence with Equation 2. Since we updated the probe feature set in step 1 as discussed above, for a node Q^2_m at step 2, the appearance confidence score a^2_m is computed as:

$$a^2_m = \frac{1}{1+\gamma}\left\{\text{sim}(P^0, P^2_m) + \gamma\text{sim}(P^1_{\mathcal{P}a(Q^2_m)}, P^2_m)\right\} \tag{3}$$

Here, γ is a parameter that controls the relative importance of the original probe feature vector P^0 vs. the newly-collected appearances $P^1_{\mathcal{P}a(Q^2_m)}$ (i.e., the appearance of the parent node of Q_m^2) during step 1. After computing S_m^2 for node Q_m^2, we then update it as the sum of its parent node score $S^1_{\mathcal{P}a(Q^2_m)}$ (step 1) and the child node score S_m^2 (step 2) to aggregate the confidence information from past and current steps, for further sorting and pruning.

For steps $i > 2$, when computing the appearance confidence score, there will be multiple feature representations preserved from previous steps (from ancestor nodes $\{P_{\mathcal{A}n^j(Q^i_m)}, j = 1, ..., i-1\}$). We then select the best appearance match $\hat{P}^*$ from step 1 to $i-1$, i.e., $\text{sim}(\hat{P}^*, P^i_m) = \max_{j=1,\ldots,i-1}\{\text{sim}(\hat{P}_{\mathcal{A}n^j(Q^i_m)}, P^i_m)\}$. Thus, the general appearance score of node Q^i_m at step i is computed as

$$a^i_m = \frac{1}{1+\gamma}\left\{\text{sim}(P^0, P^i_m) + \gamma\text{sim}(\hat{P}^*, P^i_m)\right\} \tag{4}$$

In other words, at step i, given a new candidate, we measure its similarity to the probe and possible partial trajectories in the same spirit of Equation 3. We split the similarity computation into two parts. The first part considers only the new candidate and the original (given) feature vector. The second part uses a score aggregation scheme that considers all the remaining pairs and returns one number. This can be achieved using a variety of methods but we use the straightforward approach of picking the pair (of the $i-1$ possibilities) that results in the maximum similarity score to the candidate feature vector. We analyze the impact of the choice of γ by means of experiments reported in Section 5.3.4.

The process is repeated until the person of interest disappears from the system, as described in Algorithm 1, which outputs all intermediate results from step 1 through I. This enables the retrieval of the camera and timestamp information of the person of interest by identifying the node with highest confidence score in $\mathcal{W}^I$, and then reconstructing his/her traveling path given the camera topology information.

Algorithm 1 Online search for trajectory recovery of a person of interest in a multi-camera system.

```
Given probe's first appearance $P_0$, time $t_0$, camera ID $c_0$.
Initialize $S_0 = 0$.
Construct root note: $Q^0 = (P^0, S^0, t^0, c^0)$.
for step $i = 1, \ldots, I$ do
  for $Q^{i-1} \in \mathcal{W}^{i-1}$ do
    Collect children nodes $\mathcal{C}hi(Q^{i-1})$.
    for $Q^i \in \mathcal{C}hi(Q^{i-1})$ do
      Calculate appearance confidence score $a^i$ from Equation 4.
      Calculate temporal confidence score $\tau^i$ from Equation 1.
      Calculate confidence score $S^i$ from Equation 2.
      $Q^i = (P^i, S^i + S^{i-1}_{\mathcal{P}a(Q^i)}, t^i, c^i)$.
      $\mathcal{W}^i \leftarrow Q^i$.
    end for
  end for
  Sort nodes $\mathcal{W}^i$ according to confidence score. Keep top $R$ nodes.
end for
Output sequence $\{Q^i, i = 1, \ldots, I\}$ with highest $S^I$.
```

Figure 6 presents a visualization of the retained reappearance sequences for a certain participant in our experiment from steps 1 to 4. While the ground-truth trajectory in this case was correctly recovered with rank 1 at step 4, we can see that the correct matches at steps 1 and 2 were at ranks 2 and 5 respectively, demonstrating the importance of retaining candidates at lower ranks. Red outlines around the candidates show correct matches, which rise to the top of the ranked list as the algorithm proceeds. We can see that at step 4, there are several sequences without any false detections (e.g., the rank 4 trajectory), but that may have intermediate appearances missing as discussed in the next section.

5 Experiments and Results

5.1 Dataset and Implementation Details

We perform all our experiments on the RPIfield dataset introduced in Section 3. For learning the re-id appearance model, we split the data equally into two parts, with a total number of 56 training and testing identities. To run experiments on the first 56 participants, we use the second half of the 56 identities as training data, and vice versa. We train our camera-to-camera re-id model using CASN [4] with an IDE backend [16], a recent algorithm with state-of-the-art performance on standard benchmark datasets, returning a 2048-dimensional feature vector for each candidate. All the elements of this vector are positive (because of a ReLU activation after the last convolutional layer [4]) and it is normalized to have unit l_2 norm. We use the cosine similarity function to compute $\text{sim}(x, y) \in [0, 1]$. The re-id model is the same for all the cameras; i.e., we do not learn pair-specific models.

The temporal model introduced in Section 4.1 is learned from annotated timestamps and reappearances between adjacent cameras of the 56 training identities. For all experiments in this section, we set $\alpha = 0.25$ (Equation 2), $\gamma = 0.5$ (Equation 3) and $R = 10$ (Section 4.2), except otherwise stated.

5.2 Evaluation Metrics and Results

Since we are addressing a fundamentally temporal problem, existing re-id evaluation protocols such as cumulative match characteristic (CMC) curves or mean average prevision (mAP) are not well-suited for evaluating our algorithm. In Figure 7, we present examples of the reconstructed reappearances for three participants in RPIfield using our proposed algorithm.

In each subfigure of Figure 7, the x axis is the real time in seconds. The top red bar represents the ground-truth reappearances of the person of interest starting from his/her first appearance till he/she exits the camera network. The bottom blue bar shows the correctly retrieved reappearances computed by our algorithm, and the bottom green bar shows the mis-identified reappearance of the person of interest (false alarm). From Figure 7, we can see that the proposed recovery algorithm in some cases is able to correctly retrieve every reappearance of the person of interest, e.g., Figure 7(a), providing accurate and complete path. In other cases, it may miss several intermediate steps or mistakenly identify a wrong ID and then resume tracking

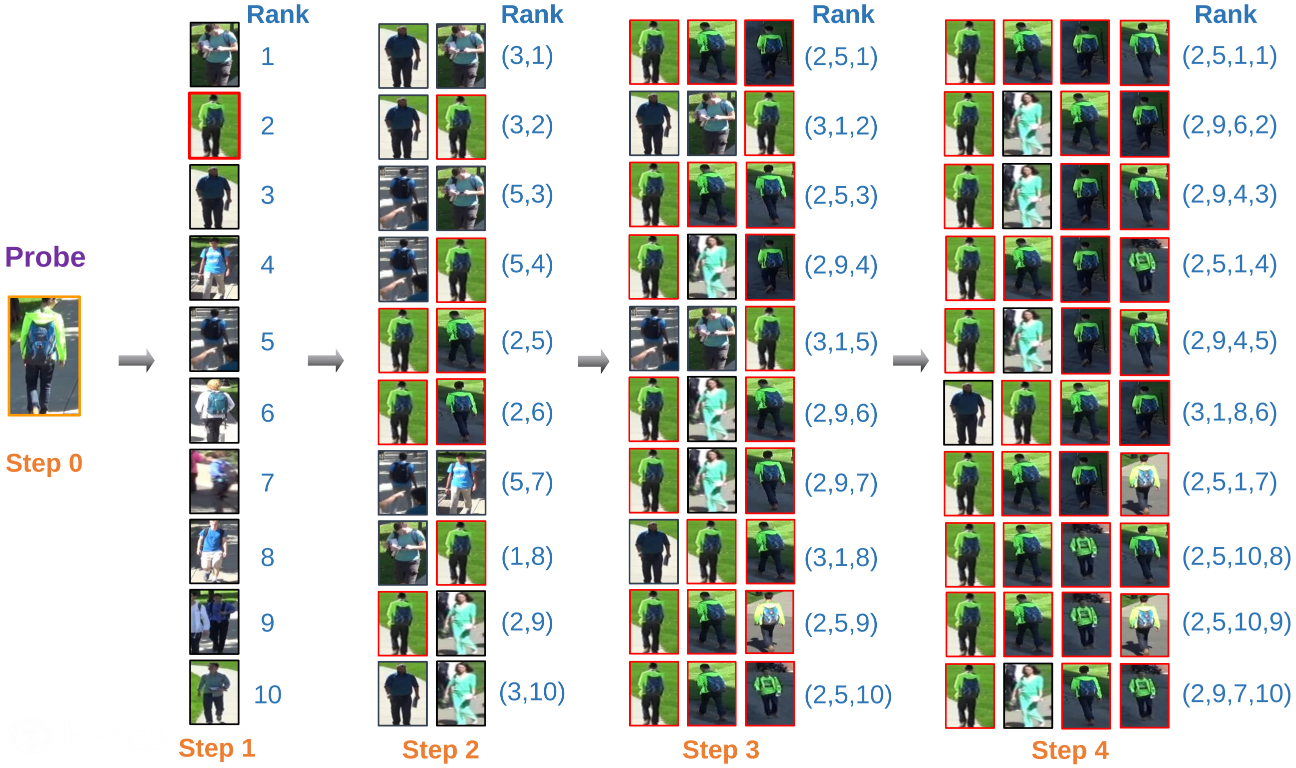


Fig. 6: A visualization of the top-10 retained reappearance sequences from step 1 through 4 for one experimental probe. At each step i, the rank sequences for each candidate along the trajectory are indicated. Red boxes around candidates indicate correct matches. In this case, the ground-truth trajectory for this probe is detected at rank 1 at step 4. Several other trajectories without false alarms (but missing intermediate reappearances) also appear in the top-10 list.

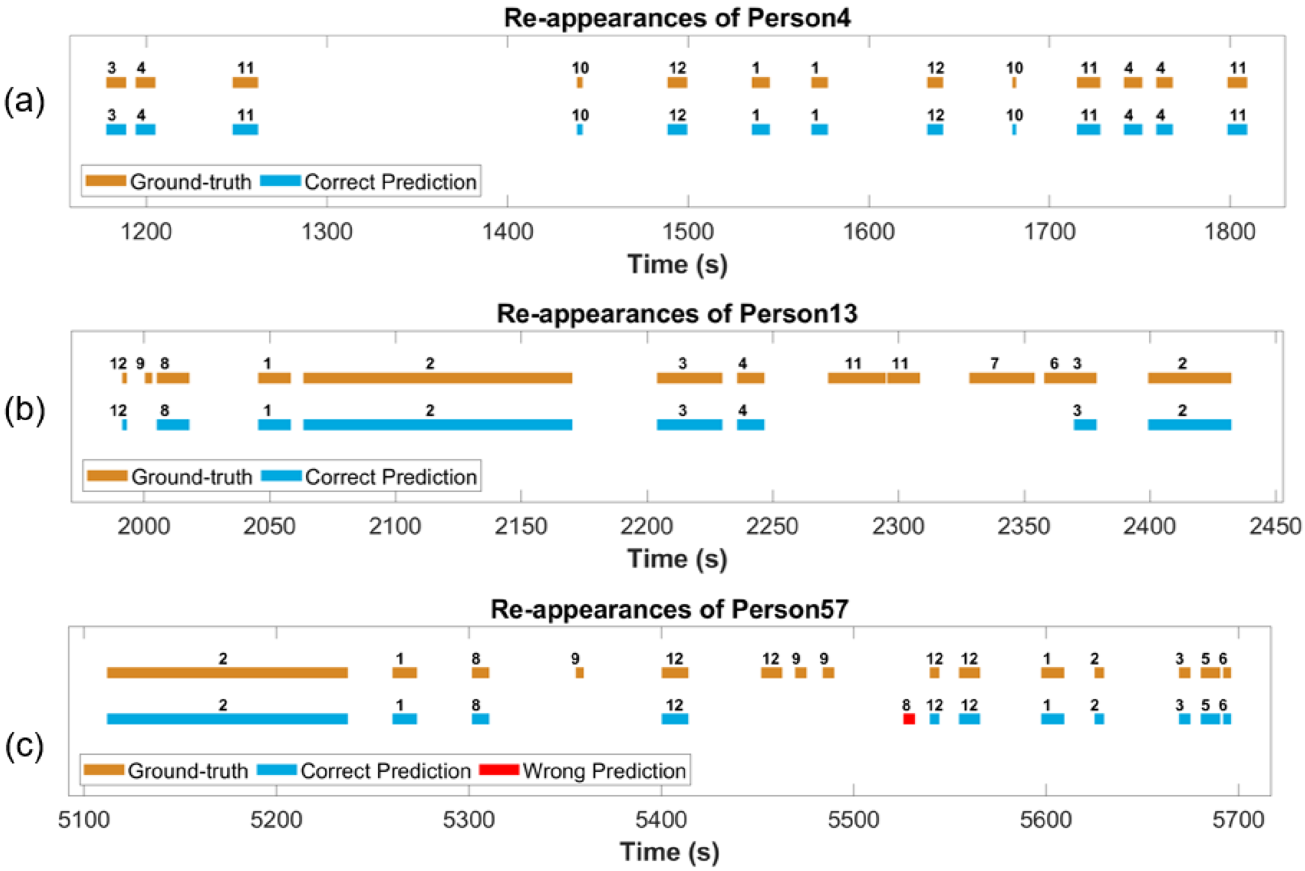


Fig. 7: Path recovery results of several participants in the RPIfield dataset.

after a certain amount of time, causing incomplete/partial-correct recovery of the probe's trajectory. For example, in Figure 7(b), after $T = 2360s$ where the person of interest is recognized by the algorithm as appearing in camera 4, the system loses track of the person due to its inaccurate pairwise re-id performance at the following steps in cameras 11, 7, and 6.

In Figure 7(c), the system recognizes a wrong person at Camera 8 after the person of interest appears at Camera 12, missing his/her true appearances at Camera 12 and 9. However, the algorithm picks up the probe's reappearance at later steps (most likely because of their stronger similarities to the first reappearance), and then resumes tracking until the person of interest leaves the system.

In these cases where the algorithm cannot fully recover the probe's accurate trajectory, we would certainly like to know how much information we miss from the reconstructed trajectory, or in other words, how confident we can be about the accuracy of the reconstructed trajectory. Since the standard CMC curve clearly does not capture these aspects, we propose three new evaluation metrics:

- trajectory duration percentage (TDP) curve
- trajectory reconstruction accuracy (TRA) curve
- tracking lifetime percentage (TLP)

We define each metric below.

1) **TDP curve.** The algorithm is likely to lose track of the person of interest at intermediate steps if encountering distracting factors, like misdetections, a sudden large crowd of distractors, illumination changes, or occlusions. Thus, from the temporal perspective, for a trajectory of a probe's reappearances computed by our algorithm, we would like to know the percentage of the correctly retrieved appearance time relative to the ground-truth total appearance time of this probe. To this end, we first calculate the duration spanned by the correctly retrieved reappearances (the sum of the length of the blue bars in Figure 7) divided by the time duration spanned by ground-truth reappearances (sum of the length of the red bars) for each participant. We then aggregate the computed percentage for all participants by specifying the x axis as percentage p (in %) and plotting the percentage of the total participants whose recovered duration percentage is no smaller than p. We call this the TDP curve. The evaluation result of our proposed algorithm on all 112 participants is shown in Figure 8(a). The average trajectory duration percentage over all participants is 80.1%, which is the area under the TDP curve shown in Figure 8(a).
2) **TRA curve.** From the retrieved reappearances shown in Figure 7, we can reconstruct the trajectory of the person of interest given the camera topology information (Figure 5). For example, the ground-truth trajectory for participant 13 in Figure 7(b) is $E \rightarrow D \rightarrow A \rightarrow B \rightarrow F \rightarrow B \rightarrow A$ (string:"EDABFBA"), while the reconstructed path is $E \rightarrow A \rightarrow B \rightarrow B \rightarrow A$ (string:"EABBA"). To measure the difference between the reconstructed trajectory and the ground-truth trajectory, we adapt the concept of edit distance (also referred to as Levenshtein distance) from information theory and linguistics for measuring the difference between two strings [44]. It is defined as the minimum number of single-character operations (insertions, deletions, or substitutions) needed to edit one string into the other. For example, assume the cost of each operation: insertion, deletion or substitution, is equal to 1. To edit "SUNDAY" into "SATURDAY", we need at least following 3 steps: 1. insertion: "SUNDAY" $\rightarrow$ "SAUNDAY"; 2. insertion: "SAUNDAY" $\rightarrow$ "SATUNDAY"; 3. substitution: "SATUNDAY" $\rightarrow$ "SATURDAY". Thus the edit distance between the strings "SATURDAY" and "SUNDAY" is 3.
In our experiments, to edit the computed trajectory string into the ground-truth string, an insertion indicates a missing reappearance identification, while a deletion or substitution indicates an incorrect appearance identification. To calculate the reconstruction accuracy, we assign equal costs to each of the operations (insertion, deletion or substitution), and then measure the edit distance d_e between reconstructed and ground-truth strings. For example, for participant 13, the edit distance is $d_e = 2$ (we need 2 insertions to change "EABBA" to "EDABFBA"). Since each participant has a different number of reappearances, i.e., a different length of the ground-truth trajectory string, we then calculate the relative edit distance $\hat{d}_e$ as the edit distance d_e divided by the length of the ground-truth trajectory to measure the relative trajectory computation error for each participant. For example, $\hat{d}_e = 28.6\%$ for participant 13. We then compute the similarity (accuracy) of the reconstructed trajectory to the ground truth as $p = 1 - \hat{d}_e = 71.4\%$, and aggregate the accuracy results for all participants in a single graph, called the TRA curve. Note that this aggregation is similar to how we compute the TDP curve. The TRA curve evaluation result of our algorithm on RPIfield is shown in Figure 8(b). The average trajectory reconstruction accuracy over all 112 participants is 82.5%.
3) **Tracking lifetime percentage.** As we discussed earlier, the algorithm may miss the person of interest for several intermediate stops, but then get back on track at a certain point. Depending on the application requirements, this may be more important when compared to the case where the algorithm successfully tracks the person for the same total duration but is not able to re-acquire the person after tracking loss. For example, in Figure 7(b), the algorithm successfully identifies the last reappearance of participant 13 in Camera 2 before he/she disappears in the system. From this perspective, the system follows the person of interest from their first appearance until their last appearance, with several middle steps missing. In this case, the tracking lifetime of our algorithm is 100% if the intermediate mistakes can be safely ignored. Thus for system tracking lifetime percentage evaluation, we first calculate

the time duration from the recovered first correct reappearance to the last correct reappearance with respect to the ground-truth time duration of the first and last reappearance for each participant (e.g., for participant 13, p =100%), and then aggregate the percentage evaluation for all participants as in (1) and (2). The aggregated evaluation results of our proposed algorithm are shown in Figure 8(c). The average tracking lifetime percentage over all participants is 97.0%. That is, in almost every case, our algorithm ultimately has recovered the person by the end of their known trajectory.

5.3 Analysis and discussion

5.3.1 Impact of pruning

To study the necessity and impact of our pruning strategy, we evaluate the computation time and corresponding computation accuracy (mean TDP) of our approach with and without pruning. To avoid intractable computation with the increasing number of search steps, we only evaluate with the first three reappearances and report results in Table 3. We found that pruning reduces the computation time by a factor of almost 200, a substantial cost reduction, while only reducing mean TDP by a small margin, 3.2%. Without pruning, the average processing time per step is almost 2 hours, which would be considered unacceptable in a real-world surveillance system where run-time demands are generally steep. In contrast, our algorithm runs in about 37 seconds per step on a standard PC.

Method	Avg. Comput. Time	mean TDP
With pruning	37s	77.6%
Without pruning	7022s	80.8%

TABLE 3: Average computation time per step and mean trajectory duration percentages, with and without pruning. Both experiments were run on the same system with a Intel i7-6700 Quad Core Processor and NVIDIA GeForce GRX 1060 GPU.

5.3.2 Impact of the temporal transition model

We investigate the impact of the learned temporal transition model by conducting experiments with different values of α in Equation 2. The results, with the mean TDP, TRA and TLP metrics, are shown in Figure 8(d). One can note that the performance when $\alpha = 0$, i.e., calculating the confidence score without the predictions from the transition model, is much lower (e.g. the mean recovered duration percentage is 8% lower) than the best case performance (achieved when $\alpha = 0.25$). This shows that the performance of an appearance-only similarity model can be significantly improved by adding temporal constraints. Note that while increasing the value of α from 0 improves performance, increasing it beyond a point seems to give diminishing returns. This is likely because of the distribution gap between the learned temporal models and the actual re-appearance time for a test person. In our training set, we have many candidate re-appearances within a short transition time-frame, leading to a relatively narrower distribution, as opposed to many scenarios in the test set where the re-appearance transition time is much larger, leading to a less reliable temporal score. Therefore, we see a decrease in the TDP and TRA metrics as we increase α beyond 0.25. However, the TLP metric is not as sensitive to α as the other two because TLP only considers the last correct re-appearance of the person of interest. In other words, because we only consider the best-case scenario, TLP is less influenced by the temporal model. On the other hand, the TDP and TRA metrics provide a comprehensive overview of the entire trajectory and intermediate mistakes (if due to the temporal score) get reflected in the final accuracy score.

5.3.3 Ablation study

In Table 4, we quantify the impact of the camera topology information and the temporal transition model on the recovered trajectories. We tested our algorithm on two types of re-id backend methods, CASN and IDE [16]. “Baseline” in Table 4 corresponds to searching all possible trajectories without using either the camera topology or the temporal model information. More specifically, we collect all pedestrians that show up in all camera views given the probe’s first appearance, and then identify reappearances based only on the appearance similarity scores. “Baseline+Topology” corresponds to searching for appearances in adjacent cameras based on the known topology information for each step, as described in Section 4.2, but without using temporal predictions from the learned transition model ($\alpha = 0$ in Equation 2). From the results, one can note that the camera topology information and temporal model predictions are critical to the overall performance, with substantial performance improvement with respect to all of the evaluation metrics discussed in Section 5.2.

Method	TDP	TRA	TLP
Baseline(IDE)	58.5%	58.8%	82.9%
Baseline(IDE)+Topology	66.8%	63.8%	87.6%
Baseline(IDE)+Topology +Temporal (**Proposed**)	**73.6%**	**76.4%**	**91.8%**
Baseline(CASN)	61.0%	64.0%	87.5%
Baseline(CASN)+Topology	70.2%	67.9%	91.2%
Baseline(CASN)+Topology +Temporal (**Proposed**)	**80.1%**	**82.5%**	**97.0%**

TABLE 4: Results of an ablation experiment of the individual modules in our trajectory recovery algorithm. Baseline: no side information; Baseline+Topology: adding camera topology; Baseline+Topology+Temporal: our proposed algorithm based on camera topology and temporal prediction. $\alpha = 0.25$ for “Baseline+Topology+Temporal”, $\gamma = 0.5$ for all experiments.

5.3.4 Impact of probe feature set update

In a real-world deployment, as simulated by our dataset, the cameras are widely spaced and likely under varying illumination conditions, as discussed in Section 4.2. Furthermore, the person of interest is also likely to appear from different viewpoints, e.g. front, back, or side, which adds to the challenges of appearance matching. For example, as shown in Figure 4 (right), the probe image sequence (Appearance

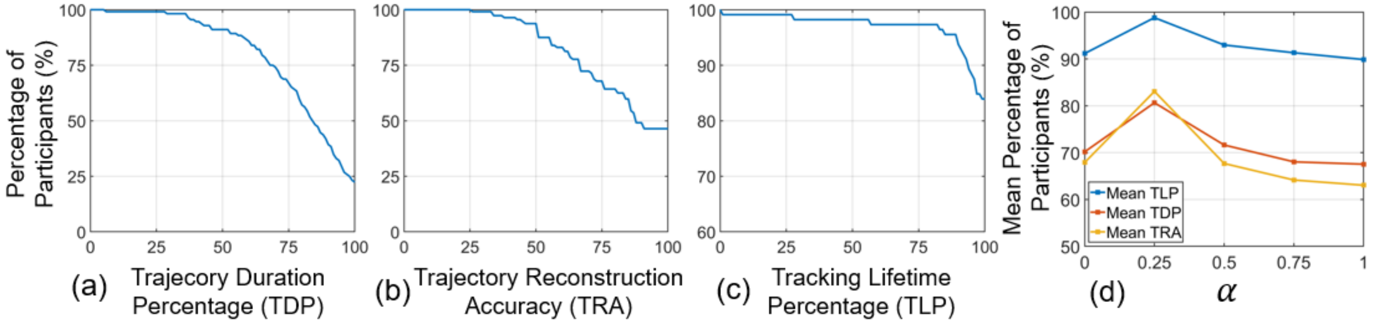


Fig. 8: Evaluation of the proposed algorithm in terms of (a) Trajectory Duration Percentage; (b) Trajectory Reconstruction Accuracy; (c) Tracking Lifetime Percentage, and (d) the effect of the learned transition temporal model. For (a)-(c), horizontal axis: percentage(%) of corresponding metric, vertical axis: percentage(%) of participants, for (d), horizontal axis: α (Eq. 2), vertical axis: mean percentage(%) of participants.

1) corresponds to the back view, while Appearances 2, 4, 8, and 10 correspond to the front view. Consequently, during searching, it is important to include the selected candidate features from previous steps (e.g., combining the feature representations from the back view in Appearance 1 and the front view in Appearance 2) into the probe feature set to enrich the variety of the feature representation.

In Table 5, we show results of our algorithm with different γ values to evaluate the influence of this updating process. $\gamma = 0$ corresponds to using a static probe feature set without any update, i.e., the probe feature is P^0 at all times. As one can note from the results, increasing γ generally improves the performance ($\gamma = 0.5$ gives the best result in our experiments) up to a certain point, beyond which the performance may deteriorate since a large γ (e.g., $\gamma = 1.0$) may introduce misdetections from previous steps, misleading the future search.

Method	TDP	TRA	TLP
$\gamma = 0.0$	72.3%	70.1%	91.6%
$\gamma = 0.25$	74.9%	77.0%	94.4%
$\gamma = 0.5$	**80.1%**	**82.5%**	**97.0%**
$\gamma = 1.0$	75.2%	75.8%	92.9%

TABLE 5: Performance improvements with probe feature update. $\alpha = 0.25$ (Equation 2) in all experiments. TDP, TRA and TLP results are mean percentage of all 112 participants.

5.3.5 Impact of selected candidate number

As discussed in Section 4.2, at each search step, as a trade-off between computational efficiency and accuracy, we retain the top R candidates with the highest confidence scores. To understand the influence of this parameter on the overall performance, we evaluate our method with different values for R, the results of which are shown in Table 6. Retaining only the first-rank candidate (i.e., $R = 1$) gives the lowest performance. This is expected since we do not anticipate that the first-rank candidate is always the true match, and any such misidentifications will likely cause later matching failures. As expected, increasing R improves the overall performance because the true matches are less likely to be pruned out from the list of retained candidates. However, it is important to note that $R = 20$ results in twice as many candidate nodes as for $R = 10$, therefore quadrupling the computation time. Increasing the value of R further will only exacerbate this situation, with little benefit of improved performance. Consequently, we chose $R = 10$ in our experiments discussed above.

Method	TDP	TRA	TLP
$R = 1$	59.0%	56.5%	86.7%
$R = 5$	72.9%	74.4%	94.5%
$R = 10$	80.1%	82.5%	**97.0%**
$R = 20$	**81.5%**	**83.0%**	**97.0%**

TABLE 6: Performance evaluation with different values for R (number of preserved nodes at each step). $\alpha = 0.25$ (Equation 2) and $\gamma = 0.5$ (Equation 3) in all experiments. TDP, TRA and TLP results are mean percentage of all 112 participants.

6 Conclusions and Future Work

In this work, we introduced spatio-temporal trajectory recovery, where the task was to automatically reconstruct the time-stamped spatial trajectory of a person of interest in a wide-area camera network. With existing benchmark person re-identification datasets lacking the kind of annotations needed to study this topic, we introduced RPIfield, a new dataset with extensive spatio-temporal annotations for known persons of interest moving along orchestrated paths in the camera network. We then proposed a new method that took a step towards solving this problem by using camera topology-informed transition time modeling and candidate search space pruning strategies. With standard re-id evaluation measures not directly relevant for this problem, we introduced three new evaluation metrics that considered various spatio-temporal aspects in the context of practical, real-world use of our algorithm. While our evaluation on RPIfield using the new metrics showed promising initial results, we posit several directions for future research:

- As discussed in Section 4.2, our current strategy for the selection of the time-window parameter T is to use a large enough number so as to not miss the true match in the accumulated gallery. While this simple strategy works well in a controlled experimental setup, a much larger and denser environment such as a major airport will lead to a very large gallery, likely having an impact on the accuracy of the recovered trajectory. An immediate possibility to address this issue is to automatically estimate this value on a per-camera-pair basis, which can be easily done by learning simple models from data (which should be easily obtainable from the site of system deployment). A better approach would be to continually fine-tune this value based on the time stamps from previously estimated trajectories, updating T for every camera pair in an online fashion.

- Our current strategy for using multishot (image sequence) information in the feature representation involves averaging all the feature vectors of each frame in a candidate's trajectory sequence. While feature averaging is one way of aggregating sequence information into a compact feature vector, there are alternative and improved methods [14] for such aggregation that can be directly plugged into our method.

  Furthermore, in our current scheme, we give equal consideration to all the candidate reappearances. While this may be sufficient in cases where reappearances are spaced apart by seconds, longer gaps between reappearances may introduce more substantial variations (e.g., changing clothes or accessories). In such cases we should investigate adaptive weighting schemes that give higher weight to the more recent reappearances.
- Our method currently uses a single re-id model, learned jointly with data from all the camera views. A simple and practical way to improve the performance when used in a real-world system is to learn camera-pair-specific re-id models. This can easily be learned from sufficient paired data.
- Given a known camera network topology, and ground-truth matches (from each pair of cameras), one can easily filter out implausible candidates and reduce the search space even further. For instance, as pedestrians walk from camera A to camera B, it is highly likely that they reappear in camera B with a certain moving direction and at a general image plane location. This means we can reliably eliminate all candidates appearing in the wrong place or walking the wrong way, helping reduce the search space and improve performance. Similarly, using information about the overlap between cameras' fields of views would likely lead to performance improvements (e.g., by fusing the appearance information of a person from the available overlapping frames).
- As discussed in Section 5.3.2, in our experiments, increasing the weight of the temporal score beyond a certain value does not bring any performance improvements. Since our training data was limited in the number of identities and not all identities pass through each pair of cameras, the amount of data available to learn these per-pair transition models is quite low (even less than 10 in a few pairs). This leads to a transition model that does not necessarily reflect the true temporal transition patterns when using the model during testing. Given the benefits the temporal model has shown, it would be worthwhile to invest in the collection of a much wider variety of transition patterns. While having a large number of people is not necessarily needed, gathering a more diverse set of per-pair temporal transition time data is crucial.

References

[1] J. Zhang, N. Wang, and L. Zhang, "Multi-shot pedestrian re-identification via sequential decision making," in *CVPR*, 2018.

[2] M. S. Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "Pose sensitive embedding for person re-identification," in *CVPR*, 2018.

[3] Y. Wang, L. Wang, Y. You, X. Zou, V. Chen, S. Li, G. Huang, B. Hariharan, and K. Q. Weinberger, "Resource aware person re-identification across multiple resolutions," in *CVPR*, 2018.

[4] M. Zheng, S. Karanam, Z. Wu, and R. J. Radke, "Re-identification with consistent attentive Siamese networks," in *CVPR*, 2019.

[5] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *ECCV*, 2018.

[6] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *ECCV*, 2018.

[7] M. Zheng, S. Karanam, and R. J. Radke, "RPIField: A new dataset for temporally evaluating person re-identification," in *CVPR Workshops*, 2018.

[8] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features." in *ECCV*, 2008.

[9] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking." in *ECCV*, 2014.

[10] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "MARS: A video benchmark for large-scale person re-identification," in *ECCV*, 2016.

[11] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking," in *ECCV Workshops*, 2016.

[12] M. Gou, S. Karanam, W. Liu, O. Camps, and R. J. Radke, "DukeMTMC4ReID: A large-scale multi-camera person re-identification dataset," in *CVPR Workshops*, 2017.

[13] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof., "Person re-identification by descriptive and discriminative classification." *SCIA: Image Analysis*, vol. 6688, pp. 91–102, 2011.

[14] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE T-PAMI*, vol. 41, no. 3, pp. 523–536, 2019.

[15] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *CoRR*, vol. abs/1610.02984, 2016.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[17] W. Li, R. Zhao, T. Xiao, and X. Wang., "DeepReID: Deep filter pairing neural network for person re-identification." in *CVPR*, 2014.

[18] D. Baltieri, R. Vezzani, and R. Cucchiara, "3dpes: 3d people dataset for surveillance and forensics," in *ACM MM Workshops*, 2011.

[19] N. Martinel, C. Micheloni, and C. Piciarelli., "Distributed signature fusion for person re-identification." in *ICDSC*, 2012.

[20] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network." in *ECCV*, 2014.

[21] O. Camps, M. Gou, T. Hebble, S. Karanam, O. Lehmann, Y. Li, R. J. Radke, Z. Wu, and F. Xiong, "From the lab to the real world: Re-identification in an airport camera network." *IEEE T-CSVT*, vol. 27, no. 3, pp. 540–553, 2017.

[22] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino, "The HDA+ data set for research on fully automated re-identification systems," in *ECCV*, 2014.

[23] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person trasfer gan to bridge domain gap for person re-identification," in *CVPR*, 2018.

[24] N. Jiang, S. Bai, Y. Xu, C. Xing, Z. Zhou, and W. Wu, "Online inter-camera trajectory association exploiting person re-identification and camera topology," in *ACM MM*, 2018.

[25] Y.-J. Cho, S.-A. Kim, J.-H. Park, K. Lee, and K.-J. Yoon, "Joint person re-identification and camera network topology inference in multiple cameras," *CVIU*, vol. 180, 2017.

[26] J. Lv, W. Chen, Q. Li, and C. Yang, "Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns," in *CVPR*, 2018.

[27] A. Yildiz and Y. Akgul, "A fast method for tracking people with multiple cameras," in *Trends and Topics in Computer Vision*, 2012.

[28] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," *CVPR*, 2018.

[29] H. Hsu, T. Huang, G. Wang, J. Cai, Z. Lei, and J. Hwang, "Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models," in *CVPR Workshops*, June 2019.

[30] C. Wu, C. Liu, C. Jiang, W. Tu, and S. Chien, "Vehicle re-identification with the space-time prior," in *CVPR*, 2018.

[31] C. Wu, M. Zhong, Y. Tsao, S. Yang, Y. Chen, and S. Chien, "Track-clustering error evaluation for track-based multi-camera tracking system employing human re-identification," *CVPR Workshops*, 2017.

[32] Y. Tesfaye, E. Mequanint, A. Prati, M. Pelillo, and M. Shah, "Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets," *IJCV*, 06 2017.

[33] F. Previtali, D. Bloisi, and L. Iocchi, "A distributed approach for real-time multi-camera multiple object tracking," *Springer MVA*, 03 2017.

[34] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, 2011.

[35] N. Anjum and A. Cavallaro, "Trajectory association and fusion across partially overlapping cameras," in *AVSS*, 2009.

[36] K. Kim and L. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *ECCV*, 2008.

[37] H. Medeiros, J. Park, and A. Kak, "Distributed object tracking using a cluster-based kalman filter in wireless camera networks," *IEEE JSTSP*, vol. 2, no. 4, pp. 448–463, 2008.

[38] S. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *ECCV*, 2006.

[39] L. Zhu, J. Hwang, and H. Cheng, "Tracking of multiple objects across multiple cameras with overlapping and non-overlapping views," in *ISCAS*, 2009.

[40] N. Anjum, M. Taj, and A. Cavallaro, "Relative position estimation of non-overlapping cameras," in *ICASSP*, 2007.

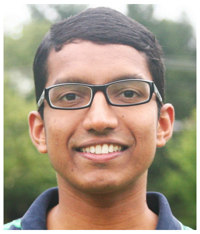
[41] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE T-PAMI*, vol. 31, no. 3, pp. 505–519, 2009.

[42] P. Dollar, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *BMVC*, 2010.

[43] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE T-PAMI*, vol. 36, 2014.

[44] F. P. Miller, A. F. Vandome, and J. McBrewster, *Levenshtein Distance*. Alphascript Publishing, 2009.

Meng Zheng Meng Zheng is a researcher in the Vision and Robotics group at United Imaging Intelligence, Cambridge, MA. She earned the Ph.D. degree in Electrical, Computer, and Systems Engineering from Rensselaer Polytechnic Institute. She received M.S. and B.Eng. degrees from the School of Information and Electronics in the Beijing Institute of Technology, China. Her research interests include vision and machine learning with a focus on image retrieval and person re-identification.



Srikrishna Karanam Srikrishna Karanam is a researcher in the Vision and Robotics group at United Imaging Intelligence, Cambridge, MA. He has a Ph.D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute. His research interests include computer vision and machine learning with a recent focus on robust and explainable perception.

Richard J. Radke Richard J. Radke joined the Electrical, Computer, and Systems Engineering department at Rensselaer Polytechnic Institute in 2001, where he is now a Full Professor. He has B.A. and M.A. degrees in computational and applied mathematics from Rice University, and M.A. and Ph.D. degrees in electrical engineering from Princeton University. His current research interests involve computer vision problems related to human-scale, occupant-aware environments, such as person tracking and re-identification with cameras and range sensors. Dr. Radke is affiliated with the NSF Engineering Research Center for Lighting Enabled Systems and Applications (LESA), the DHS Center of Excellence on Explosives Detection, Mitigation and Response (ALERT), and Rensselaer's Experimental Media and Performing Arts Center (EMPAC). He received an NSF CAREER award in March 2003 and was a member of the 2007 DARPA Computer Science Study Group. Dr. Radke is a Senior Member of the IEEE and a Senior Area Editor of *IEEE Transactions on Image Processing*. His textbook *Computer Vision for Visual Effects* was published by Cambridge University Press in 2012.