Task Order 5 Final Report

Richard J. Radke, Rensselaer Polytechnic Institute
Octavia Camps, Northeastern University
Venkatesh Saligrama, Boston Univerity
Deanna Beirne, Northeastern University

September 24, 2015

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 was given 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.

1 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., [5, 13, 34]).

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 [2, 9, 29, 36], i.e., determining effective ways to extract representative information from each cropped rectangle to produce descriptors. The second is metric learning [3, 23, 26, 27, 38, 42], 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 [9], ETHZ [29], and i-LIDS MCTS [32] 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 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 report describes the unique opportunity our team had to design and deploy a real-world reidentification algorithm in an airport, in which we had to surmount the above challenges. We call this implementation of re-id "tag-and-track", since the system begins with the user tagging a person of interest in one camera, and attempts to track them throughout the broader camera network at the airport in real time. We begin by describing important practical considerations of the airport environment in Section 2. Section 3 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 [24], described in Section 4. 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 5. Sections 6 and 7 report experimental results from the on-site system currently deployed at our airport testbeds. Section 8 includes discussion and plans for future work. Section 9 briefly summarizes the conference and journal publications supported by Task Order 5.

2 Designing a "Tag and Track" System for CLE

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 the Cleveland Hopkins International Airport (CLE, Cleveland, Ohio, USA). The system was designed to assist Transportation Security Administration officers (TSOs) monitoring CLE 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 pedestrian 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.

Over the course of the project, teams of Rensselaer and Northeastern students visited CLE for six multi-day visits to plan, install, and test the camera placement, computer vision algorithms, and system architecture.

2.1 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 at the lab using these recorded datasets, and making trips to CLE to install and test the software on-site. Furthermore, it should be noted that 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.

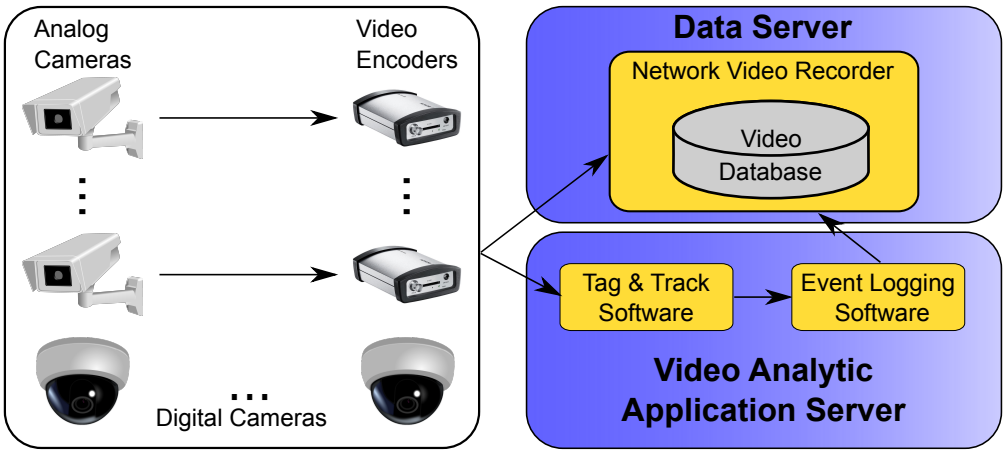

Figure 1: High-level system design of our airport human re-identification solution.

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 704×408 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 is transmitted through the secure net-

work to the analytic software modules and to an auxiliary research Network Video Recorder (NVR) 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 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.

2.2 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. 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 (~45°).

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 [37].

Finally, airports can be crowded, making the tasks of pedestrian 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 3.

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. This problem is complicated even further by the fact that the traffic flow of pedestrians 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.

2.3 CLE Tag-and-Track Testbeds

Over the course of two years, we built two testbeds in the Cleveland Hopkins International Airport (CLE) to design and evaluate the developed re-id system. The first testbed includes five cameras (which previously existed in the CLE network) leading from the garage to the terminals, and the second testbed (planned by ALERT researchers and installed for this express purpose) includes six cameras leading from the security checkpoint to different concourses. A partial floor plan of the Cleveland airport in shown in Figure 3, where the red dots represent garage testbed cameras and the yellow dots represent concourse testbed cameras.

Garage Testbed. There are five cameras serially covering the area from the parking garage through walkways to the terminals, shown in Figure 4. The resolution is 704×528 pixels, and the frame rate is 30fps. It can be seen that two cameras near the final escalator have very limited fields of view and highly distorted perspectives, so we did not use these two views in our testbed. The fields of view in the other three cameras are fairly well-connected, although people can leave the scene in the camera containing

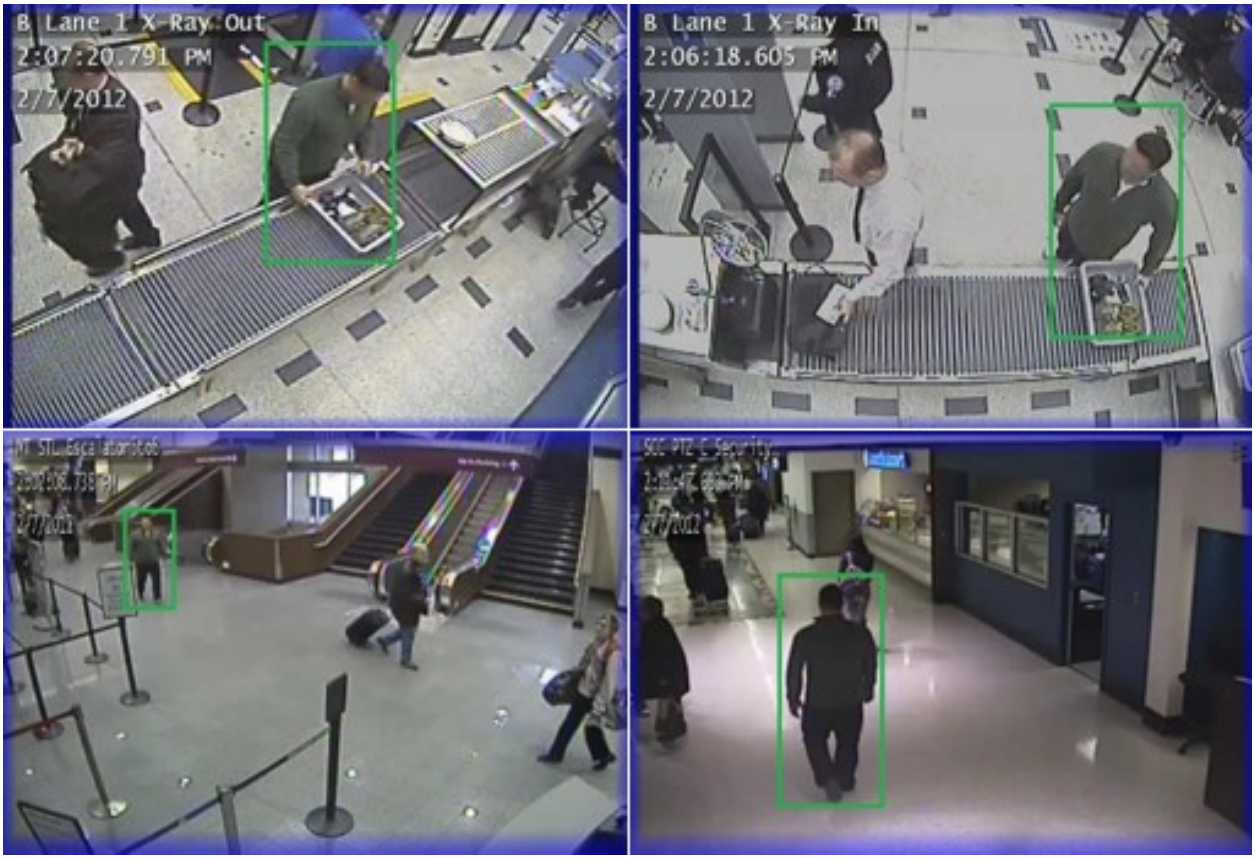

Figure 2: Sample images from airport camera videos.

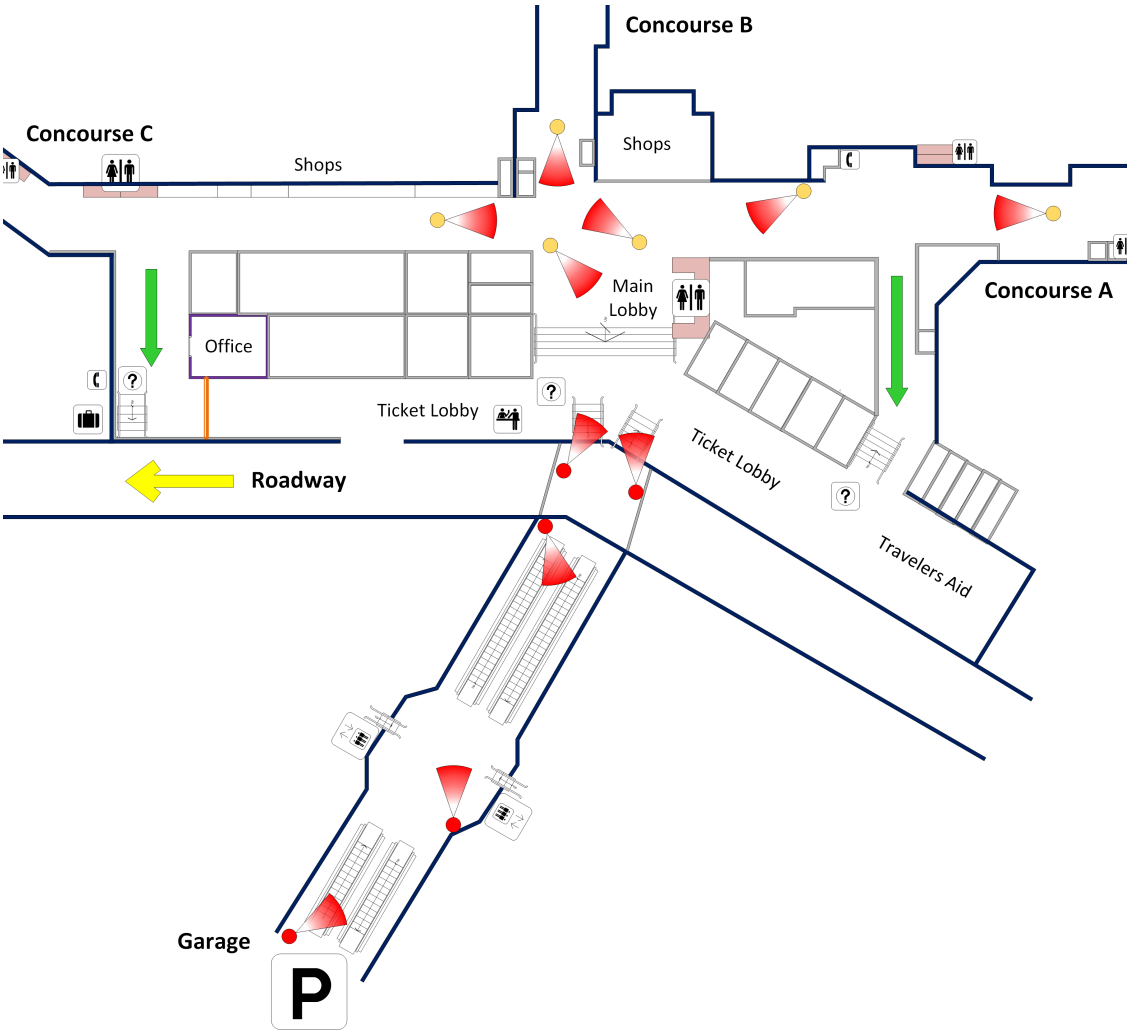

Figure 3: Partial floor plan of the CLE airport. Red dots are garage dataset cameras, yellow dots are concourse dataset cameras.

the elevator. It is also worth pointing out that the glass windows around the walkways severely affect the illumination conditions over the day, and the reflective floor is another difficulty.

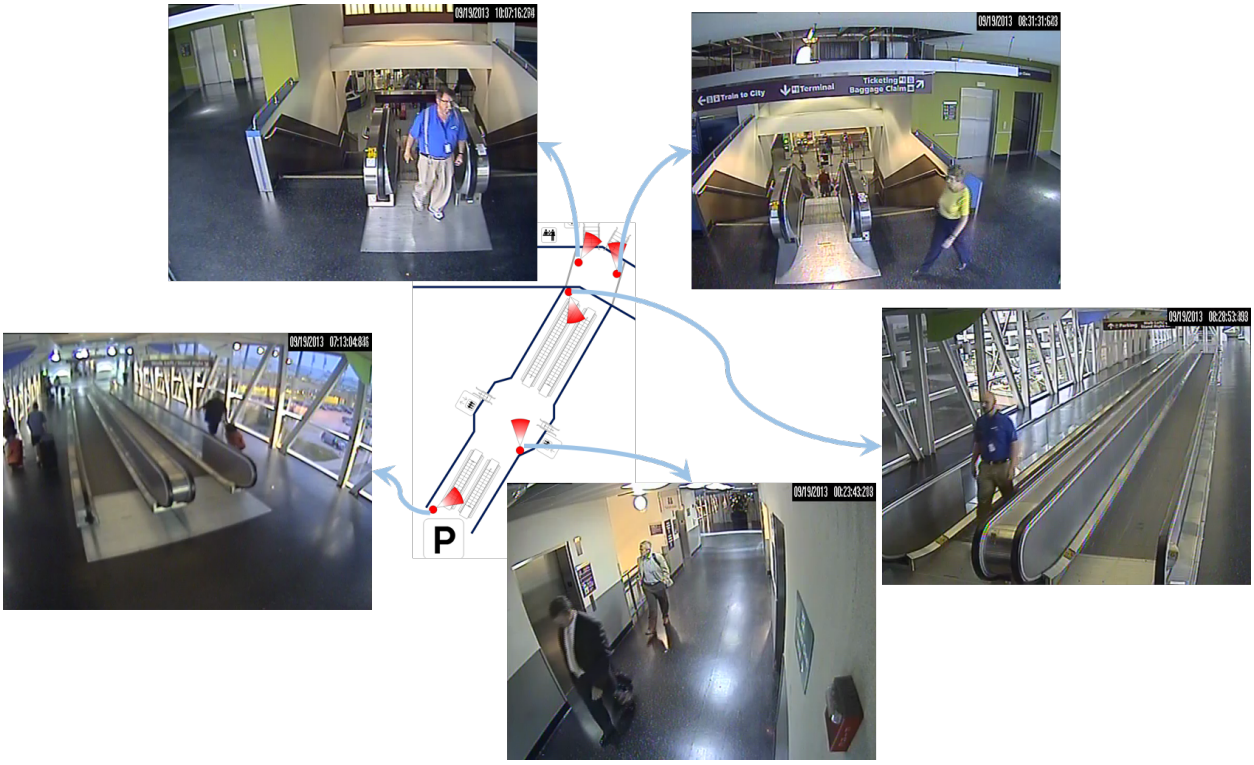

Figure 4: Sample images from five cameras in the garage testbed.

Concourse Testbed. There are six cameras whose locations were chosen by researchers to acquire better views for the purpose of re-id. Sample images are displayed in Figure 5. The cameras in this set have higher resolution and frame rate, 768×432 pixels and 60 fps respectively. However, the image quality problem still exists here, especially with the neon lights and reflective ground. The difficulty here is that people do not take straightforward routes as was the case in the garage dataset. It is hard to predict where people will reappear after the security checkpoint; they may walk around in the shopping area or have a meal in one of the restaurants. Moreover, people are likely to change clothes at the checkpoint, which means their appearance may change in those views.

3 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 6 illustrates the main steps of the process.

3.1 Detection and Tracking

The first step is using mixtures of Gaussians (MoG) [31] 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 pedestrian

detector of Dollár et al. [8], as illustrated in Figure 7. This detector uses a boosted decision tree classifier to rapidly generate pedestrian 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 pedestrian images from each camera (obtained using the ground-truthing tool in Section 5) and randomly sampled background images (to create negative samples). The pedestrian 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.

Figure 5: Sample images from six cameras in the concourse testbed.

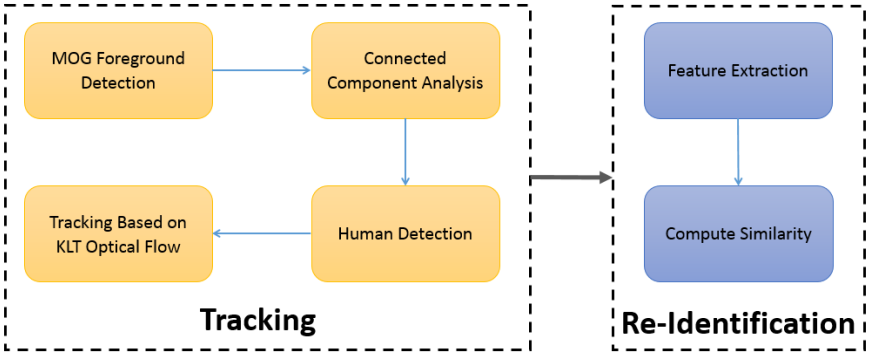

Figure 6: Block diagram outlining our human re-identification algorithm.

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 pedestrians from the previous frame. This prediction is made by detecting low-level FAST corner features [28] in each previous bounding box, removing features estimated to belong to the background [37], estimating 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 8 illustrates the idea.

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.

Figure 7: Pedestrian detection example using MoG foreground detection to reduce computational complexity.

3.2 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.

For feature extraction, we adopted the approach of Gray and Tao [9], which is particularly suitable for the low-resolution candidate rectangles generated in the airport system. The image is divided into

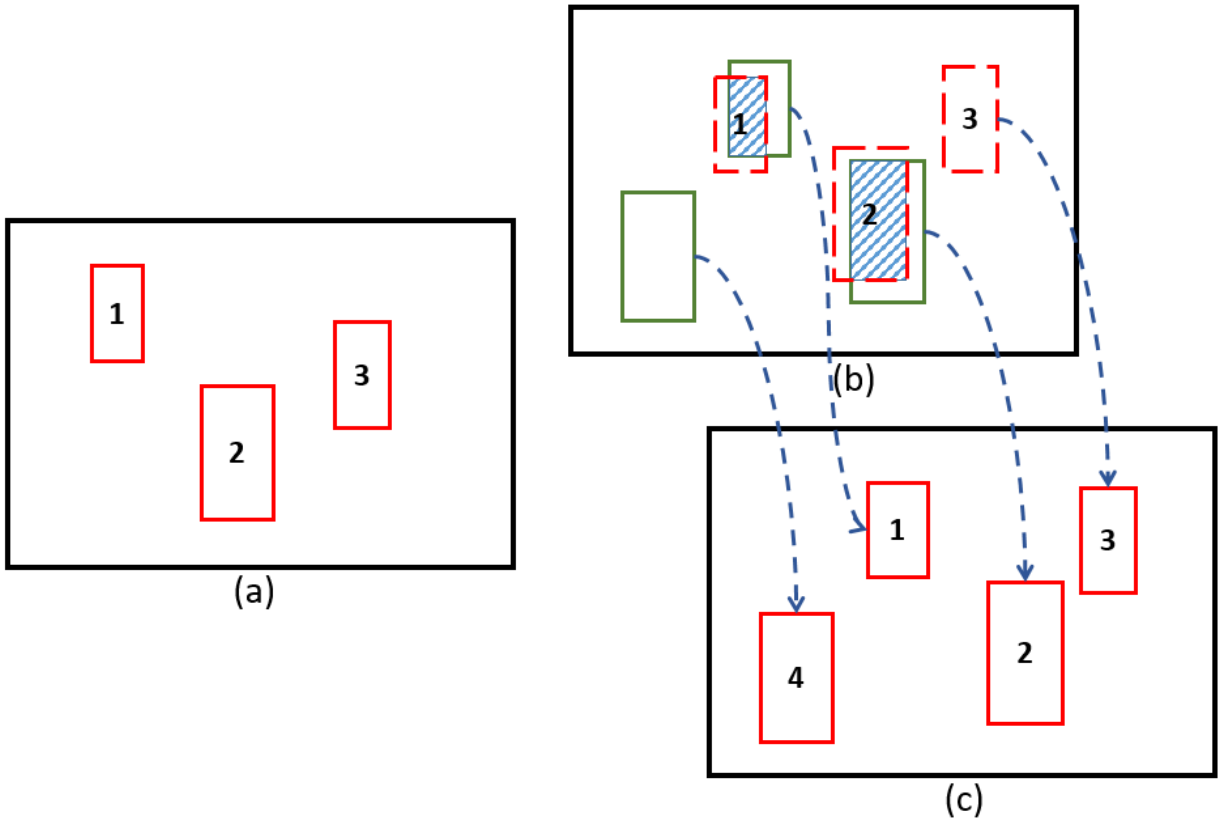

Figure 8: (a) Tracking-by-detection bounding boxes from previous frame.(b) Predicted bounding boxes using motion vector propagation from the previous frame (dashed, red) and new tracking-by-detection candidates (solid, green). (c) Final bounding boxes for current frame created by merging the two detections.

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 $\mathbf{x}_t^i \in \mathbb{R}^d$, $i = 1, \cdots, n$ and $\mathbf{x}_{c_j}^k \in \mathbb{R}^d$, $k = 1, \cdots, m$ 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 $\hat{\mathbf{x}}_{c_j}^k = \mathbf{P}\mathbf{x}_{c_j}^k$. We then determine $\mathbf{X}_{target} \in \mathbb{R}^{\hat{d}}$ 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,

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

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

Finally, the similarity score s_j is computed as

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

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. We next describe the procedure employed to learn the feature space projection matrix $\mathbf{P}$ and the weight vector $\mathbf{w}$.

To learn the projection matrix $\mathbf{P}$, we employ Local Fisher Discriminant Analysis (LFDA). The goal of the feature space projection is to find a discriminative space where samples from the same person are close, whereas the samples from different people are far apart. LFDA is particularly suitable to our problem because of data multi-modalities resulting from tracking each person. Formally, given the gallery feature vectors $\mathbf{g}_j^i$ and the probe feature vectors $\mathbf{p}_j^i$ 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 [10] are used to ensure 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

$$\mathbf{A}_{ab} = \begin{cases} 1 & \text{if } \mathbf{F}_{*a} \text{ close to } \mathbf{F}_{*b} \\ 0 & \text{otherwise} \end{cases}$$

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 [27]. The core idea is to minimize the norm of a vector $\mathbf{w}$ that satisfies the following ranking relationship:

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

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

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

where C is a margin trade-off parameter and ξ_i is a slack variable.

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

4 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 [24] to handle interprocess 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.

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 9). 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.

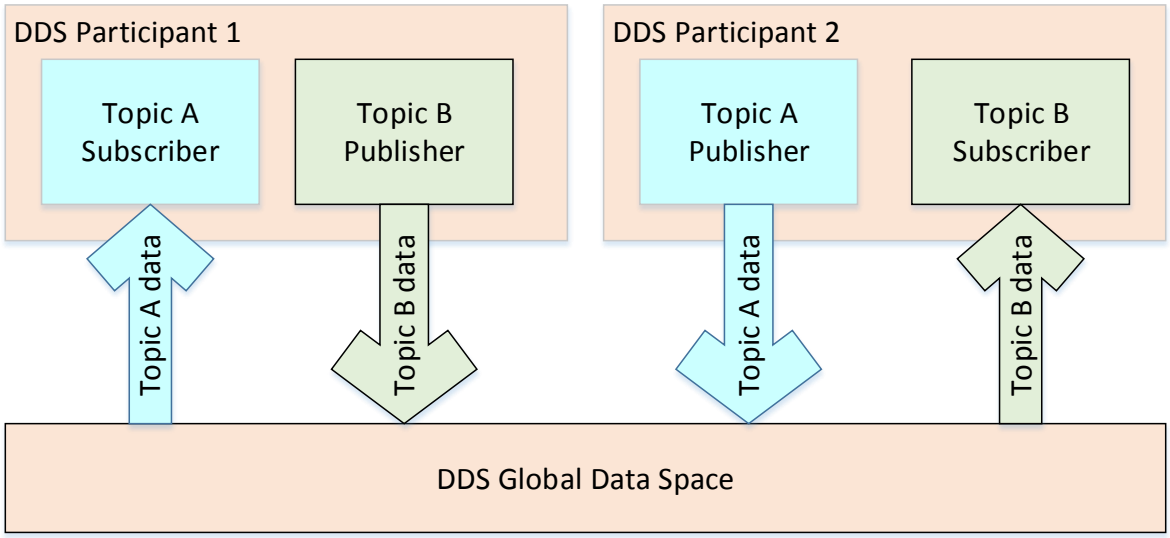

Figure 9: Block diagram showing participating entities in the publish-subscribe communication model used by DDS.

Figure 10 illustrates the DDS architecture corresponding to the re-identification software deployed on a three-camera system currently installed at the airport. Each block corresponds to a separate constantly running process performing the algorithms described in Section 3.

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 pedestrians detected in the video source.
   - *Subscribes to:* Video frames.
   - *Publishes:* Single frame candidate locations.
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 3.2. 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 3.2, 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.

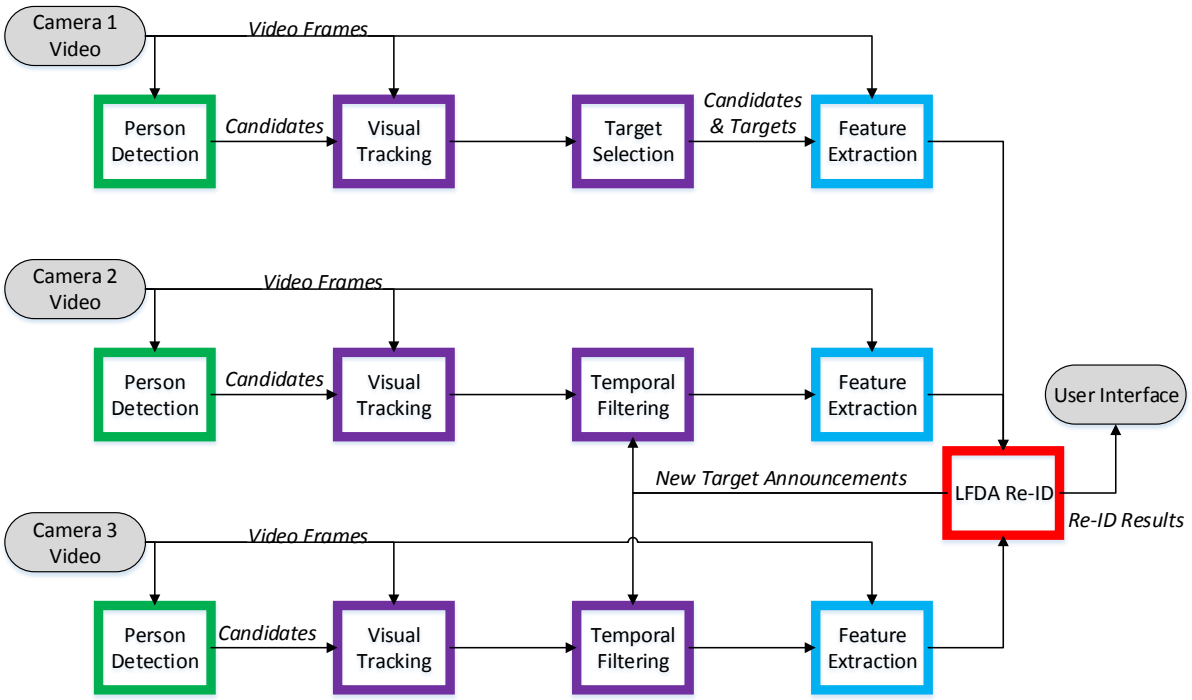

Figure 10: Block diagram showing the re-id system architecture, including processes for Candidate Detection (green), Candidate Filtering (purple), Feature Extraction (blue), and Re-Identification (red).

5 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 pedestrians 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.

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., crowdsourced marketplaces [30, 33]. 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 pedestrian 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 11, 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 [12] implemented on graphics processing units [11]. 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 [7]. 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 pedestrian to facilitate algorithm design and validation for re-identification, as discussed in Section 7.1.

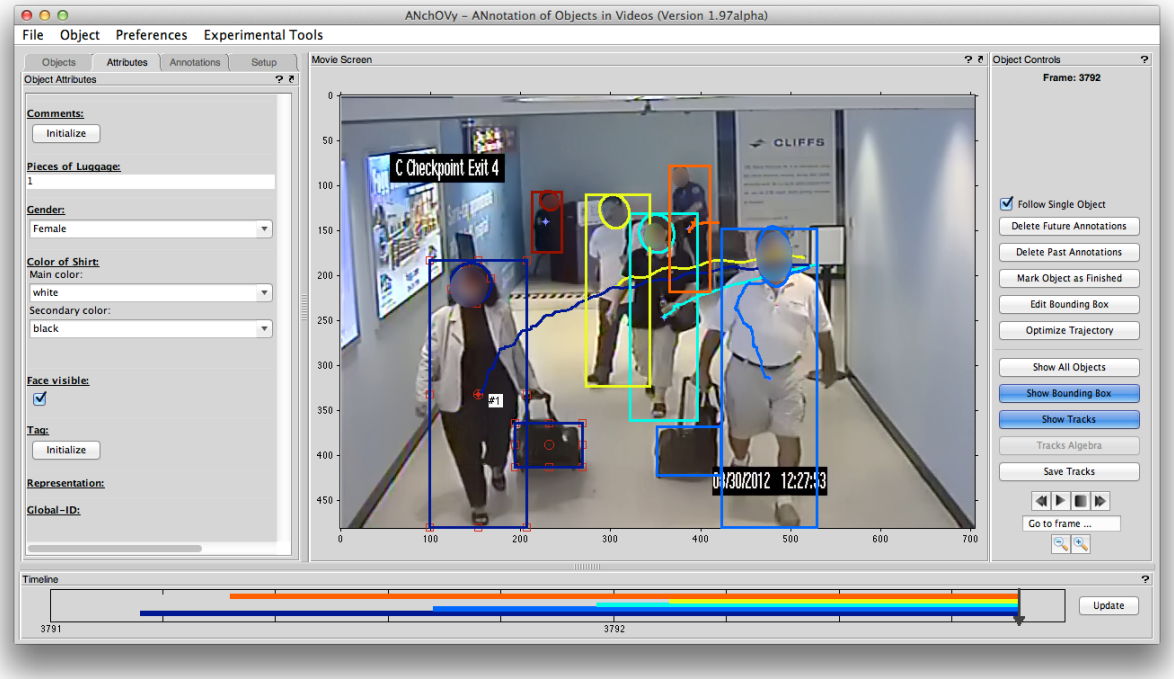

Figure 11: ANchOVy's graphical user interface showing pedestrians and their trajectories, spatial labels (full-body, head, and luggage bounding boxes) as well as other labels.

6 Experimental Results: Garage Testbed

For the garage testbed experiment, we used three cameras located in the area between the parking garage and the airport terminal. Sample images of the camera views are shown in Figures 12(a-c). People coming from the parking garage will be seen first in camera A. They then proceed to camera B (at which point there are stairs and elevators enabling them to enter or exit the scene). If they continue to move forward, they will appear in camera C and move into the terminal.

In each experiment, we tagged a person in camera A and then tracked him or her until they disappeared from the field of view. The target's feature vectors are extracted from the tracking frames. After the target leaves camera A, we begin to detect and track all the candidates in cameras B and C for a 5-minute period, as shown in Figures 12(b-c). We can see that the tracking task is very challenging in camera B, because of the distorted view angle and the crowded scene. However, as long as the person is successfully detected and tracked for a short distance, the program can still make a reliable judgment. One example re-id result is shown in Figure 12(d). We only display the top 5 candidates to the user in ascending order of similarity score. In this example, the target person is ranked second in camera B and third in camera C. Although detection and tracking in camera B is more challenging, the viewpoint (the person's back), is more similar to the tagged viewpoint, so the re-id results are better in camera B than in camera C.

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

Figure 12: Sample results from the garage testbed, (a) tagging the person of interest in camera A, (b) tracking in camera B, (c) tracking in camera C, (d) re-identification results (green boxes indicate correct candidates).

Across approximately 11 hours of run time, we automatically selected 42 targets in camera A from the output of the pedestrian detector. Each of these acts as a probe image from which feature vectors are extracted. After the target leaves camera A, we begin to detect and track candidates in cameras B and C during the following 5-minute period. One example re-id result is shown in Figure 12d. We display the top 4 candidates to the user, ranked in descending order of similarity score. In this example, the target person was ranked second in camera B and third in camera C.

Table 1 summarizes the results. The overall system found 88% of the targets at rank 10 in camera B and 38% of the targets at rank 10 in camera C. The relatively poor performance in camera C is caused by failures in the pedestrian detector. That is, frequently the true re-appearance of the target was not detected by the pedestrian detector module. As illustrated in Figure 13, the number of candidate matching pedestrians provided by the pedestrian detector in camera C is significantly lower than the number provided in camera B.

Therefore, we also computed the performance results after excluding those targets without any correct detections in the other cameras. In this case, the camera C performance is significantly improved (reaching 100% at rank 10 over this subset). Figure 14 shows cumulative match characteristic (CMC) curves for each of the two cameras and experimental subsets.

Table 1: Re-identification results for the on-site garage testbed experiment. "Default" indicates the performance of the overall algorithm (even when the correct candidate does not appear in the camera B or C gallery due to a failure of the pedestrian detector). "In-gallery" indicates the performance when we only include targets that have matching images in the camera B and C galleries.

	Re-id method	Rank 1	Rank 5	Rank 10	Rank 20
Camera B	Default	26.2	61.9	88.1	92.3
	In-gallery	28.3	66.7	94.9	100
Camera C	Default	14.3	26.2	38.1	38.1
	In-gallery	37.5	68.8	100	100

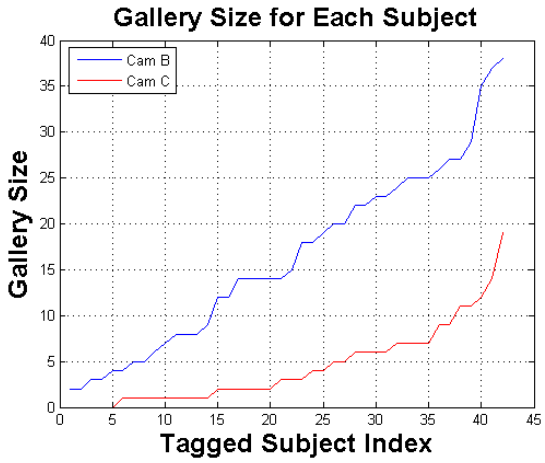

Figure 13: The number of candidates produced by the person detector for the re-id galleries in cameras B and C, as a function of the target index.

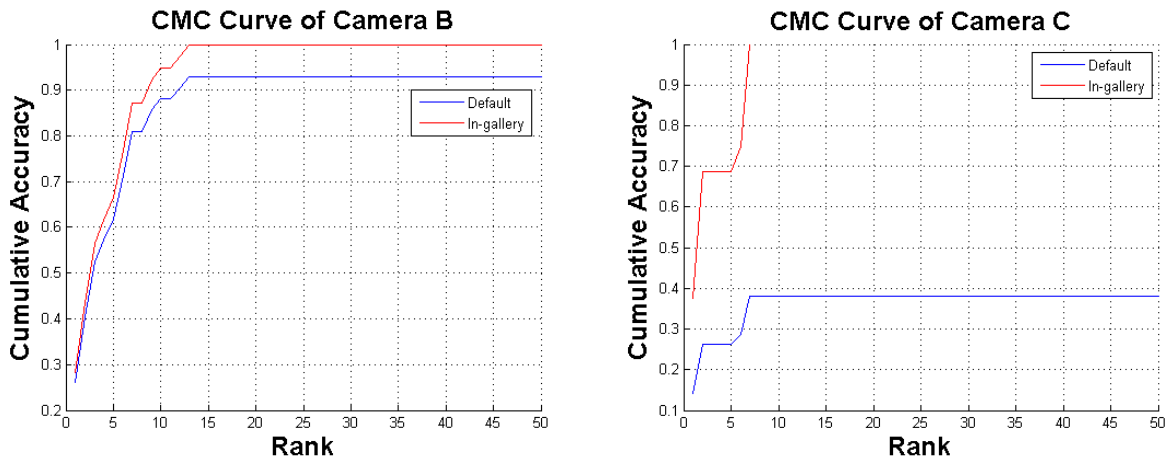

Figure 14: Cumulative match characteristic (CMC) curves corresponding to the experiments in Table 1.

7 Experimental Results: Concourse Testbed

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.

7.1 System Training

Using ANchOVy, we labeled 650 tracks of 188 pedestrians, 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 pedestrians in every twelfth video frame ranging in size from 51×30 to 267×212 pixels.

The cropped images were then used to train our pedestrian detection and re-identification algorithms. We grouped the pedestrian images based on their camera view to train camera-specific decision trees for human detection as described in Section 3.1. 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 3.2. 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).

7.2 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 15. Figure 15a shows tagging a person of interest in camera A, Figures 15b and c show detection and tracking of candidates in cameras B and C, and Figure 15d 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.

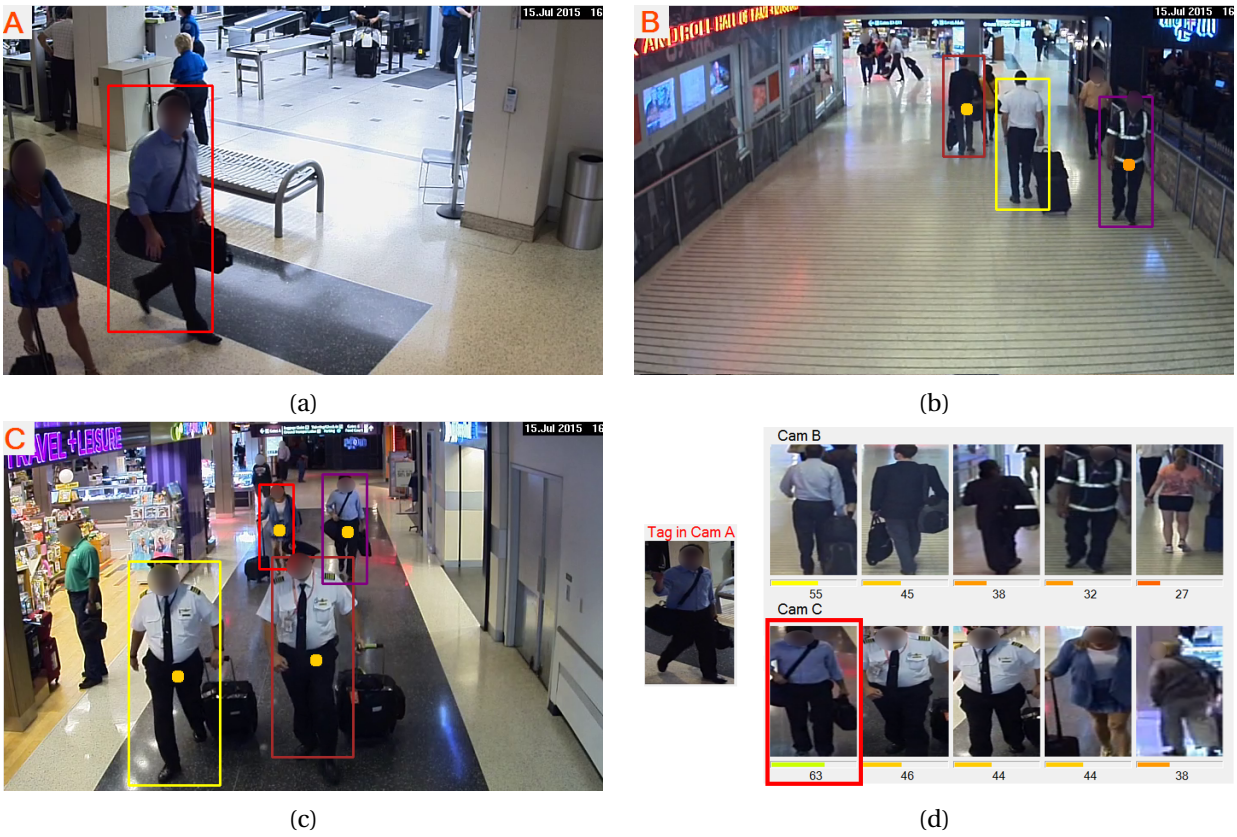

Figure 15: Snapshots from the graphical user interface developed for the airport human re-identification task. (a) Tagging the person of interest in camera A, (b) Tracking candidates in camera B, (c) Tracking candidates in camera C, (d) Re-identification results displayed to the user (red box indicates correctly re-identified candidate).

7.3 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 16. 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.

7.4 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, performance at low ranks (e.g., $n \leq 5$) is critical. To this end, we report the real-time performance of the system in terms of the rank-5 performance, i.e., the percentage of experiments in which the tagged person of interest was re-identified within the top-5 rank, and stayed within the top-5 rank throughout the 3-minute time window. The rank-5 performance in each of the two re-appearance cameras B and C is tabulated in Table 2. The cumulative match characteristic (CMC) curves for each of the two re-appearance cameras are shown in Figure 17.

We see that our system was successfully able to re-identify 58.5% of the targets that re-appeared in camera B and 61.4% of the targets that re-appeared in camera C. If we were to consider rank 10 performance, the statistics would be 83.0% and 87.7%, respectively.

We have also experimented off-line with various re-id algorithms on datasets derived from the concourse testbed. These are more ideal experiments in that system-level artifacts such as detection and tracking issues resulting from relatively low-frame rate video are absent (see Section 7.5). Based on these experiments, we project that improving these inputs to the system would improve re-id performance by as much as 15% at rank 5.

Figure 16: A sample of 20 concourse targets manually tagged for system performance evaluation.

Table 2: Re-identification results for on-site concourse testbed experiments.

Re-appearance camera	Number of experiments	Rank 5 performance	Rank 10 performance
B	94	58.5%	83.0%
C	57	61.4%	87.7%

7.5 Discussion

In this section, we discuss several issues that affected the overall performance of our system.

Video frame rate. We noticed serious compression artifacts from the video encoder while running the video at 30 frames per second. We had to reduce the frame rate to 10 frames per second to avoid this issue. Consequently, the performance of the detection and tracking modules suffered, as described next.

Detection and Tracking. 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.

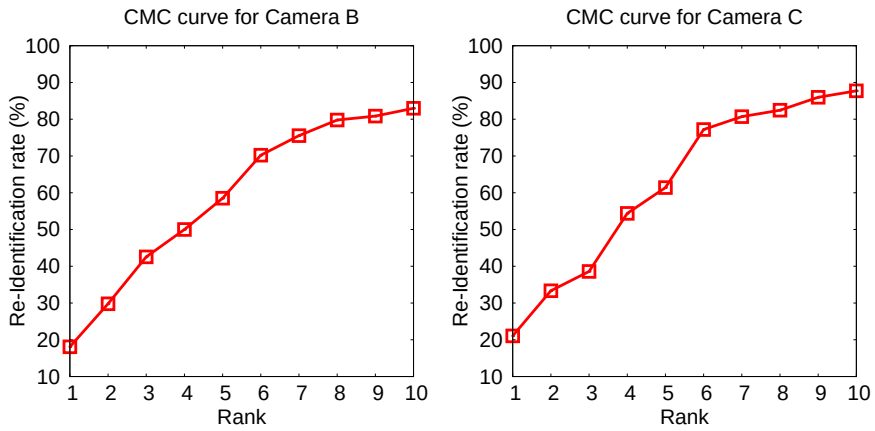

Figure 17: Cumulative match characteristic (CMC) curves corresponding to the experiments in Table 2.

7.6 Comparison with academic benchmarks

We conclude this section by comparing 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 [9], a standard benchmarking dataset. The current state of the art at rank-5 for VIPeR is around 75 – 80% [1, 25], 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:

Time-dependent gallery. In VIPeR, 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.

Hand-curated gallery vs. automatically-generated gallery. In VIPeR, 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 pedestrian detection and tracking algorithms that generate candidates in real time.

Gallery assumptions. In VIPeR, 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 pedestrian detection module may fail to detect the person even if they do appear.

8 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 results from our on-site algorithm deployment at the CLE airport.

To further improve the overall performance of our system, we could 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 [36], investigating personally-discriminative feature selection and comparison [19, 36], adaptively clustering feature vectors obtained from tracking prior to performing feature space projection [18], using kernel tricks to improve performance [38], accounting for appearance changes through design of co-occurrence kernels [40, 39], and

approaches based on structured prediction [41].

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 [11, 37].

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 5. 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.

9 Publications Supported by Task Order 5

In this section, we list and briefly summarize the conference and journal papers on human motion analysis and re-identification fully or partially supported by Task Order 5 funding. The full publications appear as an appendix to this report.

- Cheng et al. *A Convex Optimization Approach to Robust Fundamental Matrix Estimation* [6]. This paper considers the problem of estimating the fundamental matrix from corrupted point correspondences between images captured from different point of view. A general nonconvex framework is proposed that explicitly takes into account the rank-2 constraint on the fundamental matrix and the presence of noise and outliers. The main result of the paper shows that this nonconvex problem can be solved by solving a sequence of convex semi-definite programs, obtained by exploiting a combination of polynomial optimization tools and rank minimization techniques. Further, the algorithm can be easily extended to handle the case where only some of the correspondences are labeled, and, to exploit co-ocurrence information, if available. Consistent experiments show that the proposed method works well, even in scenarios characterized by a very high percentage of outliers.

- Karanam et al., *Particle Dynamics and Multi-Channel Feature Dictionaries for Robust Visual Tracking* [14]. This paper describes a novel approach to the visual tracking problem in a particle filter framework based on sparse visual representations. Current state-of-the-art trackers use low-resolution image intensity features in target appearance modeling. Such features often fail to capture sufficient visual information about the target. Here, we demonstrate the efficacy of visually richer representation schemes by employing multi-channel feature dictionaries as part of the appearance model. To further mitigate the tracking drift problem, we propose a novel dynamic adaptive state transition model, taking into account the dynamics of the past states. Finally, we demonstrate the computational tractability of using richer appearance modeling schemes by adaptively

pruning candidate particles during each sampling step, and using a fast augmented Lagrangian technique to solve the associated optimization problem. Extensive quantitative evaluations and robustness tests on several challenging video sequences demonstrate that our approach substantially outperforms the state of the art, and achieves stable results.

- Karanam et al., *Sparse Re-Id: Block Sparsity for Person Re-Identification* [16]. This paper presents a novel approach to solve the problem of person re-identification in non-overlapping camera views. We hypothesize that the feature vector of a probe image approximately lies in the linear span of the corresponding gallery feature vectors in a learned embedding space. We then formulate the re-identification problem as a block sparse recovery problem and solve the associated optimization problem using the alternating directions framework. We evaluate our approach on the publicly available PRID 2011 and iLIDS-VID multi-shot re-identification datasets and demonstrate superior performance in comparison with the current state of the art.

- Karanam et al., *Person Re-Identification with Discriminatively Trained Viewpoint Invariant Dictionaries* [15]. This paper introduces a new approach to address the person re-identification problem in cameras with non-overlapping fields of view. Unlike previous approaches that learn Mahalanobis-like distance metrics in some embedding space, we propose to learn a dictionary that is capable of discriminatively and sparsely encoding features representing different people. Our approach directly addresses two key challenges in person re-identification: viewpoint variations and discriminability. First, to tackle viewpoint and associated appearance changes, we learn a single dictionary to represent both gallery and probe images in the training phase. We then discriminatively train the dictionary by enforcing explicit constraints on the associated sparse representations of the feature vectors. In the testing phase, we re-identify a probe image by simply determining the gallery image that has the closest sparse representation to that of the probe image in the Euclidean sense. Extensive performance evaluations on two publicly available multi-shot re-identification datasets demonstrate the advantages of our algorithm over several state-of-the-art dictionary learning, temporal sequence matching, and spatial appearance and metric learning based techniques.

- Li et al., *Real-World Re-Identification in an Airport Camera Network* [17]. This paper is a preliminary version of this report (and of the journal paper [4] in review mentioned below). In this paper, we describe an end-to-end system solution of the re-identification problem installed in an airport environment, with a focus on the challenges brought by the real world scenario. We discuss the high-level system design of the video surveillance application, and the issues we encountered during our development and testing. We also describe the algorithm framework for our human re-identification software, and discuss considerations of speed and matching performance. Finally, we report the results of an experiment conducted to illustrate the output of the developed software as well as its feasibility for the airport surveillance task.

- Li et al., *Multi-Shot Re-Identification with Random-Projection-Based Random Forests* [19]. Current metric learning algorithms mainly focus on finding an optimized vector space such that observations of the same person in this space have a smaller distance than observations of two different people. In this paper, we propose a novel metric learning approach to the human reidentification problem, with an emphasis on the multi-shot scenario. First, we perform dimensionality reduction on image feature vectors through random projection. Next, a random forest is trained based on pairwise constraints in the projected subspace. This procedure repeats with a number of random projection bases, so that a series of random forests are trained in various feature subspaces. Finally, we select personalized random forests for each subject using their multi-shot appearances.

- Li et al., *Multi-Shot Human Re-Identification Using Adaptive Fisher Discriminant Analysis* [18]. While much research in human re-identification has focused on the single-shot case, in real-world applications we are likely to have an image sequence from both the person to be matched and each candidate in the gallery, extracted from automated video tracking. It is desirable to take advantage of the multiple visual aspects (states) of each subject observed during training and testing. However, since each subject may spend different amounts of time in each state, equally weighting all the images in a sequence is likely to produce suboptimal performance. To address this problem, we introduce an algorithm to hierarchically cluster image sequences and use the representative data samples to learn a feature subspace maximizing the Fisher criterion. The clustering and subspace learning processes are applied iteratively to obtain diversity-preserving discriminative features. A metric learning step is then applied to bridge the appearance difference between two cameras.

- Lo Presti et al., *Gesture Modeling by Hanklet-based Hidden Markov Model*, [20] and *Hankelet-based Dynamical Systems Modeling for 3D Action Recognition,* [21]. In these papers we propose to model an action as the output of a sequence of atomic Linear Time Invariant (LTI) systems. The sequence of LTI systems generating the action is modeled as a Markov chain, where a Hidden Markov Model (HMM) is used to model the transition from one atomic LTI system to another. In turn, the LTI systems are represented in terms of their Hankel matrices. For classification purposes, the parameters of a set of HMMs (one for each action class) are learned via a discriminative approach. This work proposes a novel method to learn the atomic LTI systems from training data, and analyzes in detail the action representation in terms of a sequence of Hankel matrices. Extensive evaluation of the proposed approach on two publicly available datasets demonstrates that the proposed method attains state-of-the-art accuracy in action classification from the 3D locations of body joints (skeleton).

- Wang et al., *Self Scaled Regularized Robust Regression* [35]. Linear Robust Regression (LRR) seeks to find the parameters of a linear mapping from noisy data corrupted from outliers, such that the number of inliers (i.e. pairs of points where the fitting error of the model is less than a given bound) is maximized. While this problem is known to be NP hard, several tractable relaxations have been recently proposed along with theoretical conditions guaranteeing exact recovery of the parameters of the model. However, these relaxations may perform poorly in cases where the fitting error for the outliers is large. In addition, these approaches cannot exploit available *a-priori* information, such as co-occurrences. To circumvent these difficulties, in this paper we present an alternative approach to robust regression. Our main result shows that this approach is equivalent to a "self-scaled" ℓ_1 regularized robust regression problem, where the cost function is automatically scaled, with scalings that depend on the a-priori information. Thus, the proposed approach achieves substantially better performance than traditional regularized approaches in cases where the outliers are far from the linear manifold spanned by the inliers, while at the same time exhibits the same theoretical recovery properties. These results are illustrated with several application examples using both synthetic and real data.

- Wu and Radke, *Improving Counterflow Detection in Dense Crowds with Scene Features* [37]. This paper addresses the problem of detecting counterflow motion in videos of highly dense crowds. We focus on improving the detection per formance by identifying scene features | that is, features on motionless background surfaces. We propose a three-way classifier to differentiate counterflow from normal flow, simultaneously identifying scene features based on statistics of low-level feature point tracks. By monitoring scene features, we can reduce the likelihood that moving features' point tracks mix with scene feature point tracks, as well as detect and discard frames with periodic jitter. We also construct a Scene Feature Heat Map, which reflects the space-varying prob-

ability that object trajectories might mix with scene features. When an object trajectory nears a high-probability region of this map, we switch to a more time-consuming and robust joint Lucas-Kanade tracking algorithm to improve performance.

- Wu et al., *Viewpoint Invariant Human Re-identification in Camera Networks Using Pose Priors and Subject-Discriminative Features* [36]. Current re-id algorithms are likely to fail in real-world scenarios for several reasons. For example, surveillance cameras are typically mounted high above the ground plane, causing serious perspective changes. Also, most algorithms approach matching across images using the same descriptors, regardless of camera viewpoint or human pose. Here, we introduce a re-identification algorithm that addresses both problems. We build a model for human appearance as a function of pose, using training data gathered from a calibrated camera. We then apply this "pose prior" in online re-identification to make matching and identification more robust to viewpoint. We further integrate person-specific features learned over the course of tracking to improve the algorithm's performance.

- Xiong et al., *Re-Id Using Kernel-based Metric Learning Methods* [38]. In this work we propose the use, and extensively evaluate the performance, of four alternative methods for Re-Identification classification: regularized Pairwise Constrained Component Analysis, kernel Local Fisher Discriminant Analysis, Marginal Fisher Analysis and a ranking ensemble voting scheme, used in conjunction with different sizes of sets of histogram-based features and linear, χ^2 and RBF-χ^2 kernels. Comparisons against the state of the art show significant improvements in performance.

- Zhang et al., *Group Membership Prediction* [40]. The group membership prediction (GMP) problem involves predicting whether or not a collection of instances share a certain semantic property. We propose a novel probability model and introduce latent view-specific and view-shared random variables to jointly account for the view-specific appearance and crossview similarities among data instances. Our model posits that data from each view is independent conditioned on the shared variables. This postulate leads to a parametric probability model that decomposes group membership likelihood into a tensor product of data-independent parameters and data-dependent factors. We propose learning the data-independent parameters in a discriminative way with bilinear classifiers, and test our prediction algorithm on challenging visual recognition tasks such as multi-camera person re-identification and kinship verification.

- Zhang et al., *A Novel Visual Word Co-occurrence Model for Person Re-identification* [39]. In this work we account for appearance transformation between camera views by means of a co-occurrence matrix of visual word joint distributions in probe and gallery images. Our appearance model naturally accounts for spatial similarities and variations caused by pose, illumination, and configuration change across camera views. Linear support vector machines are then trained as classifiers using these co-occurrence descriptors.

In addition, the following papers are currently under review:

- Camps et al., *From the Lab to the Real World: Re-Identification in an Airport Camera Network* [4]. This paper is basically the same as this report, and is meant to describe the overall joint effort between Rensselaer and Northeastern in bringing academic re-id research into the challenging airport setting.

- Zhang and Saligrama, *PRISM: Person Re-Identification via Structured Matching* [41]. PRISM views the Re-ID problem as a weighted graph matching problem, and estimates edge weights by learning

to predict them based on the co-occurrences of visual patterns in the training examples. These co-occurrence based scores in turn account for appearance changes by inferring likely and unlikely visual co-occurrences appearing in training instances. We implement PRISM on single shot and multi-shot scenarios. PRISM uniformly outperforms the state-of-the-art in terms of matching rate while being computationally efficient.

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 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. Special thanks to Srikrishna Karanam, Yang Li, and Ziyan Wu (Rensselaer) and Mengran Gou, Tom Hebble, Oliver Lehmann, and Fei Xiong (Northeastern) for developing and installing the on-site CLE re-id system and designing the underlying computer vision algorithms. Thanks to Ziming Zhang, Marc Eder, Yuting Chen, and Philip Tran (Boston University) for developing computer vision algorithms.

References

[1] 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*, pages 3908–3916, Boston, MA, 2015.

[2] L. Bazzani, M. Cristani, and V. Murino. Symmetry-driven accumulation of local features for human characterization and re-identification. *Comput. Vision and Image Understanding*, 117(2):130–144, 2013.

[3] 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.*, pages 1473–1480, Whistler, BC, Canada, 2005.

[4] O. Camps, M. Gou, T. Hebble, S. Karanam, O. Lehmann, Y. Li, R. Radke, Z. Wu, and F. Xiong. From the lab to the real world: Re-identification in an airport camera network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015. Under review.

[5] 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*, 13(4):625–638, 2011.

[6] Y. Cheng, J. Lopez, O. Camps, and M. Sznaier. A convex optimization approach to robust fundamental matrix estimation. In *CVPR*, pages 2170–2178, 2015.

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

[8] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1532–1545, 2014.

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

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

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

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

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

[14] S. Karanam, Y. Li, and R. J. Radke. Particle dynamics and multi-channel feature dictionaries for robust visual tracking. In *British Machine Vision Conference*, Swansea, UK, 2015.

[15] S. Karanam, Y. Li, and R. J. Radke. Person re-identification with discriminatively trained viewpoint invariant dictionaries. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.

[16] S. Karanam, Y. Li, and R. J. Radke. Sparse re-id: Block sparsity for person re-identification. In *IEEE/ISPRS 2nd Joint Workshop on Multi-Sensor Fusion for Dynamic Scene Understanding (MSF 15)*, 2015.

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

[18] Y. Li, Z. Wu, S. Karanam, and R. J. Radke. Multi-shot human re-identification using adaptive fisher discriminant analysis. In *British Machine Vision Conference*, Swansea, UK, 2015.

[19] Y. Li, Z. Wu, and R. Radke. Multi-shot re-identification with random-projection-based random forests. In *IEEE Winter Conference on Applications of Computer Vision*, pages 373–380, Kona, HI, 2015.

[20] L. Lo Presti, M. L. Cascia, S. Sclaroff, and O. Camps. Gesture modeling by hanklet-based hidden markov model. In *Asian Conf. on Computer Vision (ACCV)*, 2014.

[21] L. Lo Presti, M. L. Cascia, S. Sclaroff, and O. Camps. Hankelet-based dynamical systems modeling for 3d action recognition. *Image and Vision Computing*, to appear.

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

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

[24] Object Management Group. Data distribution service for real-time systems. `http://portals.omg.org/dds/`. Accessed: 2015-08-04.

[25] 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*, pages 1846–1855, Boston, MA, 2015.

[26] 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*, pages 3318–3325, Portland, OR, 2013.

[27] 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.

[28] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, 2010.

[29] 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.*, pages 322–329, Rio de Janeiro, Brazil, 2009.

[30] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. *Urbana*, 51(61):820, 2008.

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

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

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

[34] 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.*, 32(1):56–71, 2010.

[35] Y. Wang, C. Dicle, M. Sznaier, and O. Camps. Self scaled regularized robust regression. In *CVPR*, pages 3261–3269, 2015.

[36] Z. Wu, Y. Li, and R. Radke. Viewpoint invariant human re-identification in camera networks using pose priors and subject-discriminative features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):1095–1108, 2015.

[37] Z. Wu and R. J. Radke. Improving counterflow detection in dense crowds with scene features. *Pattern Recognition Letters*, 44:152–160, 2014.

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

[39] Z. Zhang, Y. Chen, and V. Saligrama. A novel visual word co-occurrence model for person re-identification. In *ECCV Workshop on Visual Surveillance and Re-Identification*, 2014.

[40] Z. Zhang, Y. Chen, and V. Saligrama. Group membership prediction. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.

[41] Z. Zhang and V. Saligrama. Prism: Person re-identification via structured matching. In *Under review*, 2015.

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