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

Octavia Camps[a], Mengran Gou[a], Tom Hebble[a], Srikrishna Karanam[b], Oliver Lehmann[a], Yang Li[b], Richard J. Radke[b,*], Ziyan Wu[b], Fei Xiong[a]

[a] *Department of Electrical and Computer Engineering, Northeastern University*
[b] *Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute*

Abstract

Human re-identification across non-overlapping fields of view is one of the fundamental problems in video surveillance. While most reported research for this problem is focused on improving the matching rate between pairs of cropped rectangles around humans, the situation is quite different when it comes to creating a re-identification algorithm that operates robustly in the real world. 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 and software architecture for our human re-identification software, and discuss the problem of obtaining ground truth. 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.

*Corresponding author

1. Introduction

Camera networks, often with little to no field of view overlap, are used to monitor large public spaces such as airport terminals, train stations and sports arenas. Thus, there has been a significant effort in the computer vision research community to address the problem of human tracking in video sequences captured by these types of networks (e.g., [4, 11, 24]).

One of the most challenging issues in multi-camera tracking is to correctly associate targets across the different viewpoints in the network. This problem is closely related to the re-identification (re-id) problem where appearance features are used to match images of people taken from different views. Researchers address the problem of re-id with emphases on feature selection [2, 7, 20, 25] and metric learning [3, 16, 18, 28, 17, 27]. Typically, re-id results are reported in the literature by evaluating and comparing the matching performance of the proposed algorithms on several standard benchmarking datasets agreed upon by the research community.

The story is very different when it comes to re-id in a real-world environment. In addition to addressing a well-defined research problem, i.e., deciding whether two bounding boxes representing humans correspond to the same person, there are many other challenges to building a reliable re-identification application for an actual surveillance system. With respect to hardware, one may need to consider camera installation locations constrained by security limitations of the site, low-quality images from legacy analog cameras equipped in the current network, data storage and transferring strategies, device synchronization, and network bandwidth. With respect to software, the system must operate in near real-time, deliver high-quality matching results with few false alarms, and have a software architecture that is robust to lags and crashes.

Another fundamental difference between real-world re-id and academic research on the problem is that most work in the latter case poses the problem as: given a probe image of a person, find the single image of the same person in a gallery of nicely cropped images taken from a different viewpoint. Then, the re-id performance is usually quantified with a curve illustrating the rank n matching rate, i.e., the percentage of probe images that matched correctly with one of the top n images in the gallery set. In the real-world case, instead of using manually cropped person images, candidates are automatically detected by a pedestrian detector algorithm running in real time. Furthermore, real-world users are unlikely to scroll through pages of candidates or wait for long periods of time for results, so performance at low ranks (e.g., $n \leq 5$) at a near real-time pace is critical.

In this paper, we present the system design of a video surveillance solution installed in a real-world airport environment, as well as an algorithm framework for human re-identification. Our goal is to help airport security officers to detect tagged people of interest in real time. The project involved numerous iterations of on-site tuning, testing, and evaluation, and we present the challenges we encountered during its development. We also describe our experiences in moving from academic computer vision algorithm development to "messy" real-world

implementation and deployment. This paper extends an earlier version of our work presented in Li et al. [12].

2. Real-World Challenges

Our video surveillance project is centered at a medium-sized airport (Cleveland Hopkins International Airport, Cleveland, Ohio, USA). The project goal is to develop an on-site, real-time video analytic system to assist the U.S. Transportation Security Administration (TSA) and airport security personnel to track specified people of interest throughout the airport's surveillance camera network. We called this task "Tag and Track". The front end requires a simple graphical user interface to allow TSA agents to "tag" the person of interest. The back end requires multiple processes running in real time to recognize, track, and compare candidates. For the efficiency and stability of the application, these modules must work in parallel and cooperate with each other smoothly. We also must deal with challenges imposed by the existing airport surveillance system. In this section, we address challenges and limitations we encountered in the real-world system design, installation, running and testing.

2.1. Data Collection, Storage and Transfer

The high-level system design is shown in Figure 1. Unlike traditional surveillance systems, in which camera videos are directly fed into monitoring screens watched by security staff, video data in an airport needs to be transmitted to workstations through a secure high-bandwidth network, and then processed by analytic software.

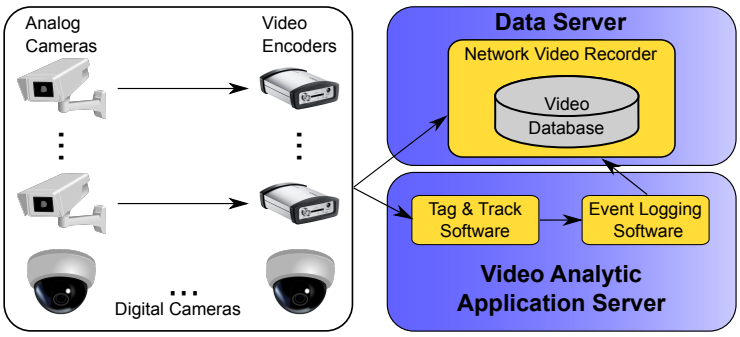

Figure 1: High-level system design of our airport human re-identification solution.

While most academic researchers likely use digital cameras in their labs, many legacy surveillance cameras in long-term installations, such as airports, are still analog. Much of the existing airport surveillance system at CLE is equipped with analog cameras, so it was necessary to install video encoders that convert the feeds into digital video data and embed video metadata. For this purpose, we use Bosch VIP X1 XF video encoders to convert the analog

video signal to the H.264 standard using a 704×408 resolution at 29.97 frames per second. All of the data is then transmitted not only to the analytic software but also to an auxiliary Network Video Recorder (NVR) application running on a data server, which stores the encoded video data for about one week.

The developed video analytic software acquires video feeds directly from the encoders and performs tracking and re-identification tasks in real time. All of the data transmissions are via a secure high-bandwidth network, and the whole system is maintained in a local Ethernet (i.e., no access to the outside Internet). Since the accessibility of the surveillance data from the airport is highly restricted, only the workstations connected in the local Ethernet are allowed to acquire the video data. Consequently, we had to conduct a large amount of software testing on-site, following a process of developing video analytic algorithms in the lab, testing them on small amounts of recorded data cleared by the airport authorities for our use, and deployed on-site after validation.

In addition, to facilitate systematic performance evaluation, every tenth frame is recorded at the processing computer. All the recorded data and events are reviewed by security officers, and brought back to the lab for analysis roughly every month.

2.2. Video Quality

We observed that several of the legacy analog cameras in the system contain serious noise and may not maintain focus over time. Figure 2 illustrates several sample images. It can also be observed that illumination conditions vary from camera to camera. Even for the same camera, the illumination can change throughout the day or with respect to weather conditions. The reflective decorative tile floor also makes foreground detection more difficult. Finally, the videos also contain periodic temporal jitter that seriously affects tracking algorithms. We discussed our solution to this problem in Wu et al. [26].

The heavy traffic environment in the airport makes detection and tracking even harder. In particular, trying to maintain accurate trajectories for each person in crowded scenes is especially challenging. We will discuss our tracking and re-id strategy in Section 3.

2.3. Camera Position

Like most public surveillance systems, the camera network at CLE does not cover the whole airport. In fact, the fields of view are mostly non-overlapping with large "blind" regions. To complicate matters further, the movements of humans in an airport are less predictable compared with other surveillance scenarios, such as vehicle traffic flow monitoring. For example, in most views there are no predefined routes or directions for people; after walking out of one view, people can walk back into the same view, while the algorithm may expect to detect the person in a different camera. There may be entrances and exits that are not covered by cameras, so that people can appear or disappear from the monitoring area with high uncertainty; people may stay for long periods or even change clothes while being out of the view of any camera, which can cause the estimation of their motion based on a fixed appearance or a transit-time model

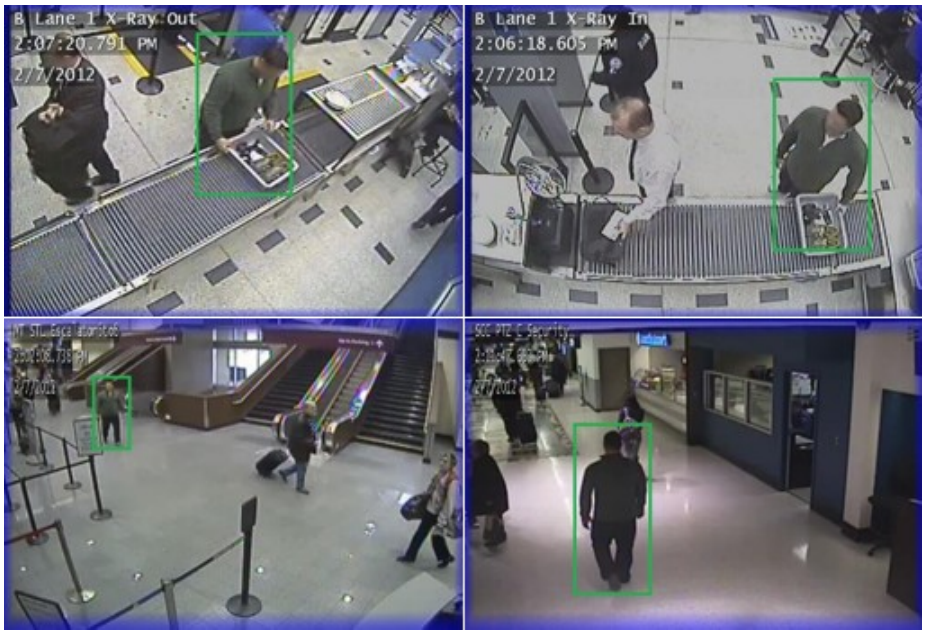

Figure 2: Sample images from airport camera videos.

to fail. Re-identification in this scenario is extremely challenging. Finally, unlike the standard datasets used to evaluate re-id algorithms, which contain images taken from cameras whose optical axes have a small angle with the ground plane, in the airport environments, the angle between the camera optical axis and the floor is usually large ($\sim 45°$), causing serious perspective effects (see Figure 2).

3. Algorithms Overview

In this section, we describe an overview of the algorithms used for the airport surveillance re-identification problem, emphasizing feasibility considerations for the real-world environment. The goal is to provide reliable re-id candidates corresponding to a tagged target person in real time. Figure 3 illustrates the major computer vision steps in the process.

3.1. Detection and Tracking

We begin with foreground detection using the mixture of Gaussians (MoG) method [22], followed by connected component analysis to group foreground pixels into blobs. The bounding box of each blob is considered as the region of interest (ROI). Each ROI is then fed into a real-time pedestrian detection algorithm as illustrated in Figure 4. For pedestrian detection, we adopted the aggregated channel features framework of Dollár et al. [6]. Specifically, a boosted decision tree classifier is used in conjunction with a sliding window approach.

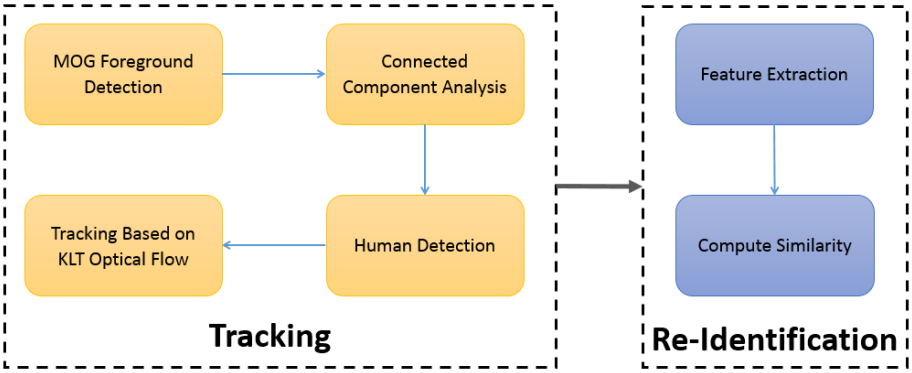

Figure 3: Block diagram outlining our human re-identification algorithm.

Figure 4: Pedestrian detection example using MoG-foreground detection to reduce computational complexity.

In our experiments, we found that training camera-specific classifier models resulted in better detection results than using a single model across all the camera views. To this end, in each camera view, we used 500 ground truth pedestrian images (forming the positive sample set) and randomly sampled background images (forming the negative sample set) to train camera-specific decision tree models. For each image, we constructed multi-scale feature pyramids by aggregating six quantized gradient orientations, L,U, and V color channels, and normalized gradient magnitudes into a ten-channel feature vector, computed over each scale. The result is a set of candidate detections of different sizes inside each ROI blob. Once the order is received to find a tagged person, human detection starts to run in all frames in each camera, since new humans may enter the scene at any time.

At the same time, a second set of candidate bounding boxes is obtained by propagating the location and bounding boxes of already tracked pedestrians from the previous to the current frame. For each bounding box, this update is performed by detecting low-level FAST corner features [19] in the previous frame cropped to the bounding box, removing any detected scene feature using

a background scene classifier [26], and tracking the remaining features with KLT trackers [15]. Averaging the resulting motion vectors yields the displacement update of the bounding box's position.

Finally, the tracking result is obtained by merging the two sets of candidate bounding boxes as follows. We compute the intersection of each new candidate detection with the propagated bounding boxes and find the maximum ratio between their area of intersection and the area of the smaller bounding box. The new candidate detection is associated with the corresponding propagated bounding 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. Propagated bounding boxes not matching any candidate detection are retained if both their aspect ratio and location in the frame are plausible. Figure 5 illustrates the idea.

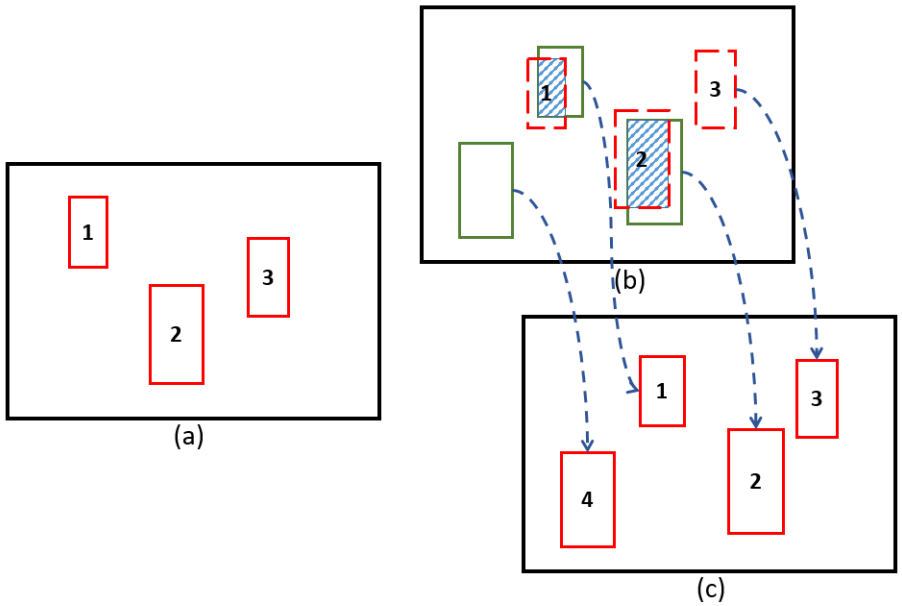

Figure 5: (a) Bounding boxes from previous frame.(b) Bounding boxes propagated from previous frame using feature detection and KLT tracker (dashed, red); new candidates generated by the human detector (solid, green). (c) Final bounding boxes for current frame created by merging the two detections.

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 jth candidate detection), we must compute an appropriate similarity score

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

to compare them. Lastly, 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 can be generated.

Feature detection for re-id in real-world scenarios is challenging, especially given the relatively small and low-quality target and candidate images and the need for real-time performance. Common descriptors, such as SIFT [14] and SURF [1], are unsuitable for this task due to their computational complexity. Instead, we found low-level features, such as color and texture histograms, to be more effective and efficient. In particular, we adopted the method described by Gray and Tao [7]. The image 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). By concatenating all the histograms we get a 2592-dimensional feature vector for each candidate. We found it was important to rectify the candidate sub-images based on simple camera calibration information to remove perspective distortion prior to feature extraction. In the future, we also plan to incorporate radiometric and color calibration across the cameras.

Many re-id algorithms in the literature define the similarity score s_j as a Mahalanobis-like distance

$$s_j = (\mathbf{X}_{\text{target}} - \mathbf{X}_j)^\top \mathbf{M} (\mathbf{X}_{\text{target}} - \mathbf{X}_j) \tag{2}$$

with $\mathbf{M} = \mathbf{P}^\top \mathbf{P}$, where $\mathbf{P}$ is a projection matrix learned from labeled data [16, 17, 27, 28]. Based on our experiments in [27] and taking into account computational considerations, in this project we decided to implement the LFDA algorithm proposed by Pedagadi et al. [17] to learn $\mathbf{P}$ using the following Fisher discriminative objective function:

$$\mathbf{P} = \max_{\mathbf{P}} (\mathbf{P}^\top \mathbf{S}^b \mathbf{P})^{-1} \mathbf{P}^\top \mathbf{S}^w \mathbf{P} \tag{3}$$

where $\mathbf{S}^w$ and $\mathbf{S}^b$ are the within and between scatter matrices, respectively. To preserve the local similarity, these matrices are defined as:

$$\mathbf{S}^w = \frac{1}{2} \sum_{i,j}^{n} \mathbf{A}^w_{i,j} (\mathbf{X}_i - \mathbf{X}_j)(\mathbf{X}_i - \mathbf{X}_j)^\top \tag{4}$$

$$\mathbf{S}^b = \frac{1}{2} \sum_{i,j}^{n} \mathbf{A}^b_{i,j} (\mathbf{X}_i - \mathbf{X}_j)(\mathbf{X}_i - \mathbf{X}_j)^\top \tag{5}$$

where $\mathbf{A}^*_{i,j}$ is an affinity matrix and n is the number of samples. The effect of applying the above projection matrix to the Euclidean distance between the target and candidate descriptors is that distances between descriptors of the same person will be smaller than distances between descriptors of different persons.

3.3. Algorithm Discussion

When developing the system for deployment at an airport, we had to consider requirements for both speed and performance. The algorithms need to be fast

enough to process multiple cameras in real time, and at the same time, find the person of interest with high confidence.

From a practical point of view, we found that it is important to consider the "big picture" of how good the results of each sub-process need to be in order to result in a confident re-id judgment, instead of trying to squeeze the best performance out of every algorithm at the possible cost of speed. For example, the MoG foreground detection is likely to fail when the surrounding illumination changes, but this can be mitigated later in the pipeline by the human detection step. In fact, a relatively sensitive foreground detection algorithm is preferable in order to ensure that we will not miss anybody in the detection stage (resulting in many false alarms that are rejected later). Similarly, there is no state-of-the-art tracking algorithm that can process multiple streams of airport-quality videos with high accuracy in real-time. The tracking algorithm we applied may fail when a candidate person is occluded, several trackers may become focused on the same person, or the bounding box may drift onto a different person. However, what it is important is to generate a sufficient number of reliable candidates for the re-id algorithm; occluded or poor-quality rectangles will simply never rise to the top of the rank-ordered candidate list.

In terms of computational cost, human detection is the most time-consuming step in the system; our implementation is close to real-time (around 15 fps on our videos). However, by filtering out ROIs with small sizes or impossible locations, and only analyzing viable ones, we highly reduced the computational cost to around 100 fps. While the process of training the re-id projection matrix is time-consuming, this is done offline. The on-line re-id process is extremely fast since it only involves a vector inner product. Thus, there is enough spare computational power in our system to consider in the future online re-id learning algorithms, such as updating representative feature vectors after the same person is confirmed in another view, discriminative model training, or the use of kernel tricks [27] to improve performance.

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, 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 [8] 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 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 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 provided 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 on a three-camera system at the airport currently installed.

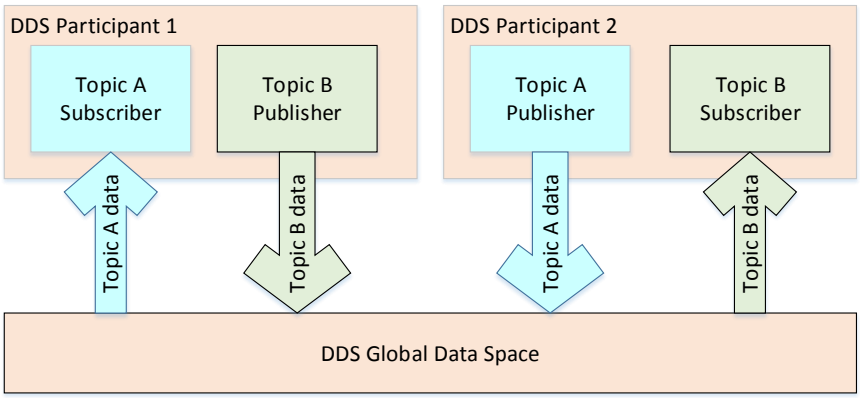

Figure 6: Block diagram showing participating entities in the publish-subscribe communication model used by DDS.

Each block corresponds to a seperate constantly running process performing the algorithms described in Section 3. In particular, the processing pipeline

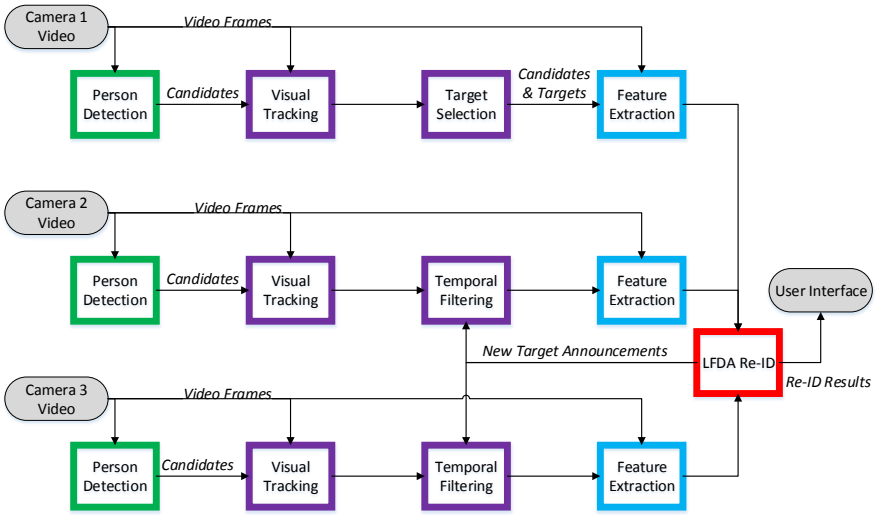

Figure 7: Block diagram showing the re-id system architecture, including processes for Candidate Detection (green), Candidate Filtering (purple), Feature Extraction (blue), and Re-Identification (red).

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.

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 hypotheses 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 [21, 23]. 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 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).

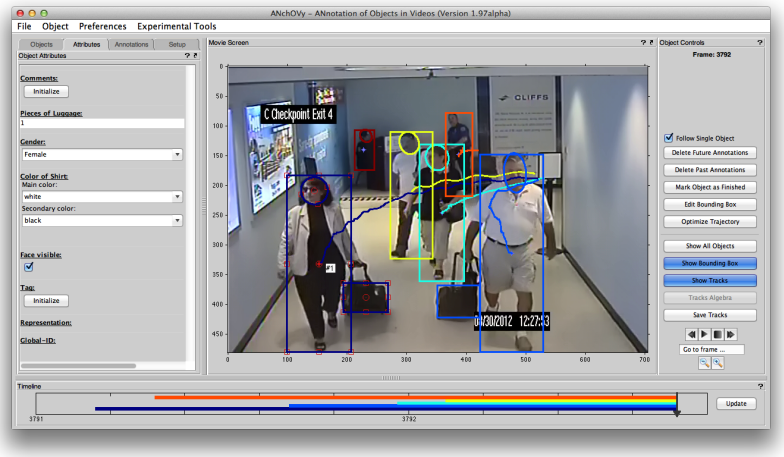

Figure 8: ANchOVy's graphical user interface showing pedestrians and their trajectories, spatial labels (full-body, head, and luggage bounding boxes) as well as other labels.

ANchOVy first automatically extracts short trajectories of moving objects visible in the video by using a featureless tracking-by-detection method [10] implemented on graphics processing units [9]. 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 [5]. 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 6.1.

6. Experimental Results

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

Figure 9: Sample results from the airport human re-identification system. (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).

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 its overall re-id performance. For these experiments, we chose to use video from three cameras located in the area between the airport terminal and the parking garage. Sample images of the camera views are shown in Figures 9a-c. People coming from the terminal 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 environment). If they continue to move forward, they will eventually appear in camera C and move into the parking garage.

6.1. 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 projection matrix $\mathbf{P}$ for the LFDA re-identification algorithm, as descibed in Section 3.2.

6.2. Performance Evaluation

To evaluate the performance of the system, we deployed it at Cleveland Hopkins International airport and ran experiments using live video feeds, recording the real-time re-identification results. 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 camera B and C during the following 5-minute period. One example re-id result is shown in Figure 9d. 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 10, the number of candidate matching pedestrains 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 11 shows cumulative match characteristic (CMC) curves for each of the two cameras and experimental subsets.

7. Conclusions

We discussed several practical challenges in implementing a real-time re-identification solution in a mid-sized airport, which might not be typically considered by academic researchers, and presented initial results from our algorithm framework tailored to this setting. However, there is still much work to be done, both in our specific environment, and more generally to make academic re-id research more closely match real-world scenarios.

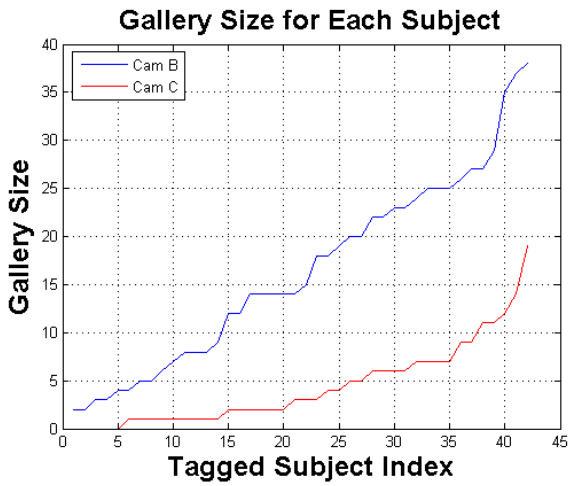

Figure 10: The number of candidates produced by the person detector for the re-id galleries in camera B and C, as a function of the target index.

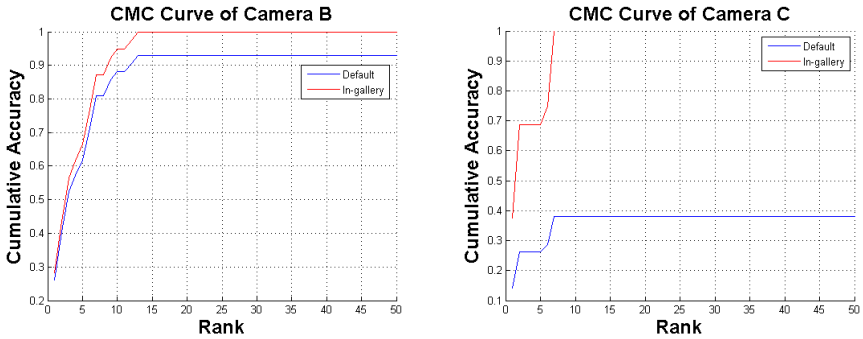

Figure 11: Cumulative match characteristic (CMC) curves corresponding to the experiments in Table 1.

	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

Table 1: Re-identification results for the on-site airport 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.

With respect to our specific environment, we are only at the beginning of our implementation and testing of the on-site re-id system, following a successful deployment of a system for real-time detection of counterflow through exit lanes described elsewhere [9, 26]. As discussed in Section 6, the performance of our deployed re-id algorithm is limited by the quality of the detected human candidates. To address this issue, we plan to use image edge information [29] to generate better pedestrian proposals prior to applying the detection algorithm.

We also note that most current human re-id algorithms are focused on the "single-shot" problem; that is, it is assumed that each person only has one image available to compute the similarity score. This assumption is mainly motivated by the limited data available in public re-id benchmark datasets. However, in real-world scenarios like the one considered here, there is a sequence of images available for each tracked person, leading to a "multi-shot" case. These images can be used to build better descriptors and generate more reliable similarity measurements. Multi-shot information could be used to train a discriminative model of the target person on-line, improving re-id performance [13]. Our next step of development is to incorporate multi-shot algorithms in our airport deployment to leverage all the available information about candidates and improve performance.

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 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.

We also had the unique opportunity to design a new re-id testbed at the airport, containing higher-quality digital cameras, positioned as carefully as possible within security and power constraints to capture a complex branching re-id scenario (i.e., a passenger exiting the security checkpoint can enter one of three concourses, after spending an unknown time in a shopping area). We

expect this new testbed to generate further challenges from both the research and practical perspectives.

8. 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 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] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[2] L. Bazzani, M. Cristani, and V. Murino. Symmetry-driven accumulation of local features for human characterization and re-identification. *CVIU*, 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 *NIPS*, 2005.

[4] 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.

[5] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2304–2311. IEEE, 2013.

[6] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *PAMI*, 2014.

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

[8] O. M. Group. Data distribution service for real-time systems, 2004.

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

[10] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[11] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *CVPR*, 2005.

[12] Y. Li, Z. Wu, S. Karanam, and R. Radke. Real-world re-identification in an airport camera network. In *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2014.

[13] Y. Li, Z. Wu, and R. Radke. Multi-shot re-identification with random-projection-based random forests. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

[16] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[17] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3318–3325. IEEE, 2013.

[18] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary. Person re-identification by support vector ranking. In *BMVC*, 2010.

[19] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 32(1):105–119, 2010.

[20] W. R. Schwartz and L. S. Davis. Learning discriminative appearance-based models using partial least squares. In *SIBGRAPI*, 2009.

[21] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. *Urbana*, 51(61):820, 2008.

[22] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[23] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. *Computer Vision–ECCV 2010*, pages 610–623, 2010.

[24] X. Wang, K. Tieu, and W. E. L. Grimson. Correspondence-free activity analysis and scene modeling in multiple camera views. *PAMI*, 32(1):56–71, 2010.

[25] 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*, 2014.

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

[27] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *Computer Vision–ECCV 2014*, pages 1–16. Springer, 2014.

[28] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011.

[29] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. 2014.