

Real-Time Airport Security Checkpoint Surveillance Using a Camera Network

Ziyan Wu and Richard J. Radke

Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, New York 12180

wuz5@rpi.edu, rjradke@ecse.rpi.edu

Abstract

We introduce an airport security checkpoint surveillance system using a camera network. The system tracks the movement of each passenger and carry-on bag, continuously maintains the association between bags and passengers, and verifies that passengers leave the checkpoint with the correct bags. We present methods for calibrating the camera network and tracking the many moving objects in the environment. We define a state machine for bag tracking and association, dividing the imaged area into several semantically meaningful regions. The real-time algorithms are validated on a full-scale simulation of a security checkpoint with several runs of volunteer groups, demonstrating high performance in a challenging environment.

1. Introduction

Airport security is a critical issue for defense and homeland security applications. Every day, millions of passengers pass through airport security screening checkpoints, which include X-ray machines to scan carry-on bags, and metal detectors or other devices to scan passengers. However, to our knowledge there are no automatic methods in place for tracking passengers and bags as they pass through such checkpoints, or for automatically maintaining the association of each item on the conveyor belt to the correct passenger. In this paper, we present a robust, real-time computer vision system to solve these problems, using data from a wide-area camera network containing both fixed and pan-tilt-zoom (PTZ) cameras.

We built a full-scale simulation of an airport security screening checkpoint, including a working conveyor belt and an artificial X-ray machine and metal detector. Several large groups of volunteers with real baggage were run through the simulation, and instructed to behave exactly as they would at an airport. The resulting multicamera video contains examples of both normal and abnormal activities characteristic of the real-world environment. The computer vision problems are complex due to the crowded environ-

ment, poor lighting/image contrast, and complex activities. For example, bags and bins are routinely reordered to simulate forgotten laptops/liquids, and all bags and passengers are at least momentarily hidden from view as they pass through the X-ray machine/metal detector. We demonstrate the results of our calibration, tracking, and association algorithms on this real multicamera video, showing that despite the challenging data we can achieve high performance¹.

2. Related work

Computer vision is widely used in environment monitoring. Many such systems are based on traffic monitoring and pedestrian tracking [2, 6, 8, 10]. Much research is focused on background/foreground modeling and understanding the basic behavior of people under surveillance. However, less research discusses the problem of tracking and associating large numbers of people and belongings.

Bhargava et al. [1] proposed a method to detect abandoned objects in a crowded scene. When an unaccompanied bag is detected, the system determines the most likely owner by searching the video history for the person who brought the bag into the scene. Cross-correlation is used to match the bag templates. Though the algorithm works well, it is somewhat time-consuming and limited to the example of isolated abandoned bags. Lv et al. [7] presented another algorithm for the same problem. They use features extracted from tracking results, including speed, direction and distance between objects, as evidence to decide on the owner of an abandoned bag based on a Bayesian inference framework. While the error rate in detection is low, it is difficult to assess the efficiency and flexibility of the algorithm since the experiment was based on simple events. Krahnstoever et al. [5] proposed a multi-camera system to solve the abandoned bag association problem. Object detection

¹Thanks to Eric Ameres, Keri Eustis, Andrew Calcutt, and RPI EMPAC for their assistance with the simulation. This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2008-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.

is done in each individual camera, and the information is combined using calibrated extrinsic parameters. The bag is associated with the owner whose track has the shortest ground plane distance to it. We note that all these systems are oriented towards associating isolated, stationary bags to owners, as opposed to maintaining associations in crowded scenes where all the objects are in motion.

3. Simulation Environment

Our simulation was constructed in a $50 \times 40 \times 30$ ft³ black-box studio. Figure 1 shows an overhead view of the studio and simulation.



Figure 1. Overhead view of the checkpoint simulation environment.

We built a full-scale simulated metal detector and X-ray machine, and constructed an operational conveyer belt and roller assembly to accurately mimic the configuration of a real carry-on baggage screening checkpoint. The conveyor belt includes a control panel by which the staff can stop or start the belt during baggage inspection.

The environment was instrumented with a camera network consisting of 19 cameras. 13 fixed cameras and a dome PTZ camera were mounted at the interstices of the room’s catwalk ceiling pointing downward to cover the entire floor. 5 PTZ cameras are mounted on the walls, panning regularly across the passenger lines and conveyor belt. Each camera delivers images at about 10Hz, which is sufficient for real-time tracking. The cameras are connected and fed by power-over-ethernet cables leading back to a power-over-ethernet switch. An application server manages the network, controls the cameras, and records the time-stamped video. The 6 PTZ cameras have resolution 704×480 and the 13 fixed cameras have resolution 640×480 .

Several groups of volunteers participated in the simulation as passengers. They were asked to go through the checkpoint with real baggage as they would at a real air-

port checkpoint (e.g., remove shoes and coats, remove laptops and liquids from bags and place them in regulation-sized bins, show agents valid ID and boarding pass at various points). Each round of simulation was monitored and recorded by the camera network. In addition, some passengers were randomly given instructions to act out certain roles, such as forgetting to remove liquids from their bags or mistakenly picking up another’s bag. More than 10 hours and 300GB of compressed multi-camera video data were collected during the month the simulation was active.

4. System Calibration

In order to relate the cameras’ images to each other, all cameras in the network must be jointly calibrated. The intrinsic parameters of each camera were calibrated prior to mounting using several positions of a planar 30×34 cm checkerboard target [11]. The intrinsic parameters of the PTZ cameras were calibrated at several different zoom levels, since the parameters vary during zooming. The RMS re-projection error from the calibration procedure was under 0.2 pixels for the fixed cameras and 0.25 pixels for the PTZ cameras.

Thus, after mounting, the extrinsic parameters of the cameras can be extracted from their essential matrices, which can be obtained with the eight-point algorithm [3]. However, since our environment is large and the fields of view of the ceiling cameras are limited, there is little overlap between the downward-pointing cameras. Instead, we used a novel approach for calibration, first calibrating the rotating PTZ cameras, and then using them as reference to calibrate the fixed cameras.

We calibrated the PTZ cameras simultaneously, using a scale bar of known length (1.18m) with two active lighting balls on its ends. First, we adjusted the pan and tilt parameters of the 6 PTZ cameras so that their fields of view overlapped. Then, we randomly placed the scale bar at 15-20 different non-coplanar positions, imaging the bar with both the room lights on and off. We extracted the centers of the actively lit balls in each on/off pair and used them as correspondences to estimate the essential matrix between each camera to be calibrated and the reference camera. Finally, we decomposed the essential matrices to obtain the extrinsic parameters (i.e., rotation matrix and translation vector) for each camera. Finally, we perform a bundle adjustment over the intrinsic and extrinsic parameters.

Next, we calibrated the fixed cameras using the PTZ cameras as reference. Each fixed camera was calibrated with respect to the closest PTZ camera, changing the pan and tilt of this camera to obtain the best image overlap. As before, the scale bar was imaged in different positions to obtain feature points, and the essential matrix decomposed to obtain the extrinsic parameters. Finally, since each fixed camera was calibrated with respect to a different reference,

the results are unified to a common coordinate system by a series of appropriate rotations and translations.

Finally, we perform a bundle adjustment over all the cameras and observed feature points in the system, minimizing the sum of variances of the reconstructed 3D positions of each point used in the calibration using the Levenberg-Marquardt Algorithm.

To evaluate the calibration precision, the length of the scale bar was reconstructed based on the estimated extrinsic parameters. The errors in the length measurement ranged from 1-4 mm (i.e., 0.1-0.3%), indicating a high-quality result. Figure 2 shows the results of the extrinsic calibration.

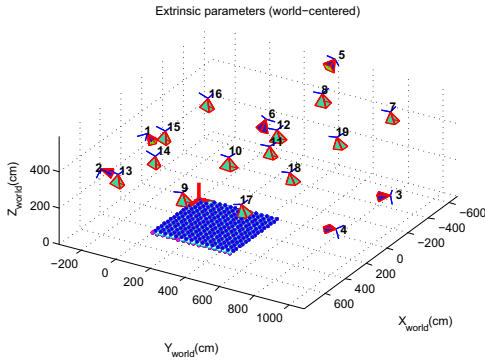


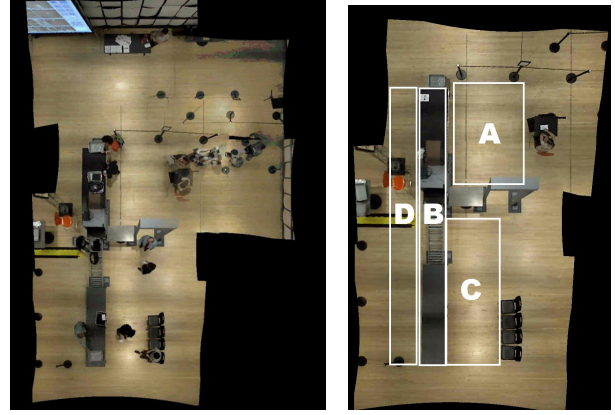
Figure 2. Extrinsic calibration results, indicating the cameras' positions with respect to the floor.

After calibration, we know the relative positions of all the cameras. Therefore, we can stitch the images of the floor taken by all the cameras together into one mosaic image, which is well-suited for display and tracking. We create the floor mosaic with a suitable combination of projective transformations, blending the image intensities at each pixel based on its distance to the boundary of each image plane containing it. We applied histogram equalization to normalize the white balance for all the cameras prior to blending. Figure 3(a) illustrates one frame of the mosaic image created from all the overhead cameras. Figure 3(b) superimposes the main areas of interest for our application onto the mosaic image, which will be discussed further in Section 6.

5. Tracking

We now describe our methods for robust, accurate real-time object tracking, with customizations for both passenger and bag tracking. The main component of both trackers is the mixture-of-Gaussians method [9]. That is, we use an adaptive background subtraction method that models each pixel intensity as a mixture of K Gaussians, and update the model parameters at each frame.

The probability of a pixel having color X_t at time t is



(a) Stitched Image

(b) Areas of interest

Figure 3. (a) Example frame of the mosaic image. (b) Labeled areas used for tracking and association. A is the Bag Drop area, B is the Conveyor Belt area, C is the Pick Up area, and D is the Staff area.

defined as:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \cdot G(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

$$G(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} \quad (2)$$

Here, $\omega_{i,t}$ is the weight of the i^{th} Gaussian at time t . We assume the RGB channels are independently distributed and share the same variance in order to simplify the computation. Thus, each covariance matrix is diagonal:

$$\Sigma_{k,t} = \sigma_{k,t}^2 \mathbf{I}_{3 \times 3} \quad (3)$$

At each pixel and time t , we determine the Gaussian mixture component k giving the highest probability of observing this color, and update its weight and parameters as:

$$\begin{aligned} \omega_{k,t} &= (1 - \alpha)\omega_{k,t-1} + \alpha \\ \mu_{k,t} &= (1 - \rho)\mu_{k,t-1} + \rho X_t \\ \sigma_{k,t}^2 &= (1 - \rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})^T (X_t - \mu_{k,t}) \end{aligned} \quad (4)$$

Here, α is a learning rate that we set to 0.1 in our system. ρ is a factor for adapting current distributions that we set to 0.05 in our system. We update the weights on the remaining mixture components ($i \neq k$) as:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} \quad (5)$$

We then re-sort the components in order of decreasing weight.

The foreground pixels corresponding to moving objects are separated from the background by defining the background model as the B mixture components comprising $T\%$ of the total weight:

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k > T \right) \quad (6)$$

In order to eliminate noise and reduce computation, we subtract the first frame of video (which normally contains no passengers and bags) from all subsequent frames of video. Then we model the background using $T = 0.3$, since pixels in active areas are frequently crossed by moving objects.

Thus, each pixel at each point in time is classified as foreground or background. We next segment the foreground pixels into blob regions using a two-pass connected components algorithm [4]. We use Kalman filtering in the prediction for each Gaussian in order to robustly track lighting changes. Figure 4 shows a sample result of the instantaneous foreground/background segmentation. Blobs detected in the conveyor belt area (Area B in Figure 3(b)) are treated as bags, while blobs detected elsewhere are detected as passengers.

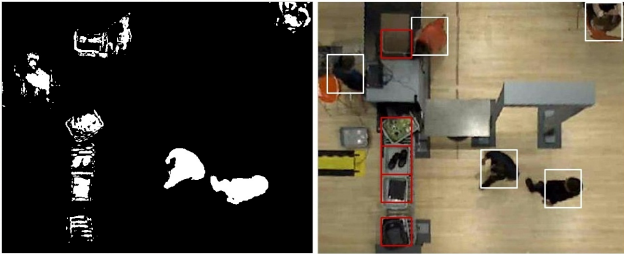


Figure 4. (a) Foreground extracted from the image. (b) Classification of detections as bags (red boxes) or passengers (white boxes).

5.1. Passenger Tracker

The detected passenger blobs are relatively easy to track against the light-colored floor. We apply the standard approach of associating blobs that have minimal Euclidean distance between two consecutive frames, which can handle most of the tracking task. However, we do have to deal with several merge-and-split events (for example, when a passenger interacts with a gate agent). It is critical to preserve correct passenger labels throughout such events.

Our efficient algorithm maintains the centroid, bounding box and orientation of each passenger blob. The orientation of the blob is defined as the angle between the x-axis and the major axis of the ellipse that has the same second-moment matrix as the blob. When two blobs merge, we maintain individual passenger positions as points on the major axis of the merged blob, separated by 60% of the major axis length. This leads to accurate passenger labels and loca-

tions throughout the merge and split process, as illustrated in Figure 5.

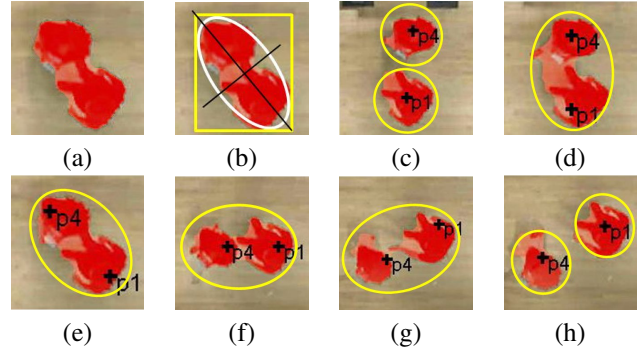


Figure 5. Updating labels when two blobs merge. Each merged blob (a) is modeled as an ellipse (b). The passenger labels and locations are maintained at points along the major axis of this ellipse (c)-(g) until they separate (h).

5.2. Baggage Tracker

Bags move along the conveyor belt region, the most critical area to monitor. With the exception of large suitcases, bags and other passenger belongings are put into regulation-size bins. Since the bags and bins only move along the one-dimensional conveyor belt region, the tracking problem is more constrained. However, bag tracking is more complex than passenger tracking, since bags undergo complex interactions and are both smaller and lower-contrast.

Since the conveyor belt region’s background is dark, dark-colored bags are difficult to detect. Hence, we apply histogram equalization to the foreground image, as illustrated in Figure 6. Figure 6(a) shows a black bag on the conveyor belt. Figure 6(b) is the detected foreground image, in which the black bag is nearly invisible. After histogram equalization of the foreground image, we get a grey-scale image as shown in Figure 6(c), in which the black bag can be easily detected as a blob. The final result is shown in Figure 6(d).

A bigger problem is that bags typically get extremely close to each other, making them hard to differentiate. The simple blob tracker often returns a connected component containing more than one bag. In order to separate them, we use the following method.

For each tracked blob with an area larger than a threshold (i.e., twice the area of a regulation-size bin), we try to evenly divide the blob into n parts, varying n from 1 to 5. We assign a score $f(n)$ to each number of parts as

$$f(n) = \frac{\frac{1}{n} \sum_{i=1}^n std(I_i)}{std([\text{mean}(I_1) \quad \text{mean}(I_2) \quad \dots \quad \text{mean}(I_n)])} \quad (7)$$

Here, I_n is the vector of pixel intensities corresponding to the i^{th} division of the blob. We select the first n for which

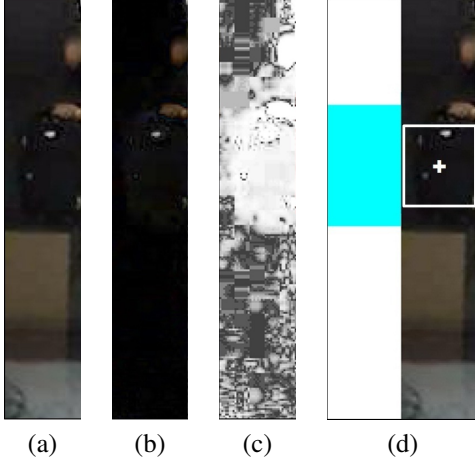


Figure 6. Using histogram equalization to detect dark-colored bags. (a) Original image. (b) Foreground extracted. (c) Foreground after histogram equalization. (d) Detected connected component and blob (the size of the annotation box is not related to the actual size of the bag).

$$f(n+1) - f(n) < \varepsilon \quad (8)$$

where ε is a threshold. Figure 7 shows the result of separating bags from a single detected blob using this method. The mean and standard deviation of the intensity for each labeled bag can also help to disambiguate the bags.

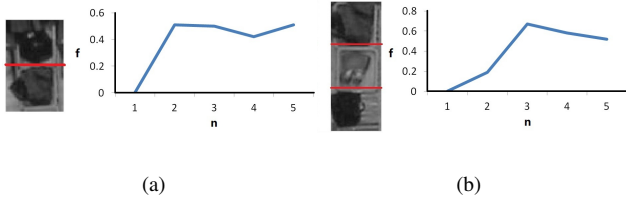


Figure 7. Result of splitting bags. (a) When there are two bags in a connected component, $f(n)$ increases sharply from $n = 1$ to $n = 2$ and remains almost unchanged from $n = 2$ to $n = 3$. (b) When there are three bags in a connected component, $f(n)$ peaks at $n = 3$.

6. Analysis and Baggage Association

The main purpose of the system is to maintain the association between bags and passengers in real time, using the results from the trackers described above. For the purposes of association, we define four semantically important zones, as depicted in Figure 3(b). These are the **Bag Drop Area**, in which passengers approach the conveyor belt area and place bags and bins onto the belt; the **Conveyor Belt** upon which bags move; the **Pick Up Area**, in which passengers recover their bags; and the **Staff Area** from which agents

can rearrange bags on the conveyor belt. We now describe the processing that takes place in each region.

6.1. Bag Drop

There are two analysis tasks for the Bag Drop Area, as illustrated in Figure 8. First, we label the passengers as they enter the region from the boarding pass checkpoint. Since the passengers singly enter this area from one direction, it is not difficult to label each passenger and filter out other objects and noise. We retain several properties for each passenger over the 10 most recent frames, including position in the area, size (area of the connected component), direction of motion, and average intensity. We assign each passenger blob detected in the current frame the label of the passenger blob in the previous frame with the closest parameters (i.e., minimum distance between feature vectors). If no blob detected in the current frame matches a label in the previous frame, we assume that the object is temporarily stationary, and its parameters should remain unchanged. If the label remains unmatched for several consecutive frames and the most recent position was close to the exit boundary of the Bag Drop Area, the passenger label is put into a queue for entering the Pick-Up area.

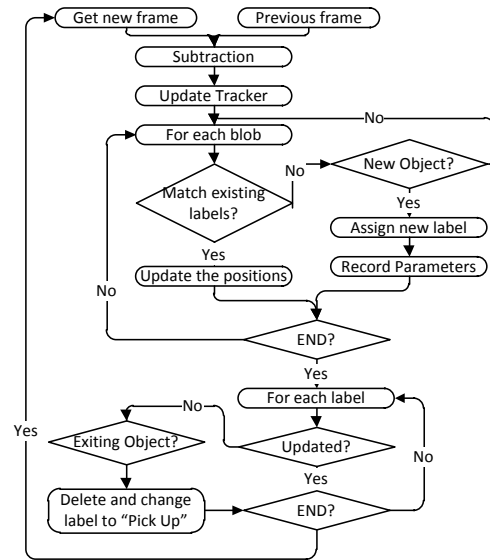


Figure 8. The processing flowchart for the Bag Drop Area.

The second task in the Bag Drop Area is to initially assign bags to passengers. When a new bag blob is detected in the Conveyor Belt area, we associate it with the label of the nearest passenger blob. More than one bag can be associated with a passenger.

6.2. Conveyor Belt Queue

The state of each bag is maintained using a finite state machine, as illustrated in Figure 9. The expected series of states for a bag is New, On Belt, Pending Decision, and Normal (meaning the correct person picked up the bag). There are also two queues containing bag labels: one for bags on the belt, and one for bags that have been removed from the belt by staff.

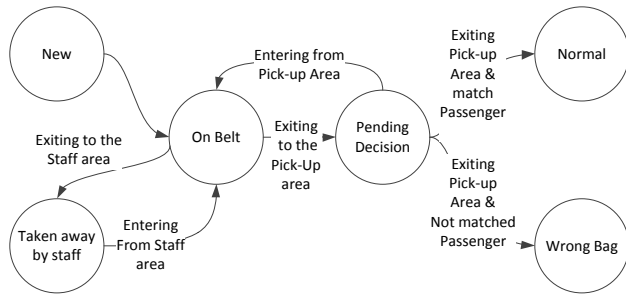


Figure 9. State machine for bag labels.

In practice, bags are frequently re-scanned by staff due to abnormalities detected by the X-ray machine. We simulated these events in our experiments; for example, bottles of water and laptops were put in some bags. After such bags go through the X-ray machine, they are taken off the line for re-scan by staff, stopping the conveyor belt and reordering the bag queue. These events are characteristic of the airport surveillance problem and are important to process correctly. When no change in position is detected for most of the bags for several consecutive frames, the conveyor belt area enters a “paused” state in which it is easy to detect the label of the bag that has been taken away from the belt. This label is removed from the belt queue and added to the re-ordering queue.

When the re-ordering queue is non-empty, the algorithm pays special attention to the area immediately in front of the X-ray machine intake. Normally, a staff agent will move the bags in line for the X-ray machine, insert the bag that must be rescanned, and restart the conveyor belt. We track the blobs in the Staff Area to determine when they move from the exit of the X-ray machine to its intake to determine when and where in the bag queue a re-ordered bag should be inserted. We compare the characteristics recorded for the bags in the re-ordering queue to new blobs on the conveyor belt entering from the staff side. The bag queue is updated accordingly upon an acceptable match (i.e., the property vector error is below a threshold). Otherwise we search the whole conveyor belt queue to match the label being re-ordered and find the best match.

Figure 10 summarizes the processing flow of the conveyor belt area.

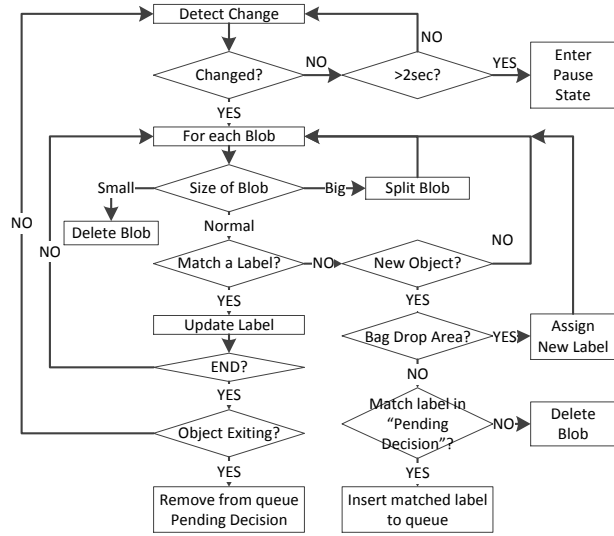


Figure 10. The processing flowchart for the conveyor belt area.

6.3. Pick-Up Area

The Pick-Up Area is crucial for deciding whether the bags were picked up by the right passengers. When a bag exits the X-ray machine, it enters the Pending Decision state. When it leaves the conveyor belt area, we detect the passenger blob with which it merges, and compare the bag owner label with the bag recipient label. to determine whether the association is correct (Normal) or incorrect (Wrong Bag). Decisions are not finalized until after passengers leave the tracking area (e.g., in case a bag is temporarily picked up or moved by someone other than its owner).

7. Experimental Results

We tested the tracking and association algorithms on 4 complete runs of simulation (25 minutes total). The first run was used to tune parameters for the algorithms, which were unchanged for the remaining runs. In all, 47 passengers and 71 bags were tracked and associated. Table 1 reports the results of the experiment. The first two rows report the performance of the tracker on bags and passengers. We can see the passengers were tracked flawlessly, while we had 3 false alarms and 4 misses for the bags. We found these errors to result from “ghost images” introduced when we created the floor mosaic image, since the belt and bags are significantly higher than the floor plane.

The second two rows report the performance on the association task. Out of the 71 true bags, 69 of them were normal and 2 were wrong-bag events that we introduced. The algorithms detected the vast majority of the events successfully. There is one false alarm indicating that a bin was

Tracker	Ground Truth	Detected	False alarm
Passengers	47	47	0
Bags	71	67	3
Normal	69	64	0
Wrong Bag	2	2	1

Table 1. Result of the experiment

mistakenly taken by another passenger. In fact, the owner of the bin removed their belongings from it, and the bin was re-arranged with other bins by another passenger.

Figure 11 shows several sample frames of a tracking and analysis result. From Figure 11(a)-(b), we see that passenger 2 has multiple bags on the conveyer belt (2 bins and a bag), and that they are correctly associated to the owner. In Figure 11(a), one of the bins is going through the X-ray machine and thus is not visible, but its label remains stable inside the X-ray machine. We see from Figure 11(f) that the passenger tracker can effectively track all the targets even when the Bag Drop area is crowded. From Figure 11(d)-(e) we see that the passenger going through the metal detector is tracked precisely. Please refer to the accompanying video for a better view of the results.

Figure 11(d)-(f) shows examples of two difficult situations. In Figure 11(d), passenger 2 has taken away his bags and left the monitoring area; however, the bin he used remains on the table in the Pick-Up area. In Figure 11(e), passenger 3 removes his belongings from his bin and puts his bin on top of the bin used by passenger 2. Hence, the labels of the two bins overlap. In general, it is difficult to tell when a bin is empty and should no longer be associated with a passenger, which may cause false alarms in association. In Figure 11(e) we see a staff member beginning to re-scan a bin belonging to passenger 4, removing it from the Pick-Up area. In Figure 11(f), the bin has been placed in the Bag Drop area ready for re-scanning. Bags to be re-scanned are often crowded by other bins/bags, increasing the likelihood of mixing labels.

8. Conclusions and Future Work

We discussed the design of an airport security checkpoint surveillance system. We addressed the setup and calibration of a large camera network and algorithms for tracking and associating passengers and bags. The system is robust to the crowded scenes and complex interactions typical of the airport environment.

From the experimental results we can see that while most of the bags were correctly tracked and associated, some complicated situations remain. For example, staff may merge or rearrange bin contents or introduce entirely new bins never touched by a passenger. In more complex videos not analyzed here, families enter the checkpoint and the

person-to-bag association problem is much harder (and correct association in the pick-up area is less critical, as long as all bags are recovered by the correct family). We plan to investigate more robust and adaptive tracking and association algorithms for such cases. Additionally, we want the system to be capable of detecting and understanding more types of abnormal behaviors besides baggage mis-associations, leveraging the images from the continuously-panning PTZ cameras. We also imagine an active vision system in which the PTZ cameras adaptively focus on abnormal or difficult-to-process events. Finally, we hope to extend our camera network tracking testbed to further applications, including security monitoring and anomaly detection, flow analysis in crowded scenes, and other large-scale environment simulations.

References

- [1] M. Bhargava, C. Chia-Chih, M. Ryoo, and J. Aggarwal. Detection of abandoned objects in crowded environments. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 271–276, 2007.
- [2] G. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 594–601, 2006.
- [3] R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [4] B. Horn. *Robot Vision*. The MIT Press, 1986.
- [5] N. Krahnstoeber, P. Tu, T. Sebastian, A. Perera, and R. Collins. Multi-view detection and tracking of travelers and luggage in mass transit environments. In *Proc. 9th IEEE Intl. Workshop Perform. Eval. Track. Surveill.*, pages 67–74, 2006.
- [6] M. Li, Z. Zhang, K. Huang, and T. Tan. Rapid and robust human detection and tracking based on omega-shape features. In *16th IEEE International Conference on Image Processing*, pages 2545–2548, 2009.
- [7] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia. Left-luggage detection using Bayesian inference. In *Proc. 9th IEEE Intl. Workshop Perform. Eval. Track. Surveill.*, pages 83–90, 2006.
- [8] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 205–210, 2004.
- [9] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.
- [10] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 259, 2003.
- [11] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2002.

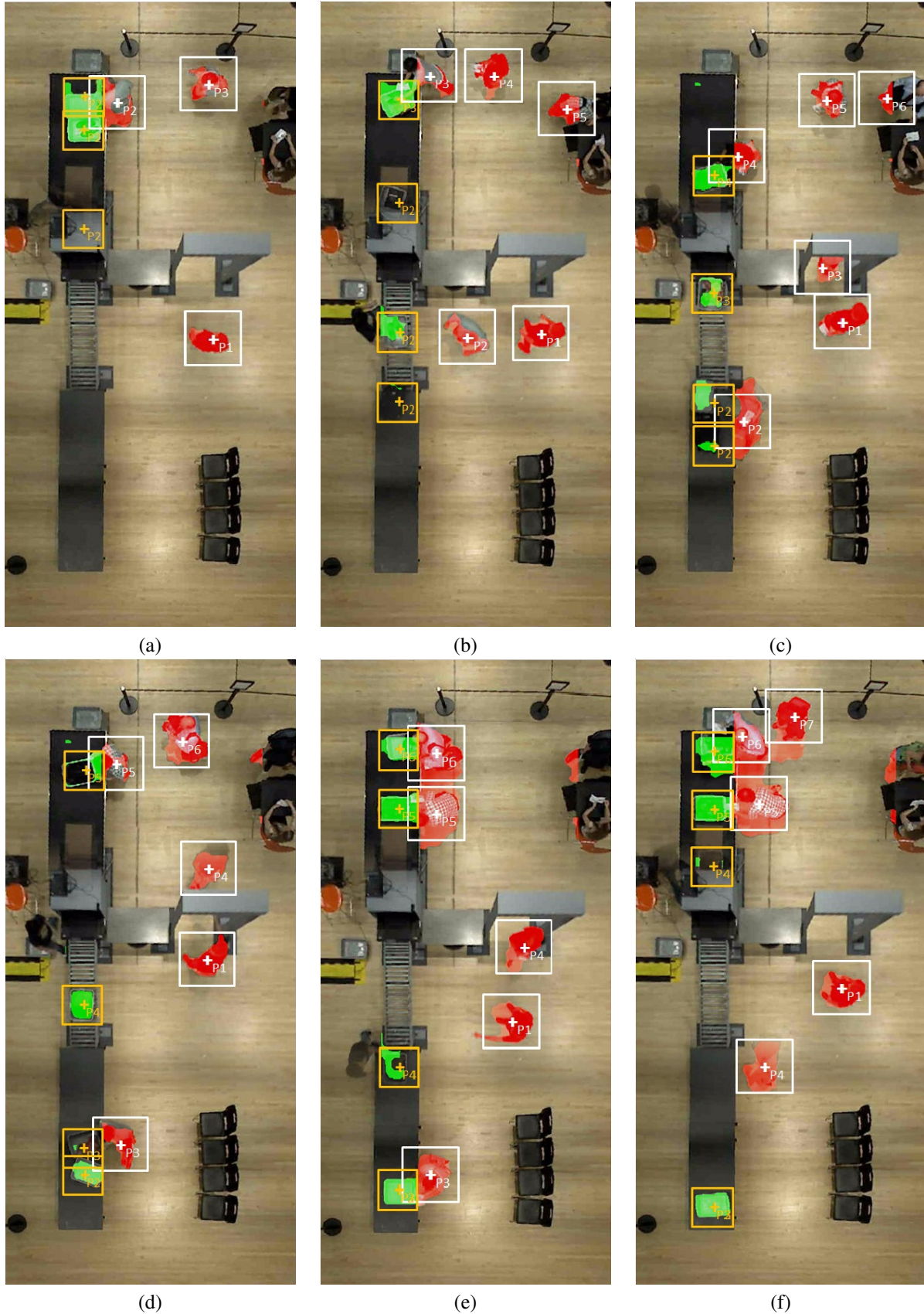


Figure 11. Example results from one experiment at (a) 420, (b) 550, (c) 700, (d) 830, (e) 894 and (f) 1058 frames. Bags on the belt are given the label of their associated passenger.