This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



# From the Lab to the Real World: Re-Identification in an Airport Camera Network

Octavia Camps, *Member, IEEE,* Mengran Gou, *Student Member, IEEE,* Tom Hebble, *Student Member, IEEE,* Srikrishna Karanam, *Student Member, IEEE,* Oliver Lehmann,Yang Li, *Student Member, IEEE,* Richard J. Radke, *Senior Member, IEEE,* Ziyan Wu, *Student Member, IEEE,* and Fei Xiong, *Student Member, IEEE*

***Abstract*—Over the past ten years, human re-identification has received increased attention from the computer vision research community. However, for the most part, these research papers are divorced from the context of how such algorithms would be used in a real-world system. This paper describes the unique opportunity our group of academic researchers had to design and deploy a human re-identification system in a demanding real-world environment: a busy airport. The system had to be designed from the ground up, including robust modules for real-time human detection and tracking, a distributed, low-latency software architecture, and a front-end user interface designed for a specific scenario. None of these issues are typically addressed in re-identification research papers, but all are critical for an effective system that end users would actually be willing to adopt. We detail the challenges of the real-world airport environment, the computer vision algorithms underlying our human detection and re-identification algorithms, our robust software architecture, and the ground-truthing system required to provide training and validation data for the algorithms. Our initial results show that despite the challenges and constraints of the airport environment, the proposed system achieves very good performance while operating in real time.**

***Index Terms*—Re-identification, camera network, video analytics.**

## I. Introduction

LARGE networks of cameras are ubiquitous in urban life, especially in densely populated environments such as airports, train stations, and sports arenas. For cost and practicality, most cameras in such networks are widely spaced, so that their fields of view are non-overlapping. Automatically matching objects, especially humans, that re-appear across different cameras in such networks is a key research question in computer vision (e.g., [1]–[3]).

In recent years, the fundamental research question has been distilled into the *human re-identification* or *re-id* problem. That is, given a cropped rectangle of pixels representing a human in one view, a re-id algorithm produces a similarity score for each candidate in a gallery of similar cropped human rectangles from a second view. Computer vision research in re-id largely focuses on two issues. The first is feature selection [4]–[8], i.e., determining effective ways to extract representative information from each cropped rectangle to produce descriptors. The second is metric learning [9]–[14], i.e., determining effective ways to compare descriptors from different viewpoints. Feature selection and metric learning should work together so that images of the same person from different points of view yield high similarity while images of different people yield low similarity. Re-id algorithms are typically validated on benchmarking datasets agreed upon by the academic community, notably the VIPeR [5], ETHZ [6], and i-LIDS MCTS [15] datasets.

However, feature selection and metric learning only represent two aspects of creating an effective real-world re-id algorithm. In practice, a re-id system must be fully autonomous from the point that an end user draws a rectangle around a person of interest to the point that candidates are presented to them. This implies that the system must automatically detect and track humans in the field of view of all cameras with speed and accuracy. The candidates in the re-id gallery in practice are thus automatically generated and are typically much lower-quality than the hand-curated gallery of a benchmark dataset; in fact, many candidate rectangles may not even represent humans! Furthermore, in a typical branching camera network, the camera in which the target reappears is unknown, so there are actually several separate galleries to search. The timing of the reappearance is also unknown; the galleries will be constantly updated with new candidates over the course of minutes or hours instead of presented to the algorithm all at once.

Additionally, the deployment of a re-id algorithm in a real-world environment faces many practical constraints not typically encountered in an academic research lab. In contrast to recently-purchased, high-quality digital cameras, a legacy surveillance system is likely to contain low-quality, perhaps even analog, cameras whose positions and orientations cannot be altered to improve performance. The video data collected by cameras in the network is likely to be transmitted to secure servers over limited-bandwidth links, and these servers are likely to have limited storage since many cameras' data must be compressed and archived. These servers are also likely to be closed off from the internet, so that any algorithm upgrades and testing must be physically done on-site. Since the algorithm must run autonomously, a robust, crash-proof

O. Camps, M. Gou, T. Hebble, O. Lehmann and F. Xiong are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: camps@neu.edu).

S. Karanam, Y. Li, R.J. Radke and Z. Wu are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: rjradke@ecse.rpi.edu).

Corresponding author: R.J. Radke. 

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



software architecture is required that takes advantage of any possible computational advantage (e.g., parallel or distributed processing) while still guaranteeing low latency. On the front end, the algorithm must run in real time, updating a ranked list of matching candidates as fast as they appear in each potential camera, and the results must be presented to the user in an easy-to-use, non-technical interface.

This paper describes the unique opportunity our team had to design and deploy a real-world re-identification algorithm in an airport, in which we had to surmount the above challenges. Our testbed consisted of three cameras, one located just after a security checkpoint in which the subject of interest is "tagged", and two located at the entrances to different concourses, in one of which the subject will re-appear. The entire system operates using the airport's network infrastructure in real time. We begin by describing important practical considerations of the airport environment in Section II. Section III describes our solutions to the human detection and tracking, feature selection, and metric learning problems for re-id. These algorithms are implemented in a modular, low-latency software architecture based on the open standard Data Distribution Service (DDS) middleware [16], described in Section IV. To train the computer vision algorithms for the airport cameras and validate our results on stored data, we undertook a substantial semi-automated ground-truthing effort, discussed in Section V. Section VI reports experimental results from the on-site system deployed at our airport testbed, as well as results from offline data that validate the choice of the algorithm components. Finally, Section VII concludes the paper with discussion and plans for future work. This paper extends an earlier version of our work presented in Li et al. [17].

## II. Design of a "Tag and Track" System for the Real World

In this section we present an overview of the real-world challenges we faced when designing and implementing a "tag and track" surveillance system for a medium-size airport in the United States. The system was designed to assist Transportation Security Administration officers (TSOs) monitoring the Cleveland Hopkins International Airport (CLE, Cleveland, Ohio, USA), while using their existing surveillance camera network.

The specifications for the system demanded the ability to manually "tag" a person of interest in a video feed, and automatically track the tagged individual across the camera network, in real time. Thus, the system was designed with a front-end user interface to allow a TSO to select a video feed and tag an individual. In addition, the system was designed to be able to detect possible candidates in the remaining views, track and compare them against the tag, and present the results to the TSO in a visual interface in a timely fashion.

The design of the "tag and track" system incorporates several modules addressing challenging problems in computer vision, such as human detection, tracking, and re-identification. These modules need to work in parallel and communicate with each other, reliably, in real time, and use data from the existing surveillance video network. As described next, these requirements imposed additional challenges that had to be addressed while designing, implementing, deploying and testing the system at the CLE airport.

### A. Data Transfer, Storage, and Collection

Figure 1 illustrates a high-level overview of the "tag and track" system, showing the data flow across its components. An important characteristic of airport security systems is that, unlike most traditional surveillance networks, the data must be always transmitted through *secure* high-bandwidth networks. As a result, the whole system needs to work in a local Ethernet with no access to the outside Internet. That is, only workstations connected to this local Ethernet are allowed access to the video data. The impact of this fact on the design process of the system was a very significant increase in cost, both in terms of time and dollars. Since this policy precludes remote debugging and testing of the software, the design cycle consisted of first collecting small sets of data on-site, developing and testing software in the lab using these recorded datasets, and making trips to CLE to install and test the software on-site. Furthermore, due to the sensitive nature of the data, all recorded data had to be first approved by the airport authorities before it could be taken to the labs, severely limiting the amount of data that could be collected for our purposes.

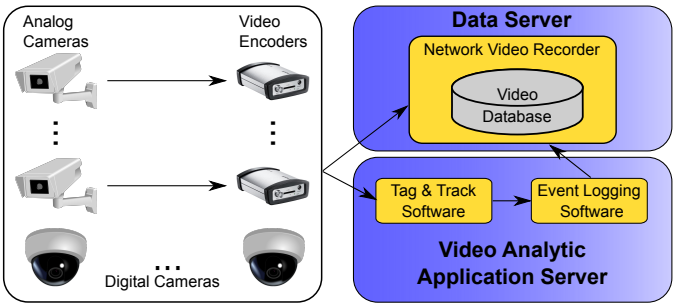



Fig. 1. High-level system design of our airport human re-identification solution.

The camera network at the CLE airport, like most real-world networks, is equipped with a heterogeneous mix of analog and digital cameras. Indeed, most of the existing infrastructure consists of analog cameras, necessitating the installation of video encoders to convert their feeds to the H.264 standard with a $704 \times 408$ resolution at 29.97 frames per second. In particular, we used Bosch VIP X1 XF video encoders for this purpose.

All the data and metadata generated by the Bosch encoders and digital cameras is transmitted through the secure network to the analytic software modules and to an auxiliary research Network Video Recorder application that runs a data server to store encoded video data, which is overwritten approximately every week. In addition, to facilitate systematic performance evaluation, every tenth frame is recorded at the processing workstation. As mentioned above, all the recorded data and events are reviewed by security officers before they can be brought back to the lab for analysis. Finally, the core of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



system consists of the video analytic software developed to acquire video feeds directly from the encoders and to perform the tracking and re-identification tasks in real time.

### B. Poor Data Quality and Challenging Environments

Real-world surveillance data is more challenging than research oriented databases used to benchmark algorithms for tracking and/or re-identification. Unfortunately, many of the legacy analog cameras in the airport network provide poor quality video, corrupted by heavy noise and often out of focus, as illustrated in the sample images shown in Figure 2. Additionally, illumination conditions can vary significantly from camera to camera and even for the same camera (near windows) due to the time of day or weather conditions. Other factors that we found particularly challenging include that in many places the floor is highly reflective, making the problem of foreground detection harder, and that the videos show periodic temporal jitter that needs to be taken into account during tracking [18].

Finally, airports can be crowded, making the tasks of human detection and tracking during heavy traffic even harder. In particular, maintaining accurate trajectories for each person in this type of environment can be very challenging. We discuss our strategies to address these challenges in Section III.

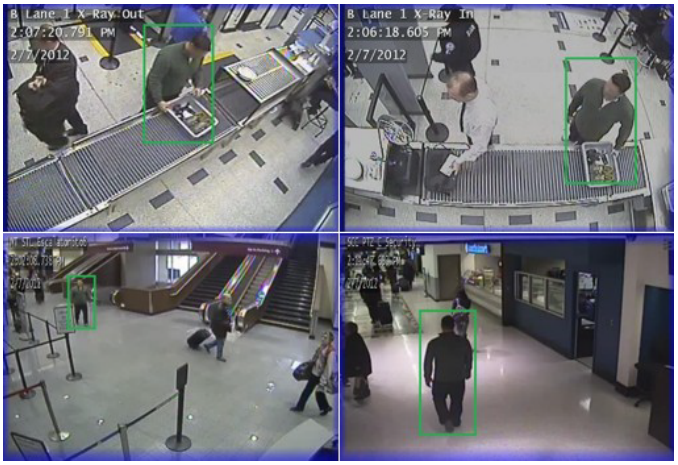



Fig. 2. Sample images from airport camera videos.

### C. Camera Positioning and Traffic Flow

Surveillance cameras used in large public spaces are widely spatially distributed, so the network often has large "blind regions". Moreover, unlike cameras used in standard re-id databases, airport cameras are often oriented at sharp angles to the floor ($\sim$45°).

Thus, successfully tracking a target across the airport network hinges on solving the challenging problem of re-identifying a target using images with severe perspective distortion and taken from very different viewpoints, as illustrated in Figure 2. This problem is complicated even further by the fact that the traffic flow of humans in an airport is hard to predict with high certainty. In an airport, there are no predefined routes since there are multiple alternatives to go from one location to another. Moreover, people can retrace their steps, walk in or out through exits not covered by the camera network, spend long periods of time in shopping or eating areas, or even change clothing while out of the view of the network. All of these factors make it difficult to reliably use appearance or transit time models in this scenario.

## III. Algorithm Overview

In this section, we describe the key computer vision aspects of our deployed system: human detection and tracking, feature selection, and descriptor comparison for re-identification. Figure 3 illustrates the main steps of the process.

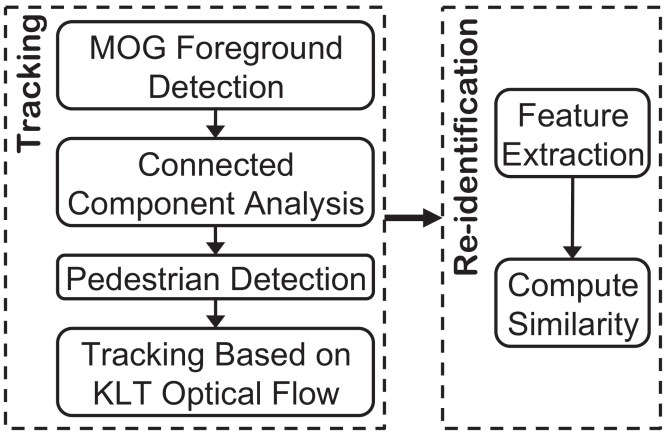



Fig. 3. Block diagram outlining our human re-identification algorithm.

### A. Detection and Tracking

The first step is using mixtures of Gaussians (MoG) [19] to detect foreground pixels and group them into blobs; the bounding boxes of these blobs define regions of interest (ROIs). ROIs with small sizes or impossible locations are discarded. Each viable ROI is input to the aggregated channel features human detector of Dollár et al. [20], as illustrated in Figure 4. This detector uses a boosted decision tree classifier to rapidly generate human candidates. We found it was important to train a specific classifier for each camera in the network to obtain good results, which was accomplished using person images from each camera (obtained using the ground-truthing tool in Section V) and randomly sampled background images (to create negative samples). The human detection runs at several scales within each ROI, resulting in a set of candidate detections of different sizes within each foreground blob. Since our system must run in real time, it was critical to restrict the candidate search to only viable ROIs, resulting in a human detector that runs at about 100 frames per second.

Our approach to tracking the detected human candidates is twofold. First, we perform tracking-by-detection in each frame as described above. Second, another set of candidate bounding boxes is generated in each frame by predicting the bounding box locations of tracked humans from the previous frame. This prediction is made by detecting low-level FAST corner features [21] in each previous bounding box, removing features estimated to belong to the background [18], estimating

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



Fig. 4. Human detection example using MoG foreground detection to reduce computational complexity.

the motion vector for each feature with the KLT tracker [22], and averaging the resulting motion vectors to update the location of the bounding box in the current frame.

The tracking-by-detection and motion-prediction bounding boxes are merged at the current frame to produce a final set of human detections as follows. We compute the intersection of each tracking-by-detection bounding box with each motion-prediction bounding box and find the maximum ratio between the area of intersection and the area of the smaller bounding box. The new tracking-by-detection box is associated with the corresponding motion-predicted box if this ratio is above a predefined threshold (in our experiments, we used 0.8); otherwise, it is used to initialize a new track. Motion-predicted bounding boxes not matching any tracking-by-detection box in the previous frame are retained if both their aspect ratio and location in the frame are plausible. Figure 5 illustrates the idea.

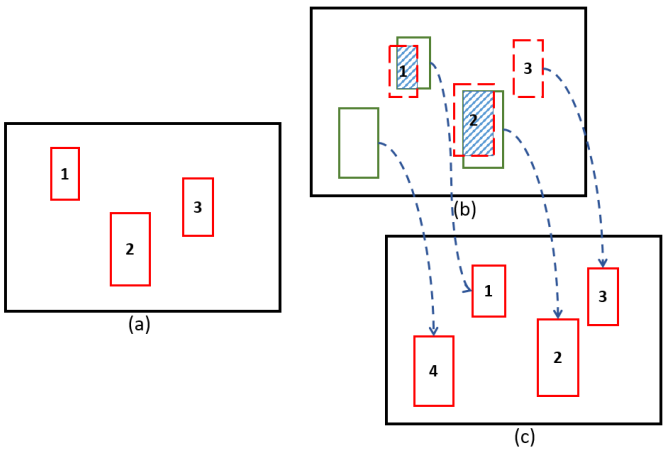



Fig. 5. (a) Tracking-by-detection bounding boxes from previous frame.(b) Predicted bounding boxes using motion vector propagation from the previous frame (dashed, red) and new tracking-by-detection candidates (solid, green). (c) Final bounding boxes for current frame created by merging the two detections.

Overall, the detection and tracking algorithms are tuned to produce a large number of human candidates in each camera for the subsequent re-id algorithms; we err on the side of allowing false alarms (i.e., poor detections or inaccurate tracks) as opposed to tolerating missed detections. This is important since the re-id algorithm can never recover if the tagged person of interest is missed in a subsequent camera, while we assume that occluded or poor-quality human detections will never rise to the top of the rank-ordered re-id candidate list.

We note that while some re-id algorithms describe humans in terms of body-based models [23] or gait analysis [24], rectangular bounding boxes are the standard method for describing people for re-id, and are computationally lightweight. Also, while several sophisticated human and object tracking algorithms exist [25]–[29], our choice of the KLT tracker is motivated primarily by its high speed of operation. Real-time tracking in several cameras simultaneously is critical for the success of the overall system.

### B. Re-identification

The re-identification process has three key steps. First, a feature descriptor needs to be extracted from each candidate detection. Second, given a pair of descriptors $\mathbf{X}_{\text{target}}$ and $\mathbf{X}_j$ (one from the tagged target and the other from the $j^{th}$ candidate detection), we must compute an appropriate similarity score

$$s_j = f(\mathbf{X}_{\text{target}}, \mathbf{X}_j) \tag{1}$$

to compare them. Finally, by ranking the similarity scores $\{s_j, j = 1, \ldots, n\}$ in each frame, an ordered list of "preferred" candidates to be shown to the user is generated.

Re-id researchers have proposed several different schemes for feature extraction. An early and popular scheme was proposed by Gray and Tao [5], in which each rectangle was divided into several horizontal stripes and color and texture histograms were extracted from each stripe. Subsequently, Bazzani *et al.* [4] proposed an accumulation of local features that model the overall chromatic content, the spatial arrangement of color, and the recurrence of textured patterns. These features were then weighted by taking appropriate symmetry and asymmetry information into account. Ma *et al.* [30] encoded descriptors comprised of pixel spatial location, intensity, and gradient information into Fisher vectors [31] and demonstrated impressive performance gains. Zhao *et al.* [32] densely divided each image into several patches and extracted color histograms and SIFT [33] features from each patch, producing so-called dense features. Recently, Liao *et al.* [34] proposed a very powerful feature representation based on locally maximal horizontal occurrence of color and texture information that achieved state of the art results.

In our work, we adopted the approach of Gray and Tao [5], which is (1) particularly suitable for the low-resolution candidate rectangles generated in the airport system and (2) enables features to be computed very quickly in an online fashion. Furthermore, color and texture histograms have also been empirically shown to provide good input to metric learning [35], [36].

Following this approach, each rectangle is divided into 6 horizontal strips. Inside each strip, 16-bin histograms are computed over 8 color channels (RGB, HSV, and CbCr) and 19 texture channels (including the response of 13 Schmid filters and 6 Gabor filters). The histograms are concatenated to form a $d-$dimensional feature vector for each candidate, where $d = 2592$.

Given a track of images for the target and each candidate, we extract features for each image as described above. Let

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



$\mathbf{x}_t^i \in \mathbb{R}^d$, $i = 1, \cdots, n$ and $\mathbf{x}_j^k \in \mathbb{R}^d$, $k = 1, \cdots, m$ denote the $n$ feature vectors of the target and the $m$ feature vectors of the $j^{th}$ candidate, respectively. We then project each of these feature vectors to a learned discriminative space using a projection matrix $\mathbf{P} \in \mathbb{R}^{\hat{d} \times d}$. Specifically, $\hat{\mathbf{x}}_t^i = \mathbf{P}\mathbf{x}_t^i$ and $\hat{\mathbf{x}}_j^k = \mathbf{P}\mathbf{x}_j^k$. We then determine $\mathbf{X}_{\text{target}} \in \mathbb{R}^{\hat{d}}$ and $\mathbf{X}_j \in \mathbb{R}^{\hat{d}}$ as the mean feature vector in the projected feature space for the target and each candidate. Specifically,

$$\mathbf{X}_{\text{target}} = \frac{1}{n} \sum_{i=1}^{n} \hat{\mathbf{x}}_t^i$$

$$\mathbf{X}_j = \frac{1}{m} \sum_{k=1}^{m} \hat{\mathbf{x}}_j^k$$

Finally, the similarity score $s_j$ is computed as

$$s_j = \mathbf{w}^\top |\mathbf{X}_{\text{target}} - \mathbf{X}_j| \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^{\hat{d}}$ corresponds to a metric learned to compensate for the differences between the target camera and the candidate camera.

Next, we discuss some prior work in metric learning followed by the procedure we employed to learn the feature space projection matrix $\mathbf{P}$ and the weight vector $\mathbf{w}$.

Like feature extraction, metric learning has also been an active area of research in the re-identification community. The goal of metric learning is to learn a feature space where feature vectors of images belonging to the same person in different cameras stay close whereas those corresponding to different people are far apart. This goal is typically translated to a mathematical formulation involving pairwise constraints on the feature vectors. Prosser *et al.* [12] used the above pairwise constraints to formulate a RankSVM model, learning a weight vector that was used to rank gallery candidates. Mignon and Jurie [10] employed a generalized logistic loss minimization formulation while enforcing pairwise similarity and dissimilarity constraints. Zheng *et al.* [37] formulated a relative distance comparison problem and learned the distance metric using a logistic objective function in a soft margin framework. Xiong *et al.* [13] extended some of these popular algorithms to accommodate the use of the kernel trick. Based on equivalence constraints, Koestinger *et al.* [38] proposed to learn the covariance matrix in Mahalanobis distance directly from the feature difference between image pairs. Wang *et al.* [39] approached the problem from a transfer learning perspective, learning a discriminative and shared feature space in a multi-task learning framework. The learned space was then used to perform re-id in camera views containing little training data, demonstrating the transferability of the feature space. Finally, most recently, Liao *et al.* [34] extended the above work by projecting features into a discriminant subspace simultaneously.

In our system, the projection matrix $\mathbf{P}$ is learned using Local Fisher Discriminant Analysis (LFDA), which has shown promising results for re-id [11]. This choice was primarily motivated by the real-world nature of our problem. Since we track each candidate, we have a sequence of images for each person. The naive way most metric learning methods discussed above deal with such data is by considering the average feature vector of all the available data for each person. However, such feature averaging can result in the loss of discriminative information available in the image sequence. On the other hand, LFDA maximizes the between-class scatter of the data while minimizing the within-class scatter. Intuitively, this means that all the feature vectors belonging to the same person are brought close together while the feature vectors of different people are pushed far apart, which is exactly what we wish to achieve. Since LFDA does this without the need to consider the average feature vector, it is particularly suitable to our problem setting.

Formally, given the $m_j$ gallery feature vectors $\mathbf{g}_j^i$, $i = 1, \cdots, m_j$ and the $n_j$ probe feature vectors $\mathbf{p}_j^i$, $i = 1, \cdots, n_j$ of the $j^{th}$ person in the training set, we construct the feature matrix $\mathbf{F} = \begin{bmatrix} \{\mathbf{g}_j^i\} & \{\mathbf{p}_j^i\} \end{bmatrix}$. In LFDA, locality preserving projections [40] are used to ensure that the feature vectors of each person are close in the transformed space, thereby preserving the local structure of the data. To this end, we define an affinity matrix $\mathbf{A}$ that captures the closeness of the feature vectors $\mathbf{F}_a$ and $\mathbf{F}_b$, where $\mathbf{F}_a$ is the $a^{th}$ column of $\mathbf{F}$. The $k$-nearest neighbors rule ($k = 7$) is used to determine this closeness. The values of the affinity matrix are defined as

$$\mathbf{A}_{ab} = \begin{cases} 1 & \text{if } \mathbf{F}_a \text{ is close to } \mathbf{F}_b \\ 0 & \text{otherwise} \end{cases}$$

The within-class and between-class scatter matrices are then defined as

$$\mathbf{S}_w = \frac{1}{2} \sum_{a,b=1}^{N} \mathbf{A}_{ab}^w (\mathbf{F}_a - \mathbf{F}_b)(\mathbf{F}_a - \mathbf{F}_b)^\top$$

$$\mathbf{S}_b = \frac{1}{2} \sum_{a,b=1}^{N} \mathbf{A}_{ab}^b (\mathbf{F}_a - \mathbf{F}_b)(\mathbf{F}_a - \mathbf{F}_b)^\top$$

where $\mathbf{A}_{ab}^w$ and $\mathbf{A}_{ab}^b$ are defined as

$$\mathbf{A}_{ab}^w = \begin{cases} \frac{\mathbf{A}_{ab}}{n_c} & \text{if class}(\mathbf{F}_a) = \text{class}(\mathbf{F}_b) = c \\ 0 & \text{if class}(\mathbf{F}_a) \neq \text{class}(\mathbf{F}_b) \end{cases}$$

$$\mathbf{A}_{ab}^b = \begin{cases} \mathbf{A}_{ab}(\frac{1}{N} - \frac{1}{n_c}) & \text{if class}(\mathbf{F}_a) = \text{class}(\mathbf{F}_b) = c \\ \frac{1}{N} & \text{if class}(\mathbf{F}_a) \neq \text{class}(\mathbf{F}_b) \end{cases}$$

where $n_c$ denotes the number of available feature vectors for the person in the training set with index $c$. Finally, the feature space transformation matrix $\mathbf{P}$ is learned as

$$\mathbf{P} = \underset{\mathbf{P}}{argmax}\ \text{trace}\{(\mathbf{P}^\top \mathbf{S}_w \mathbf{P})^{-1} \mathbf{P}^\top \mathbf{S}_b \mathbf{P}\}$$

After learning the matrix $\mathbf{P}$, we compute the mean feature vector for each person in the training set as $\bar{\mathbf{g}}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{P}\mathbf{g}_j^i$ and $\bar{\mathbf{p}}_j = \frac{1}{m} \sum_{i=1}^{m} \mathbf{P}\mathbf{p}_j^i$.

To learn the weight vector $\mathbf{w}$, we employ the RankSVM formulation of [12]. The core idea is to minimize the norm of a vector $\mathbf{w}$ that satisfies the following ranking relationship:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



$$\mathbf{w}^\top(|\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_i| - |\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_j|) > 0, \quad i, j = 1, 2, \ldots, K \text{ and } i \neq j$$

where $K$ is the number of people in the training set. The RankSVM method learns $\mathbf{w}$ by solving the following minimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{w},\xi}(\tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{K}\xi_i) \\ &s.t.\ \mathbf{w}^\top(|\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_i| - |\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_j|) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{3}$$

where $C$ is a margin trade-off parameter and $\xi_i$ is a slack variable.

It should be noted that while the process of learning $\mathbf{P}$ and $\mathbf{w}$ is time-consuming, it is done offline. On the other hand, the on-line re-id process is extremely fast since it only involves a vector inner product.

## IV. System Architecture

We approached the deployment of our re-identification algorithms at the airport with several criteria in mind.

- **Modular Architecture:** The framework must define high-level functional blocks and the communication among them to allow the easy and reliable interchange of functional components as research yields new algorithms and approaches.
- **Real-time Operation:** Communication and data transfer between framework components must not prevent the real-time operation of the complete system.
- **Task-level Parallelism:** To perform full functionality in real time, the system must allow for the framework components to operate in parallel while ensuring that all the modules are working synchronously.
- **Language-agnostic API:** Efficient multi-institutional collaboration requires accommodating a variety of code development environments. For example, the framework must support native and managed processes written in C++ and C#.
- **Real-time Logging:** All results must be recorded to allow for later performance evaluation, without inhibiting real-time operation.
- **Simulated Environment:** The framework must have the ability to simulate deployment using recorded videos to enable reliability testing and algorithm performance evaluation prior to actual deployment.

For these reasons, we selected the open standard Data Distribution Service (DDS) middleware [16] to handle inter-process communication and guarantee compatibility as new components are added to the system. DDS is designed for real-time applications requiring low latency and high throughput.

Although our system uses shared memory exclusively, the physical transport used by DDS is configured at runtime using a transport type-agnostic API allowing application components to be distributed across multiple machines if necessary. To minimize communication overhead, DDS contains automatic peer discovery and peer-to-peer data transfer without needing to run additional message brokers or servers. Custom data structures are defined using an interface description language (IDL) that closely resembles C++ class definitions. These structure definitions correspond to a common data representation that allows access from many programming languages including C++, C#, and Java.

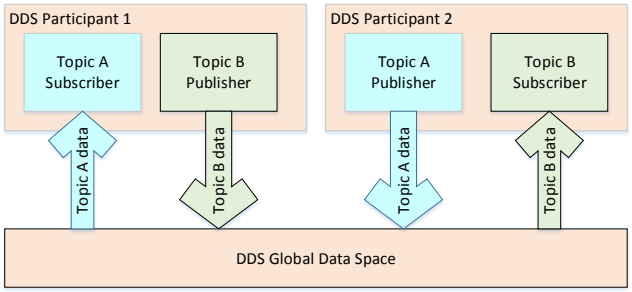



Fig. 6. Block diagram showing participating entities in the publish-subscribe communication model used by DDS.

DDS uses a loosely-coupled publish-subscribe communication model. In this model, participating processes contain objects for publishing (writing) and subscribing to (reading) data from a global data space managed by DDS (Figure 6). The global data space is organized into a number of "topics" defined by a unique pair of name and IDL-defined data type. To access the global data space, programs merely inform DDS of the topic name and data types they would like to read and/or write to; the creation of new topics is handled automatically by DDS. From a programming perspective, the behavior of a participant in the publish-subscribe model is independent of other participants. For example, the process responsible for publishing video frame data does not need to account for which or how many other processes are reading the data. DDS is configured at runtime by reading an XML file containing Quality of Service (QoS) policies to control aspects of how and when data is distributed by the middleware. QoS can control attributes such as the maximum size of global data space or how much data for each topic can be available to subscribers to read. These attributes of DDS help ensure reliability as new components are added while keeping the framework flexible enough to handle new methods from our research. In addition, the DDS implementation provides tools for the recording and playback of DDS communications allowing us to examine not only the re-id results but any communication within the framework.

Figure 7 illustrates the DDS architecture corresponding to the re-identification software deployed the three-camera system installed at the airport. Each block corresponds to a separate constantly running process performing the algorithms described in Section III.

In particular, the processing pipeline contains the following modules:

1) **Candidate Detection:** The first module in the processing pipeline publishes the single frame locations of humans detected in the video source.
   - *Subscribes to:* Video frames.
   - *Publishes:* Single frame candidate locations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



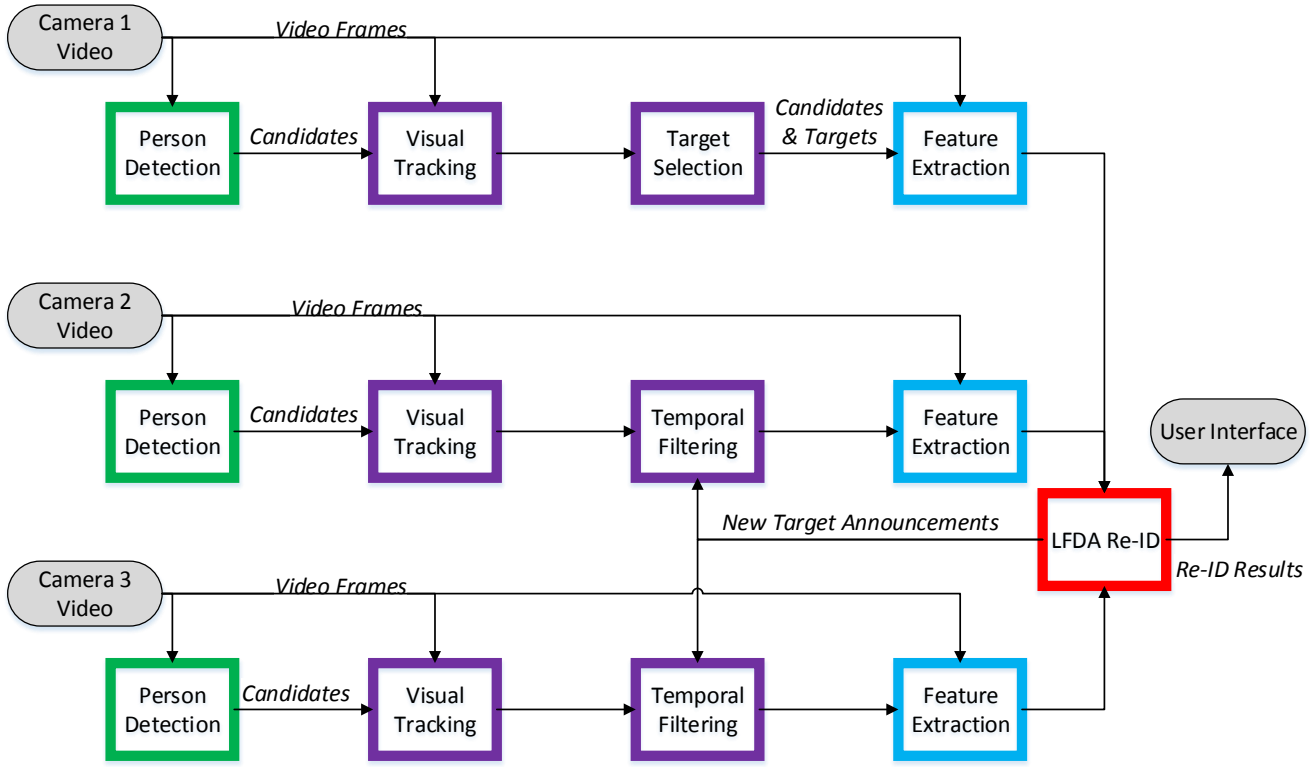



Fig. 7. Block diagram showing the re-id system architecture, including processes for Candidate Detection (green), Candidate Filtering (purple), Feature Extraction (blue), and Re-Identification (red).

2) **Candidate Filtering:** This module is used for additional processing of candidates prior to re-id, such as tracking or grouping detections known to be the same person. By subscribing to new target announcements from the Re-identification module, this module can also act as a temporal filter for potential candidates.
   - *Subscribes to:* Video frames (optional); Candidates from Candidate Detection module or other instances of Candidate Filtering module; New target announcements from the Re-Identification module (optional).
   - *Publishes:* Candidate and Target locations.
3) **Feature Extraction:** This module is responsible for preparing potential candidates and targets for re-id by calculating a vector of feature values as described in Section III-B. Since feature extraction is generally the most computationally intensive task in re-id, it is performed only on the most promising candidates that have passed the spatial and temporal filtering in the previous modules.
   - *Subscribes to:* Video frames (optional); Candidates and targets from Candidate Filtering module.
   - *Publishes:* Candidate and Target locations with identifying feature vectors.
4) **Re-Identification:** The last computer vision module is responsible for generating the final re-id results. It uses the feature vectors calculated by the previous module to compare the active target with all candidates from each camera as described in Section III-B, and provides a sorted list and difference score for each candidate.
   - *Subscribes to:* Video frames (optional); Candidates and Targets from Feature Extraction module.
   - *Publishes:* New target announcements; Re-id results.
5) **Graphical User Interface:** The final module is responsible for visualizing the re-id results using images of the target and top candidates as well as any other desired information regarding candidates and targets (e.g., video display with candidate bounding boxes). This module does not publish any data.
   - *Subscribes to:* Video frames; Candidates and Targets from Feature Extraction module; Re-id results.

## V. Data Collection and Ground Truth Generation

To develop the computer vision algorithms and system architecture described here, we required a comprehensive video footage database with high-accuracy ground truth labels for hypothesis validation, parameter tuning, and performance evaluation. In particular, we required accurate bounding boxes for humans in thousands of frames of videos from several cameras, and when possible, metadata such as gender, clothing color, motion type, and interactions with others that might be useful for future analysis.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



One strategy to achieve accurately annotated visual content is to divide the labeling task into many smaller tasks executed by a large number of people enlisted through, e.g., crowd-sourced marketplaces [41], [42]. However, crowdsourcing is not a viable practice for labeling sensitive, proprietary videos. Therefore, we opted to employ in-house, specially trained personnel to generate reliable ground truth. In our case, the limiting factor is the time required for bounding box delineation, requiring up to 3.5 hours to process one video minute for a single human without any computational intervention.

For this purpose, we designed a computer-aided ground truthing system called "Annotation Of Objects In Videos (ANchOVy)", a toolbox for cost-effective surveillance footage labeling. ANchOVy's unified graphical user interface, shown in Figure 8, was designed for an ergonomic, low-latency video labeling workflow and includes features to safeguard against worker errors (e.g., automated label propagation, continuous auto-save function, role-based content control).

ANchOVy first automatically extracts short trajectories of moving objects in the video by using a featureless tracking-by-detection method [27] implemented on graphics processing units [43]. Then, the human worker identifies and labels an object of interest in a highly sparse set of frames. Next, the missing labels are automatically inferred by connecting the previously collected short trajectories using Hankel matrices of the trajectories [28]. The worker inspects the inferred results and can take corrective actions, which will trigger a recalculation and update using the added label information. This procedure is repeated until a satisfactory label quality is achieved. Finally, the worker assigns a unique global identification number to each tracked person to facilitate algorithm design and validation for re-identification, as discussed in Section VI-A.

## VI. Experimental Results

In this section we summarize the training of the system and report the results of a set of experiments using real-world airport videos to evaluate the overall re-id performance. For these experiments, we chose to use video from three cameras located in the area after the central checkpoint area. Of the three cameras, one camera (camera A) corresponds to the central checkpoint area. The other two cameras (cameras B and C) show views of the hallways heading towards different concourses. This camera network has an interesting branching scenario in the sense that people that appear in the view of camera A can go to either of the two concourses after spending an indefinite amount of time in the central area. Since camera A corresponds to the central area, we choose this camera to tag persons of interest.

### A. System Training

Using ANchOVy, we labeled 650 tracks of 188 persons, each identified by a unique global ID, in multiple image sequences recorded across CLE's distributed camera network. The ground truth labeling process produced tightly cropped images of humans in every twelfth video frame ranging in size from $51 \times 30$ to $267 \times 212$ pixels.

The cropped images were then used to train our human detection and re-identification algorithms. We grouped the person images based on their camera view to train camera-specific decision trees for human detection as described in Section III-A. We also used the ground truth bounding boxes and global IDs to learn the feature space projection matrix $\mathbf{P}$ and the metric vector $\mathbf{w}$ for each of the two camera pairs (A, B) and (A, C), as described in Section III-B. Using five-fold cross-validation on these training images, we set the dimension of the transformed feature space $\hat{d} = 300$ for re-id in the camera pair (A, B) and $\hat{d} = 200$ for the camera pair (A, C).

### B. User Interface

For the real-time experiments, we had to design a graphical user interface (GUI) to make it easy to tag persons and score algorithm performance, illustrated in Figure 9. Figure 9a shows tagging a person of interest in camera A, Figures 9b and c show detection and tracking of candidates in cameras B and C, and Figure 9d shows the final re-identification results that are presented to the user of the interface. In this particular example, we note that the person of interest re-appeared in camera C, and was successfully re-identified at rank 1.

### C. Experimental Protocol

To evaluate the performance of the system, we deployed it at the Cleveland Hopkins International airport and ran experiments using live video feeds, recording the real-time re-identification results. In each experiment, a target person was manually tagged in camera A, and re-identified in cameras B or C. A sample of 20 such target images is shown in Figure 10. We set a re-appearance time window of 3 minutes, i.e., we waited for 3 minutes for the target person to re-appear. We define a *valid* experiment as one in which the person of interest re-appeared within the set time window. An *invalid* experiment is one in which the person of interest did not re-appear in either camera B or camera C within the set time window. In total, across approximately 15 hours of run-time, we performed 198 experiments, out of which 151 experiments were valid. We use only the valid experiments to report performance statistics.

### D. Performance Statistics

Of the 151 valid experiments, there were 94 cases in which the person of interest re-appeared in camera B and 57 cases of re-appearance in camera C. Since the end-users of the system are unlikely to scroll through pages of candidates, the correct match should appear within an easily scannable "line-up" for the system to be usable. To this end, we report the real-time performance of the system in terms of the rank-n performance where $n \in \{5, 10\}$, i.e., the percentage of experiments in which the tagged person of interest was re-identified within the top-n rank, and stayed within the top-n rank throughout the 3-minute time window. The performance in each of the two re-appearance cameras B and C is tabulated in Table I. The cumulative match characteristic (CMC) curves for each of the two re-appearance cameras are shown in Figure 11.

In the on-site experiments, we observed serious compression artifacts from the video encoder while running the video at 30

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



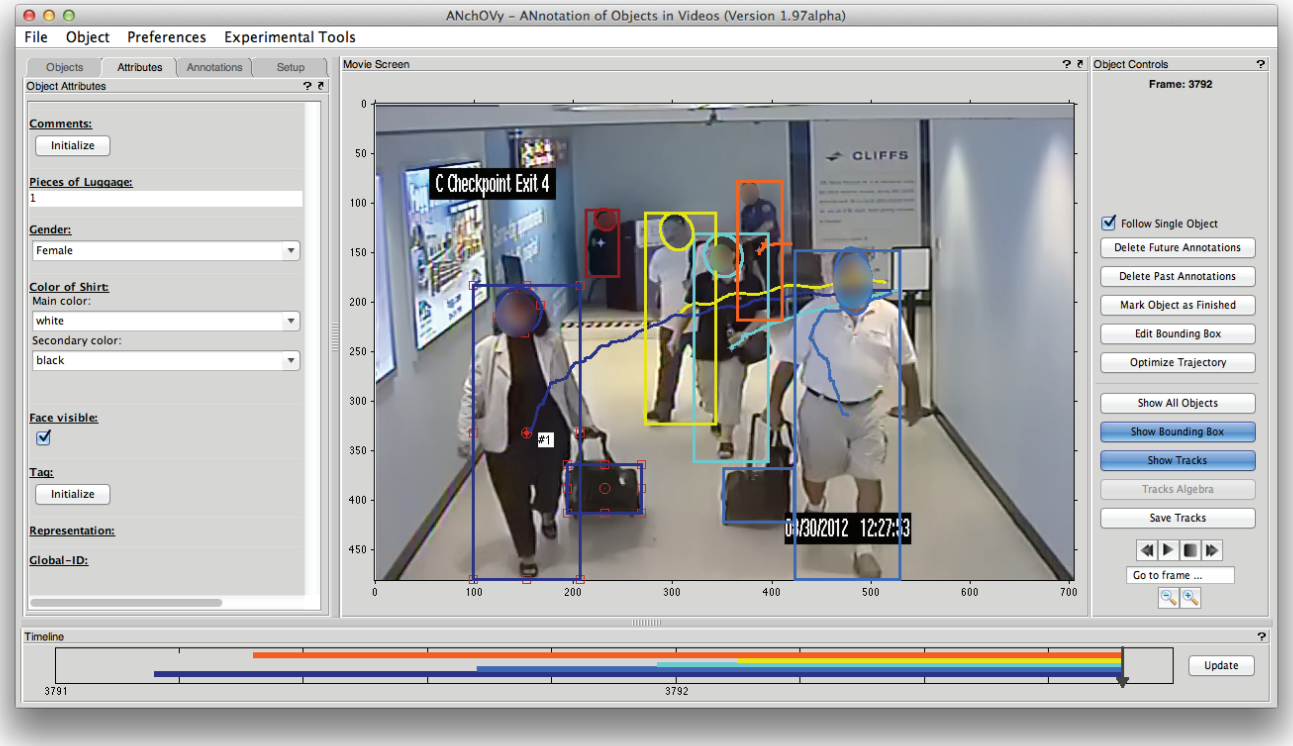



Fig. 8. ANchOVy's graphical user interface showing humans and their trajectories, spatial labels (full-body, head, and luggage bounding boxes) as well as other labels.

TABLE I
RE-IDENTIFICATION PERFORMANCE FOR ON-SITE AIRPORT EXPERIMENTS.

| Re-id camera | # experiments | rank-5 | rank-10 |
|---|---|---|---|
| B | 94 | 58.5% | 83.0% |
| C | 57 | 61.4% | 87.7% |

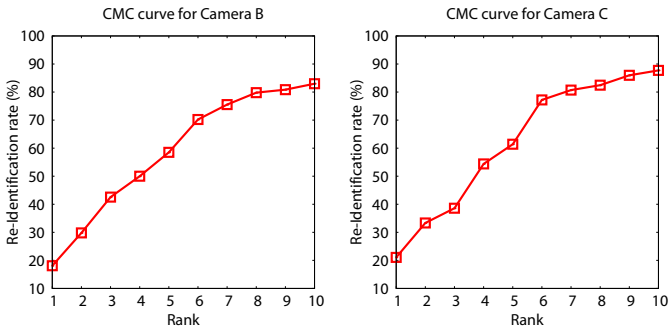



Fig. 11. Cumulative match characteristic (CMC) curves corresponding to the experiments in Table I.

frames per second. We had to reduce the frame rate to 10 frames per second to avoid this issue. Due to the relatively low frame rate of the video, there were several cases of missed detection, i.e., cases in which the tagged person of interest was not detected upon re-appearance in camera B or C. Specifically, in camera B, out of the 94 valid experiments, there were 12 cases in which the person of interest was not detected. This number was 5 out of 57 valid experiments in camera C. The low video frame rate also resulted in inaccurate tracking of the FAST corner features, resulting in person tracking errors. Furthermore, in some experiments, due to the high crowd density, large occlusions also contributed to tracking errors. Since we compute the mean feature vector for the track of images available for each candidate, errors in tracking resulted in errors in downstream re-identification. These issues and their implications are explored in more detail in the next section.

### *E. Evaluating system components*

A real world system invariably involves errors in the detection and tracking modules. For the detector, these errors typically involve detections not corresponding to a person, which we call invalid detections. On the other hand, errors in the tracking module are more complicated. A person currently being tracked can be lost for a few frames before being tracked again. If these two "tracklets" are not associated to correspond to the same person, a tracking error results. In this section, we analyze the behavior of the end-to-end system in the presence and absence of these errors. To this end, we created an offline evaluation dataset. Specifically, from 15 hours of video data from the same three airport cameras as above, we randomly extracted 40 5-minute video clips. We ran each of these video clips through the entire tracking module in Figure 3. We manually annotated the detections produced by the detector as being valid or invalid, using the overlap ratio,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



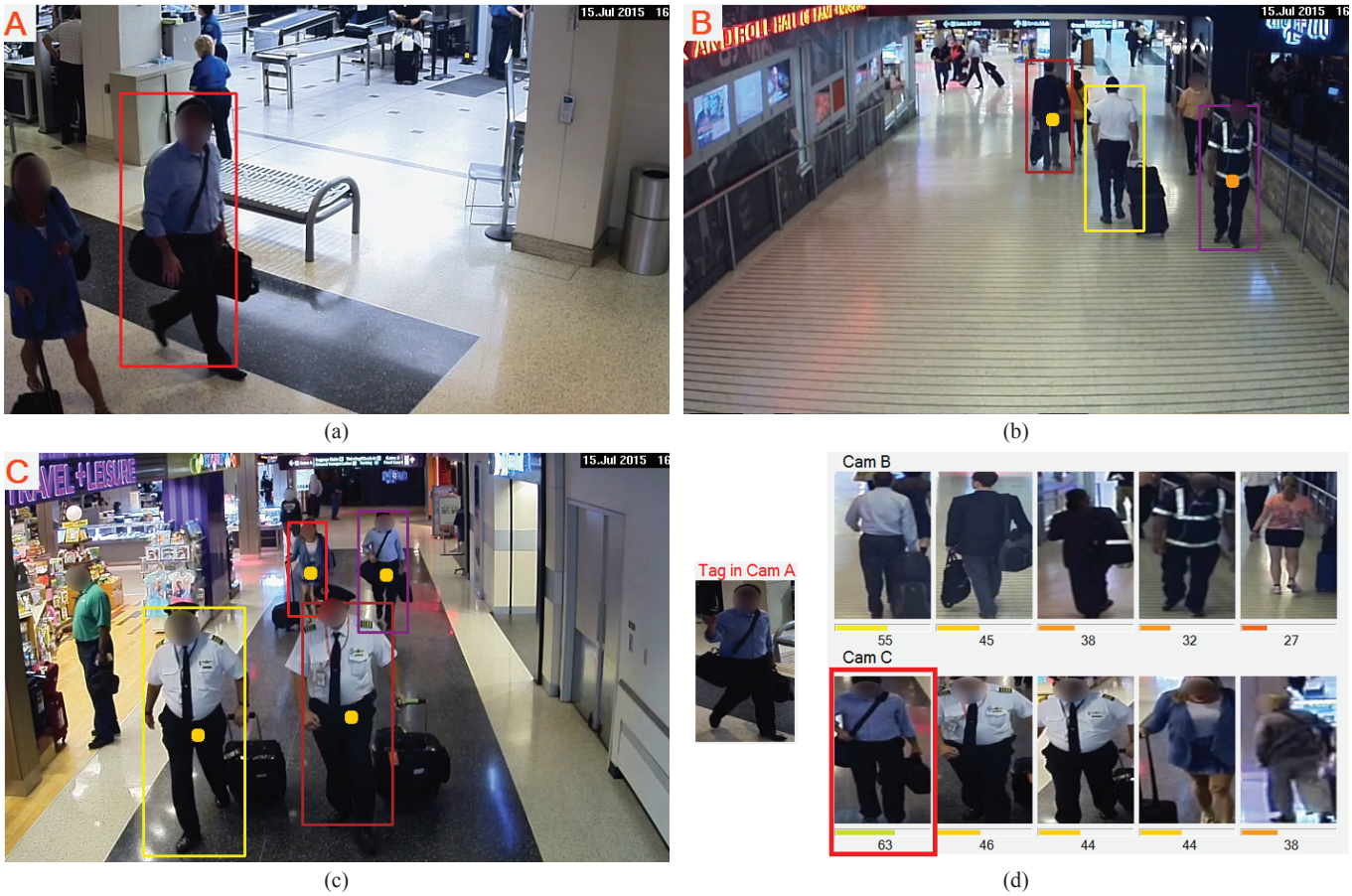



Fig. 9. Snapshots from the graphical user interface developed for the airport human re-identification task. (a) Tagging the person of interest in camera A, (b) Tracking candidates in camera B, (c) Tracking candidates in camera C, (d) Re-identification results displayed to the user (red box indicates correctly re-identified candidate).

Fig. 10. A sample of 20 targets manually tagged for system performance evaluation.

$o_r = \frac{\text{area}(D \cap G)}{\text{area}(D \cup G)}$, where D represents a detected bounding box and G represents the corresponding ground truth. We define a detection to be valid if $o_r > 0.5$. We also associated any broken intra-camera tracklets and inter-camera re-appearances manually. By varying the information in the gallery set, we defined the following four experimental protocols:

1) Include both valid and invalid detections in the gallery set; no manual intra-camera association.
2) Include only valid detections in the gallery set; no manual intra-camera association.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



3) Include both valid and invalid detections in the gallery set; manual intra-camera association in the gallery camera.
4) Include only valid detections in the gallery set; manual intra-camera association in the gallery camera.

Since the real-world end-to-end system typically has both valid and invalid person detections as well as tracking errors, Protocol 1 above mimics the system we deployed at the airport. Protocol 2 discards invalid detections, assuming an ideal person detection algorithm that only outputs valid detections, thereby helping evaluate the impact of the detector module. Protocol 3 involves manual tracklet association, helping mimic an ideal tracking module. This protocol therefore helps evaluate the impact of the tracking module. Finally, Protocol 4 assumes both the detector and tracking modules are ideal. This helps us understand the best-case performance of the system when all the modules work perfectly.

We performed re-id experiments using all four protocols discussed above in both Cameras B and C. In all the experiments, we used LFDA to learn the feature space projection and rankSVM to rank the gallery candidates. The results obtained are plotted in the CMC curves shown in Figure 12.

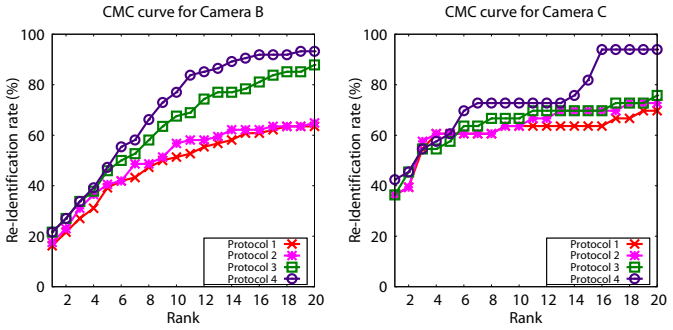



Fig. 12. CMC curves corresponding to experiments on the offline dataset with the four different evaluation protocols.

As can be seen from the results, Protocol 1 gives the worst performance at most ranks whereas Protocol 4 gives the best performance. This is natural, since Protocol 1 involves both detection and tracking errors. If we assumed ideal detector and tracking modules (Protocol 4), the best-case performance of the end-to-end system would be approximately $5.4\%$ and $6.1\%$ higher than the corresponding real-world system at rank-1 for Cameras B and C, respectively. Comparing the results of Protocols 1 and 2, invalid detections only have a marginal effect on the re-id performance, with Protocol 2 resulting in a rank-1 performance improvement of $1.4\%$ in Camera B and no improvement in Camera C, respectively. From these results, we can conclude that the learned feature space plays an important role in the overall operation of the system. Since the ideal tracking module decreases the gallery set dramatically, Protocol 3 has better performance than Protocols 1 and 2.

Next, we also analyzed the impact of the different algorithmic components used in the re-id module. To this end, we performed experiments on the same data as above with Protocol 1 and different combinations of the LFDA and rankSVM metrics. We also compared these metrics with a recently proposed method, XQDA [34], that achieves state-of-the-art performance. The results obtained are shown in the CMC curves in Figure 13. We see that using a combination of LFDA for feature space projection and rankSVM for ranking candidates results in better performance than the corresponding individual components alone. Furthermore, we see that while XQDA gives slightly better performance in Camera B, the combination of LFDA and rankSVM performs much better in Camera C.

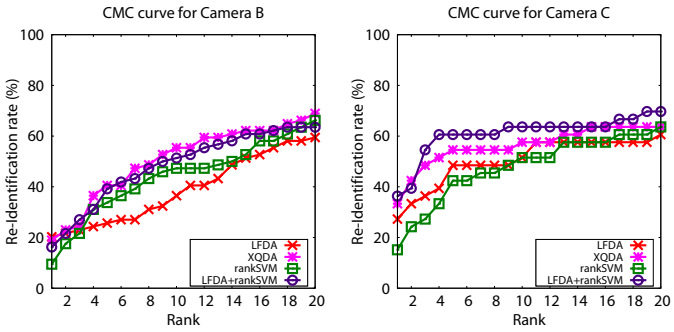



Fig. 13. CMC curves corresponding to experiments on the offline dataset to evaluate the different components used in the re-identification module.

### F. Comparison with academic benchmarks

Here, we compare our end-to-end system with how academic papers approach and evaluate re-id algorithms. Typically, academic research on re-id is evaluated on VIPeR [5] and iLIDS-VID [44], standard benchmarking datasets. The current state of the art at rank-5 for VIPeR is around $75-80\%$ [45], [46] and around $57\%$ for iLIDS-VID [44], while our system was able to achieve a rank-5 performance of about $60\%$. While we cannot compare these numbers directly, we mention several aspects of how performance on academic benchmarking datasets is different from our real-world implementation:

1) **Time-dependent gallery.** In VIPeR and iLIDS-VID, the gallery subjects are fixed. However, in our case, the goal is to re-identify a person of interest who may re-appear in the gallery camera after an indefinite amount of time. This results in a search over a gallery set that expands over time.
2) **Hand-curated gallery vs. automatically-generated gallery.** In VIPeR and iLIDS-VID, the gallery set consists of images of persons generated by a ground-truthing mechanism. However, in our implementation, the gallery is automatically constructed by using the raw outputs of human detection and tracking algorithms that generate candidates in real time.
3) **Gallery assumptions.** In VIPeR and iLIDS-VID, a key assumption is that the gallery set contains the person of interest. However, this assumption does not hold in our implementation, since (1) the person of interest may never appear in the camera generating a particular gallery, and (2) the human detection module may fail to detect the person even if they do appear.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556538, IEEE Transactions on Circuits and Systems for Video Technology



We note that the SAIVT-SoftBio dataset [47] was constructed from a multi-camera surveillance network that closely mimics the real-world re-id problem. However, the three issues raised above hold even in this case. We emphasize that for a system that works in an end-to-end fashion in real-time, people must be automatically detected on-the-fly, resulting in a dynamic gallery set that is constantly adding and removing candidates, and may not even contain the target. This is a key difference between real-world operation and academic benchmark datasets.

### G. Suggestions for a practical re-id system

We conclude this section with some suggestions for implementing a practical re-id system in a real-world setting such as the one described in this paper.

- **Focus on the end user.** We believe a primary requirement of a practical re-id system should be ease of use. It is likely that the end user will not be a computer vision expert, so a robust crash-resistant software with an easy-to-use front-end is critical.
- **Software architecture.** From our experience at the Cleveland airport, we learned that working with live real-time camera data can be extremely demanding on the system and software. The choice of powerful computer and robust software architecture to drive the system is critical.
- **Good tracking.** Based on our offline dataset analysis, the intra-camera tracking module critically affects performance. A powerful and efficient tracker would help to reduce the gallery size and improve the accuracy significantly. Since false person detections can be eliminated by the learned metrics in the re-id module, one should use a more sensitive detector to minimize missed candidates.
- **Usability testing.** While a computer vision expert can appreciate the difficulty of the problem, the constraints on speed, and the performance of our algorithm in the face of these challenges, airport staff are likely to expect the system to work perfectly most of the time (e.g., 99% performance at rank 1!). For future development, it would be important to engage end-users in the front-end design of the system from the beginning, in which the human is in the loop to visually inspect the candidates generated by the system and provide simple annotations as to their validity [48]. This would lead to evolving usability tests to evaluate end-users' comfort and calibrate their expectations.

## VII. Conclusions

We discussed several practical challenges we faced while designing, implementing, deploying and testing a real-time re-identification system in an airport. In particular, we highlighted the differences between the re-id problem as it is posed in academia and how it must be solved in practice, and presented initial results from our on-site algorithm deployment at the CLE airport.

To further improve the overall performance of our system, we plan to integrate several ideas from our "more academic" research on re-id, such as weighting the features based on the estimated pose and movement direction of the candidate prior to descriptor comparison [7], investigating personally-discriminative feature selection and comparison [7], [49], adaptively clustering feature vectors obtained from tracking prior to performing feature space projection [35], learning discriminative dictionary based representations [36], [50], and using kernel tricks to improve performance [13].

Our current system requires annotated training data in each camera to tune the human detectors and the feature space projection matrices. While it is reasonable to expect that such training data can be obtained in a critical environment such as an airport, the human effort to annotate such large datasets cannot be discounted. To this end, we will also investigate a transfer learning approach to re-id [39], in which discriminative metrics can be learned from small amounts of data in a subset of the camera views and transferred to new views.

The DDS software architecture has allowed our team to successfully evaluate many different algorithms and system configurations quickly. Since security procedures prevent remote access to the airport's camera network, installation and debugging of the system requires one or more researchers to physically visit the airport. The application framework described here has made these trips very efficient, allowing quick installation and initial testing of new components with almost no time needed for on-site debugging. We are currently tuning the robust DDS software architecture to run for days at a time and recover from crashes, and creating an intuitive user interface that allows the user to easily retain possible matches and reject others. Throughout the project, we have been able to apply lessons learned from a previous project involving a system for real-time detection of counterflow through exit lanes at the same airport [18], [43].

We also plan to further investigate the challenges of re-id in branching camera networks across very long time scales. In the scenario described here, one of the two candidate galleries will never actually contain the tagged person, while there could be a very long time lag before the gallery for the correct concourse contains an image of the person. We plan to investigate temporal and predictive models for person re-appearances in this challenging scenario, leveraging the ground-truthing framework from Section V. Separately, we are investigating a re-id scenario in a light rail environment, in which persons of interest re-appear after days instead of minutes, corresponding to a potentially huge gallery. Typical CMC curves are insufficient to characterize performance in such systems since they ignore the temporal aspect of the constantly updating gallery.

## Acknowledgments

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. Thanks to Michael Young, Jim Spriggs, and Don Kemer for supplying the airport video data. Thanks to John Beaty for managing the project, to Deanna Beirne and Rick Moore for helping to set up and maintain the described system, and to Alyssa White for coordinating the ground-truthing effort. Thanks to Vivek Singh and Arun Inanje of Siemens Corporation, Corporate Technology, for providing and configuring the system hardware.

## REFERENCES

[1] K.-W. Chen, C.-C. Lai, P.-J. Lee, C.-S. Chen, and Y.-P. Hung, "Adaptive learning for target tracking and true linking discovering across multiple non-overlapping cameras," *Multimedia*, vol. 13, no. 4, pp. 625–638, 2011.

[2] O. Javed, K. Shafique, and M. Shah, "Appearance modeling for tracking in multiple non-overlapping cameras," in *IEEE Conf. Comput. Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 26–33.

[3] X. Wang, K. Tieu, and W. E. L. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 56–71, 2010.

[4] L. Bazzani, M. Cristani, and V. Murino, "Symmetry-driven accumulation of local features for human characterization and re-identification," *Comput. Vision and Image Understanding*, vol. 117, no. 2, pp. 130–144, 2013.

[5] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Eur. Conf. Comput. Vision*, Marseille, France, 2008, pp. 262–275.

[6] W. R. Schwartz and L. S. Davis, "Learning discriminative appearance-based models using partial least squares," in *Brazilian Symp. on Comput. Graphics and Image Process.*, Rio de Janeiro, Brazil, 2009, pp. 322–329.

[7] Z. Wu, Y. Li, and R. Radke, "Viewpoint invariant human re-identification in camera networks using pose priors and subject-discriminative features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1095–1108, 2015.

[8] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-identification," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 144–151.

[9] J. Blitzer, K. Q. Weinberger, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Inform. Process. Syst.*, Whistler, BC, Canada, 2005, pp. 1473–1480.

[10] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Providence, RI, 2012, pp. 2666–2672.

[11] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Portland, OR, 2013, pp. 3318–3325.

[12] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by support vector ranking." in *Brit. Mach. Vision Conf.*, Aberystwyth, UK, 2010.

[13] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Eur. Conf. Comput. Vision*, Zurich, Switzerland, 2014, pp. 1–16.

[14] W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Colorado Springs, CO, 2011, pp. 649–656.

[15] UK Home Office, "i-lids multiple camera tracking scenario definition," https://www.gov.uk/imagery-library-for-intelligent-detection-systems, accessed: 2015-08-04.

[16] Object Management Group, "Data distribution service for real-time systems," http://portals.omg.org/dds/, accessed: 2015-08-04.

[17] Y. Li, Z. Wu, S. Karanam, and R. Radke, "Real-world re-identification in an airport camera network," in *Proc. Int. Conf. Distributed Smart Cameras*, Venezia Mestre, Italy, 2014.

[18] Z. Wu and R. J. Radke, "Improving counterflow detection in dense crowds with scene features," *Pattern Recognition Lett.*, vol. 44, pp. 152–160, 2014.

[19] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Fort Collins, CO, 1999, pp. 246–252.

[20] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014.

[21] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, 2010.

[22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Imaging Understanding Workshop*, 1981.

[23] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu, "Human re-identification by matching compositional template with cluster sampling," in *IEEE Int. Conf. Comput. Vision*, Sydney, Australia, 2013, pp. 3152–3159.

[24] M. S. Nixon, T. Tan, and R. Chellappa, *Human identification based on gait*. Springer Science & Business Media, 2010, vol. 4.

[25] X. Mei and H. Ling, "Robust visual tracking using $\ell_1$ minimization," in *IEEE Int. Conf. on Comput. Vision*, Kyoto, Japan, 2009, pp. 1436–1443.

[26] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Miami, FL, 2009, pp. 983–990.

[27] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Eur. Conf. Comput. Vision*, Firenze, Italy, 2012, pp. 702–715.

[28] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *IEEE Int. Conf. Comput. Vision*, Sydney, Australia, 2013, pp. 2304–2311.

[29] S. Karanam, Y. Li, and R. J. Radke, "Particle dynamics and multi-channel feature dictionaries for robust visual tracking," in *Brit. Mach. Vision Conf.*, Swansea, UK, 2015, pp. 183.1–183.12.

[30] B. Ma, Y. Su, and F. Jurie, "Local descriptors encoded by fisher vectors for person re-identification," in *Eur. Conf. Comput. Vision Workshops and Demonstrations*, Firenze, Italy, 2012, pp. 413–422.

[31] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *IEEE Conf. Comput. Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 3384–3391.

[32] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Portland, OR, 2013, pp. 3586–3593.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Boston, MA, 2015, pp. 2197–2206.

[35] Y. Li, Z. Wu, S. Karanam, and R. J. Radke, "Multi-shot human re-identification using adaptive fisher discriminant analysis," in *Brit. Mach. Vision Conf.*, Swansea, UK, 2015.

[36] S. Karanam, Y. Li, and R. J. Radke, "Person re-identification with discriminatively trained viewpoint invariant dictionaries," in *IEEE Int. Conf. on Comput. Vision*, Santiago, Chile, 2015, pp. 4516–4524.

[37] W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by relative distance comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 653–668, 2013.

[38] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Providence, RI, 2012, pp. 2288–2295.

[39] X. Wang, W.-S. Zheng, X. Li, and J. Zhang, "Cross-scenario transfer person re-identification," *IEEE Trans. Circuits and Systems for Video Technology*, vol. PP, no. 99, 2015.

[40] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Inform. Process. Syst. 16*, Vancouver and Whistler, Canada, 2003, pp. 153–160.

[41] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon Mechanical Turk," *Urbana*, vol. 51, no. 61, p. 820, 2008.

[42] C. Vondrick, D. Ramanan, and D. Patterson, "Efficiently scaling up video annotation with crowdsourced marketplaces," in *Eur. Conf. Comput. Vision*, Crete, Greece, 2010, pp. 610–623.

[43] T. Hebble, "Video analytics for airport security: Determining counter-flow in an airport security exit," Master's thesis, Northeastern University, 2015.

[44] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *Eur. Conf. Comput. Vision*, Zurich, Switzerland, 2014, pp. 688–703.

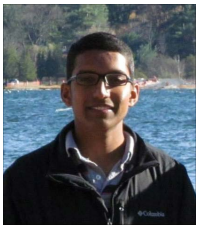
[45] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Boston, MA, 2015, pp. 3908–3916.

[46] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1846–1855.
[47] A. Bialkowski, S. Denman, S. Sridharan, C. Fookes, and P. Lucey, "A database for person re-identification in multi-camera surveillance networks," in *Int. Conf. Digital Image Computing Techniques and Applications*, Fremantle, Australia, 2012, pp. 1–8.
[48] C. Liu, C. Loy, S. Gong, and G. Wang, "Pop: Person re-identification post-rank optimisation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 441–448.
[49] Y. Li, Z. Wu, and R. Radke, "Multi-shot re-identification with random-projection-based random forests," in *IEEE Winter Conf. Applicat. Comput. Vision*, Kona, HI, 2015, pp. 373–380.
[50] S. Karanam, Y. Li, and R. Radke, "Sparse re-id: Block sparsity for person re-identification," in *IEEE Conf. Comput. Vision and Pattern Recognition Workshops*, Boston, MA, 2015, pp. 33–40.

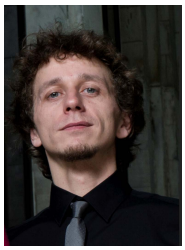
**Octavia Camps** Octavia Camps received B.S. degrees in computer science in 1981 and in electrical engineering in 1984, from the Universidad de la Republica (Montevideo, Uruguay), and M.S. and Ph.D. degrees in electrical engineering in 1987 and 1992, from the University of Washington, respectively. Since 2006 she is a Professor in the Electrical and Computer Engineering Department at Northeastern University. From 1991 to 2006 she was a faculty member at the departments of Electrical Engineering and Computer Science and Engineering at The Pennsylvania State University. In 2000, she was a visiting faculty at the California Institute of Technology and at the University of Southern California and in 2013 she was a visiting faculty at the Computer Science Department at Boston University. Her main research interests include dynamics-based computer vision, image processing, and machine learning. She is a member of IEEE.

**Mengran Guo** Mengran Gou is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering at Northeastern University. He received an M.S. degree from the Pennsylvania State University and B.Eng. degree from Harbin Institute of Technology in China. His research interests are person re-identification and activity recognition.

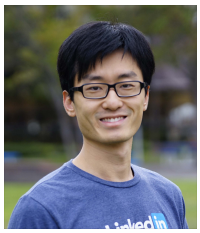
**Tom Hebble** Tom Hebble received his B.S. and M.S. degrees in Electrical Engineering from Northeastern University, Boston, Massachusetts. He is currently working as a research engineer at Scientific Systems Company Inc. His research interests include video analytics and signal processing with a focus on real-time applications.



**Srikrishna Karanam** Srikrishna Karanam is a Ph.D. student in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute, Troy NY. He received the Bachelor of Technology degree in Electronics and Communication Engineering from the National Institute of Technology Warangal in 2013 and the Master of Science degree in Electrical Engineering from Rensselaer Polytechnic Institute in 2014. His research interests include computer vision, video processing, machine learning, and optimization.



**Oliver Lehmann** Oliver Lehmann graduated as a Diplom-Ingenier with highest honors in 2010 at the Berlin Institute of Technology. He received an M.S. degree with combined majors in mechanical engineering and in computer engineering, in October 2011, from the Technical University of Berlin, Germany. In 2015 he received a Ph.D. degree from Northeastern University. He is currently at Siemens Corporate Research, Princeton, NJ.



**Yang Li** Yang Li received a Ph.D. degree in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute in 2015. He received a B.Eng. degree in Electrical Engineering from Hong Kong Polytechnic University in 2010. As a graduate student, he was affiliated with the DHS Center of Excellence on Explosives Detection, Mitigation and Response (ALERT). His research interests include pedestrian surveillance, object tracking and human re-identification in non-overlapping camera networks.

**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 Service 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.

**Ziyan Wu** Ziyan Wu received a Ph.D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute in 2014. He has B.S. and M.S. degrees in Engineering from Beihang University. He joined Siemens Corporate Research as a Research Scientist in 2014. His current research interests include 3D object recognition and autonomous perception.

**Fei Xiong** Fei Xiong received the B.S. degree in Automation and M.S. degree in Pattern Recognition from Huazhong University of Science and Technology, in 2004 and 2007, respectively, and Ph.D. degree in Electrical Engineering from Northeastern University, Boston, in 2014, respectively. He joined Amazon Inc. in 2014. His current research interests include visual tracking and object recognition, video segmentation, manifold learning, and 3D reconstruction.