Correlating Belongings with Passengers in a Simulated Airport Security Checkpoint

Ashraful Islam
Rensselaer Polytechnic Institute
Troy, NY
islama6@rpi.edu

Yuexi Zhang
Northeastern University
Boston, MA
zhang.yuex@husky.neu.edu

Dong Yin
Northeastern University
Boston, MA
yin.d@husky.neu.edu

Octavia Camps
Northeastern University
Boston, MA
camps@coe.neu.edu

Richard J. Radke
Rensselaer Polytechnic Institute
Troy, NY
rjradke@ecse.rpi.edu

ABSTRACT

Automatic algorithms for tracking and associating passengers and their divested objects at an airport security screening checkpoint would have great potential for improving checkpoint efficiency, including flow analysis, theft detection, line-of-sight maintenance, and risk-based screening. In this paper, we present algorithms for these tracking and association problems and demonstrate their effectiveness in a full-scale physical simulation of an airport security screening checkpoint. Our algorithms leverage both hand-crafted and deep-learning-based approaches for passenger and bin tracking, and are able to accurately track and associate objects through a ceiling-mounted multi-camera array. We validate our algorithm on ground-truthed datasets collected at the simulated checkpoint that reflect natural passenger behavior, achieving high rates of passenger/object/transfer event detection while maintaining low false alarm and mismatch rates.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Tracking**; *Object detection*;

KEYWORDS

Airport security, camera networks, video analytics

ACM Reference Format:
Ashraful Islam, Yuexi Zhang, Dong Yin, Octavia Camps, and Richard J. Radke. 2018. Correlating Belongings with Passengers in a Simulated Airport Security Checkpoint. In *International Conference on Distributed Smart Cameras (ICDSC '18), September 3–4, 2018, Eindhoven, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3243394.3243703

ICDSC '18, September 3–4, 2018, Eindhoven, Netherlands

ACM ISBN 978-1-4503-6511-6/18/09…$15.00
https://doi.org/10.1145/3243394.3243703

1 INTRODUCTION

Video surveillance is a critical aspect of airport security, and can make a particular difference at security screening checkpoints. For example, automatically tracking the flow rates of passengers can help determine when a new lane should be opened. Robustly maintaining associations between passengers and their belongings can help detect thefts in real time, or mitigate ownership disputes after the fact. Maintaining a specific passenger's identity as s/he moves through the airport can enable "risk-based screening" in which certain passengers are given more or less scrutiny. In this paper, we present a first step in these directions: a set of computer vision algorithms specifically tailored to the problems of tracking and associating passengers and divested objects at security checkpoints. The algorithms were designed and tested in a highly accurate reproduction of a security checkpoint, ensuring their direct applicability to the real-world scenario.

In particular, we address three main problems. The first is passenger tracking through a network of several cameras with slightly overlapping fields of view. We combine optical flow detectors with a deep-learning detector trained on overhead passenger images. The second problem is tracking passenger belongings (here, confined to the contents of standard-sized bins). A passenger can divest objects into one or more bins. We solve the bin tracking problem with a background-aware correlation filter, and use a simple template matching algorithm to determine whether a bin is empty or contains divested objects. The third problem is passenger-to-bin association, which is important both at the moment of divestment and at the table where passengers pick up their belongings after screening. Since the problem of bin ownership is critical, we use a deep-learning-based arm pose detector to detect passenger contacts with bins. We test and train our algorithms with footage collected from the mock checkpoint, and validate the results against hand-annotated ground truth for passenger/bin locations and transfer events. We demonstrate that the algorithms operate at high detection and low false alarm rates, indicating their promise for real-world deployment.

[1]This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2013-ST-061-ED0001-04 and by the National Science Foundation under Grant Numbers IIS-1318145 and ECCS-1404163. 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 Deanna Beirne, John Beaty, and Carl Crawford for the design, data collection, and management of the simulated checkpoint. Thanks to Tim Rupprecht for the evaluation tool. Thanks to Mengran Guo for additional help on algorithm development.

2 RELATED WORK

Many modern video surveillance systems use some form of computer vision algorithm. One of the most common scenarios is the detection of abandoned objects. San Miguel et al. [12] proposed an approach to detect unattended or stolen objects in surveillance video based on three simple detectors. Initially, the moving regions are detected and then classified as static/dynamic and human/non-human objects. Next, objects that are detected as static and non-human are analyzed with each detector. The best detection hypothesis is selected based on a fusion-based approach to discriminate between stolen objects. Singh et al. [14] used a dual-time background subtraction algorithm and an approximate median model to detect abandoned objects. Lin et al. [9] adopted a similar approach for detecting abandoned luggage by combining short- and long-term background models to extract foreground objects. However, these approaches, while performing well in a simple environment for a particular task (detecting abandoned objects), are not suitable to solve the problems of tracking and association in a complex real-world airport surveillance system.

Several methods specially tackled multi-camera tracking problems. Stauffer et al. [15] used a planar tracking correspondence model to reliably track objects in multiple cameras with limited visual overlap. Chen et al. [3] used inter-camera transfer models to track objects in multiple cameras even with non-overlapping views. They generally worked with side-view cameras or cameras with large fields of view where calculating homography matrices between cameras is tractable, which is not the case in our system.

The most closely related work is by Wu et al. [16], who created a realistic airport checkpoint environment and a real-time system to track baggage and passengers and maintain correct associations. They used Gaussian mixture models to segment foreground blobs from the background and defined a state machine for bag tracking and association. Though the results were promising, the simulation environment was not entirely realistic in that the cameras were much further off the ground (10m) than would be practical, the camera image quality was relatively poor, and the illumination of the space was uncharacteristic (e.g., no natural light).

3 SIMULATION ENVIRONMENT

Our algorithms are trained and tested in a custom-built mock checkpoint testbed, illustrated in Figure 1, which is located at Northeastern University's Kostas Research Institute in Burlington, MA, USA. The testbed includes real-world airport equipment arranged at realistic scale and configuration, including rollers, bins, tables, podiums, and automated conveyor belts. The testbed also contains a real x-ray machine for baggage and a walk-through metal detector for passengers (neither of which is activated in our experiments). An accurately-sized plywood structure plays the role of a millimeter-wave advanced imaging technology (AIT) passenger screener.

The testbed contains many fixed-focus Bosch Flexidome IP 7000 RD cameras mounted on the 10-foot-high ceiling grid and pointed downward, five of which are used in this project to cover passengers' entry to the checkpoint, travel through the metal detector/AIT, and post-screening baggage pickup, as well as each bin's entire path (except when occluded by the x-ray). Each camera has a frame rate of 30 Hz and resolution of 1920×1080 pixels.

Figure 1: Divestment area of the mock checkpoint.

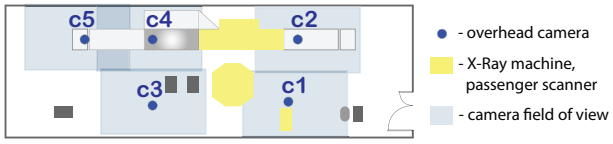


Figure 2: Floor plan of the mock checkpoint.

We label the cameras from 1 to 5 according to the typical path of the passengers, as illustrated in Figure 2. Camera 2 (divestment into bins) and Camera 4 (bin pickup) are the most important ones for our algorithm and the ones where our quantitative evaluation is performed.

Over several months, large groups of volunteers have been recorded in the testbed. Each of the volunteers was given an instruction card, directing him/her to divest items onto the conveyor, proceed through personal screening, pick up his/her divested items, and exit the screening area. In some cases, passengers were instructed to perform anomalous activities, such as taking an item from another passenger's bin. Many such group recordings were used to develop the algorithms described here; several recordings were manually annotated for quantitative evaluation, as discussed in Section 7.

4 PASSENGER DETECTION AND TRACKING

We combine two approaches for passenger detection and tracking: an optical flow-based segmentation and a deep learning-based person detection algorithm.

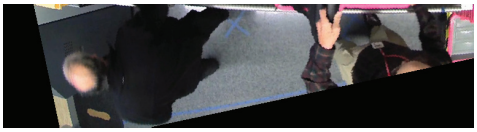
4.1 Single-Camera Detection

We adopt optical flow-based segmentation to extract passenger blobs from the background. Since the initial frames of the video normally contain no passengers, subtracting the initial frames from all subsequent frames of the video would be a natural approach. However, we found that quite often there are substantial matches between the color of the background and the passengers' clothes, in which case background subtraction fails. We found that incorporating motion-based segmentation is effective regardless of the color of passengers' clothes.

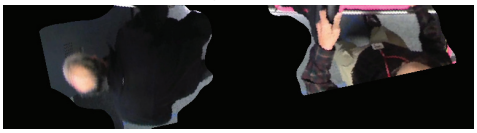
We incorporated a local-global optical flow method that is robust to local noise and at the same time produces dense flow fields, as suggested by Bruhn et al. [1]. We construct flow fields only in the passengers' areas (i.e., corresponding to the ground, not the conveyor). We then apply Gaussian blurring to the flow field, and assign the pixels that have flow magnitude greater than a threshold as foreground pixels belonging to the passengers.

We found that this flow-based method provides a crude estimate of each passenger's location. We then refine this region by incorporating a deep-learning based approach based on the Faster R-CNN architecture [11]. We decided to use this architecture because it has been shown to be capable of detecting multiple targets simultaneously and at fast speeds. Faster R-CNN attains this performance by using a region proposal network (RPN) that shares the full convolutional features used by the detection network to create region proposals (i.e., bounding boxes that are likely to contain an object of interest) almost for free. The network can then simultaneously predict object bounding boxes and objectness scores (i.e., the likelihood that the box contains the desired object) at every position. The two networks are trained together end-to-end by alternating fine-tuning for region proposal and object detection.

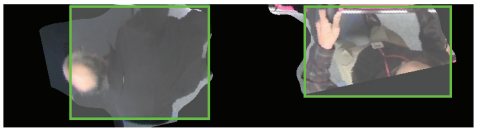
The off-the-shelf Faster R-CNN network is available pre-trained using the VGG-16 net [13], which in turn was trained with ImageNet [6]. In order to use this architecture for our passenger tracking problem, we fine-tuned it using hand-labeled data captured at the testbed. We labeled 676 frames containing persons whose head and upper body were visible. Retraining was done using a learning rate starting at 0.001, with a decay of 0.1 at 60,000 and 80,000 steps, and steepest gradient descent (SGD) with 0.9 momentum, and threshold of positive detection 0.7. Figure 3 illustrates the steps of passenger detection and refinement in an example frame.



(a) Original frame.



(b) Crude estimate from optical flow.



(c) Refined bounding boxes from R-CNN.

Figure 3: Steps for passenger detection.

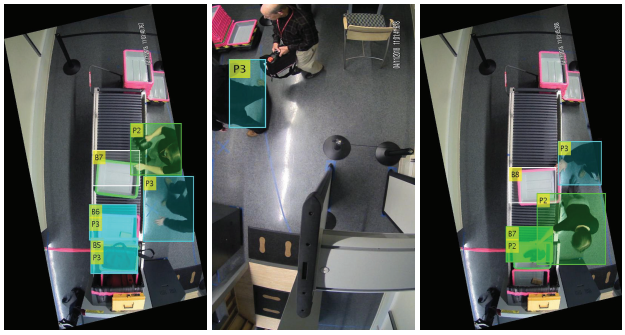
Optical flow also provides the direction of motion of the passengers, and hence the passengers' probable positions in the next frame. To track a passenger, we calculate the mean magnitude of the optical flow vector around the bounding box using a weighted average:

$$\bar{\theta} = \frac{\sum_{\boldsymbol{x} \in W} \theta(\boldsymbol{x}) M(\boldsymbol{x})}{\sum_{\boldsymbol{x} \in W} M(\boldsymbol{x})}$$

Here, $\boldsymbol{x}$ is the pixel coordinate in bounding box window W, and $\theta(\boldsymbol{x})$ and $M(\boldsymbol{x})$ are the direction and magnitude of the optical flow at location $\boldsymbol{x}$. From $\bar{\theta}$ we have an estimate of the direction of motion of the passenger. We search for a blob in this direction in the optical flow field of the next frame, associate this blob with the corresponding passenger, and refine the bounding box using R-CNN. If the passenger does not move, there is no blob from the optical flow field, and we keep the bounding box at the same location.

4.2 Multi-Camera Person Tracking

Referring to Figure 2, the general flow is that a passenger enters the scene in Camera 1, then appears in Camera 2 where s/he divests items. S/he next leaves Camera 2 and appears in Camera 1 again; then s/he walks to the Camera 3 and Camera 4 area where s/he retrieves items from bins, and finally leaves the scene through Camera 5. Although the passengers appear in Camera 1 initially, we start labelling and tracking the passengers in Camera 2, since the field of view of Camera 1 has high radial distortion and thus is not suitable for tracking effectively. We use the frames of Camera 1 when passengers leave Camera 2 (i.e., to track the passengers from Camera 2 to Camera 3 or to maintain correct labelling if passengers return to Camera 2's view, as illustrated in Figure 4). We also note that while there is overlap between cameras, it is so limited that we cannot calculate robust homography matrices between them. Since the cameras are synchronized, when a passenger leaves one camera s/he should immediately appear in the next camera in a predictable region, which we search to maintain the correct passenger label. The passenger tracking is finalized after s/he leaves Camera 5.



(a) Camera 2, 46.4 sec (b) Camera 1, 48.5 sec (c) Camera 2, 51 sec

Figure 4: (a) Passenger P3 leaves Camera 2's FOV, (b) appears in Camera 1, and (c) reappears in Camera 2 to grab a bin and put it on the conveyor belt.

5 DIVESTED ITEM DETECTION AND TRACKING

Passengers place bins that move along the conveyor belt area. Divested items like bags, backpacks, and wallets are placed into the bins. The current project is constrained so that only one divested item is placed into each bin, and no objects are placed on the belt

outside of bins. Since the bins generally move along the long axis of the rollers/conveyor belt/table and come out of the x-ray machine in the same order as they enter, bin tracking is generally easier than passenger tracking.

5.1 New Bin Detection

First, we detect each incoming bin in the conveyor belt area. Since the conveyor belt region is mostly dark and an empty bin is light gray, we detect an incoming bin simply by detecting a change of intensity profile in the conveyor belt region. In an ideal situation, if a grey bin is placed with perfect alignment on a black conveyor belt, the intensity profile along the direction of motion of the conveyor belt will be a rectangular window signal with the same length as the width of the bin and magnitude proportional to the height of the bin. Even though an incoming bin might not be placed with perfect alignment on the conveyor belt in practice, we can detect a bin by measuring the Euclidean distance between an ideal rectangular window function and the horizontal intensity profile of the incoming blob. In our case, we assume that the bin rectangular window is W =100 pixels wide and has grayscale intensity 140.

We subtract the current conveyor belt area from the first frame (where there are no bins on the conveyor belt) and calculate the intensity profile $t(m)$ by summing the intensity of this subtracted region in the horizontal direction (see Figure 5). We next measure the cross-correlation between the intensity profile $t(m)$ of the conveyor belt and the ideal window function $s(m)$:

$$C(n) = \sum_m t(m)s(m-n)$$

If $\max C(n) > \tau$ where τ is a threshold, we assign a new bin id to the belt region covering the horizontal length between $\hat{n}$ and $\hat{n}+W$, where $\hat{n} = \arg\max C(n)$.

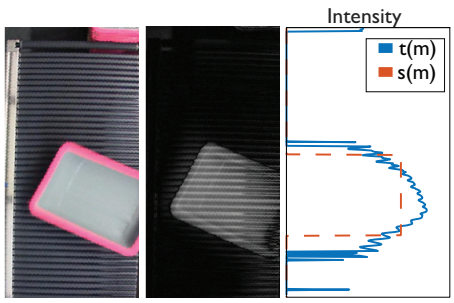


Figure 5: (a) Original image with incoming empty bin. (b) Result of subtracting static background (note that the "stripes" are due to the pattern of the underlying rollers). (c) Intensity profile of the foreground image compared to the window signal.

5.2 Bin Tracking

After assigning a unique id and bounding box to an incoming bin, we perform correlation filter (CF) based tracking to track the bin through the conveyor belt. CF trackers model an object using an adaptive correlation filter and track the target with convolution. A target window is selected in the first frame, from which the filter is initialized. The tracking is performed by correlating the filter over a search window in the next frame. The point where the correlation response is maximum is assigned as the new target location. The correlation is performed in the Fourier domain, so the tracking is extremely computationally efficient. Many methods have recently been proposed to exploit this filter-based tracking strategy; we adopt the Background Aware Correlation Filter (BACF) [7], which uses discriminative and well-generalized multi-channel features and is quite robust to initialization. Moreover, the BACF tracker uses the surrounding background around the target for negative samples at the training stage. We occasionally re-initialize the correlation filter when the appearance of the bin changes (for example, when an item is placed into the bin) so that the tracker becomes robust to sudden appearance changes of the target.

5.3 Divested Item Detection

We apply a template matching algorithm to determine whether there is any divested item (denoted DVI) inside the bin. There are several issues to consider. First, we do not have tight segmentation of the bin, so the bounding box of a bin may cover the background conveyor belt area and/or some parts of neighboring bins. Therefore, simple approaches like color histograms will not work in situations when the bin is diagonally aligned on the conveyor belt or the bounding box expands to nearby bins. Moreover, there might be sudden variations of color intensity of a bin for various reasons, such as a passenger's hand waving above the bin, placement of the bin in a shaded region, or angular displacement of the bin.

To overcome these issues, we adopt a robust template matching algorithm from Dekel et al. [5], named Best-Buddies Similarity (BBS) Template matching. In this algorithm, the Best-Buddies Pairs (BBP) are detected from two sets of points. A pair of points is called a Best-Buddies Pair if each point is the nearest neighbor of the other corresponding point. The ratio of the number of BBPs to the total number of points in the set is called the BBS. As BBS relies on a small subset of point pairs, it is robust to appearance or illumination changes. Also, instead of using actual distance values, BBS counts the number of Best-Buddies Pairs, which is a robust similarity measure.

We record a template image to represent an empty bin. Since the bounding box around a target bin might not be tight enough, we expand the bounding box window on both sides (≈20% of the bin width), and divide the window into several overlapping rectangular regions with the same area as the template image. We then apply the BBS measure between the template image and the candidate windows of the target image in RGB color space. The maximum BBS value represents the similarity between the template and the target. We store this value for several consecutive frames (30 frames in our case). If this similarity value is less than a threshold (0.5 in our method) for several frames, then we consider the bin as containing a divested item. Otherwise, we mark the bin as empty.

6 PASSENGER-TO-DVI ASSOCIATION

The main task of the system is to associate bins with passengers, maintain the associations, and detect whether there is an anomalous

event (e.g., theft). A simple approach to association is to measure the distance between the centroid of a bin and nearby passengers, and associate the bin with the nearest passenger. However, this approach fails in situations when there are several passengers in close proximity. Specifically, we need to determine which passenger actually interacts with items in a bin. We perform association in two situations: in Camera 2 when divestiture occurs (the DVI drop area) and in Camera 4 when the bin is emptied (the pick-up area).

6.1 Multi-Person Upper Body Pose Estimation

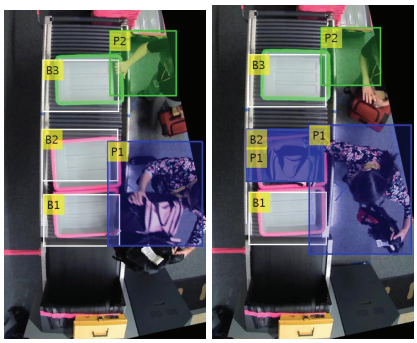
When the passengers are well-separated in the images, it is easy to associate them with their belongings and bins by simply using proximity. However, when passengers come closer to each other (as is common at security checkpoints), proximity alone does not work well, leading to incorrect associations. In order to overcome this challenge, we used an upper body detector to capture the events when the passengers divest or pick up personal items. In particular, we used a convolutional pose machine deep architecture network [2], fine-tuned to detect the two arms of each passenger.

The pose machine uses a multi-stage network with two parallel branches. A feature map of an image is first extracted using a VGG-19 network [13], and is then passed to the first stage of each branch. Afterwards, confidence maps of the locations of each body part are predicted by the first branch, and the second branch predicts the association relationship between the parts. The predictions from the two branches, along with the image features, are then delivered to the next stage.

To train this network, we built our own dataset following the training protocol from the COCO challenge [10] and the framework from CAFFE [8] to construct a database that can be properly read by the designed network. We prepared all the data by using a customized Sloth labeling tool [4]. We first defined the categories to be labeled. For each person, we labeled 7 joints in total: the head, two wrists, two elbows, and two shoulders. We used 364 frames from 3 videos from Cameras 2 and 4. During training, we fine-tuned the pre-trained model from [2] with the initial learning rate 8e−6 with decay of 5e−4 every 130,000 iterations. We trained 300,000 iterations in total.

6.2 DVI Drop Area

The passengers enter Camera 2 from Camera 1 after the boarding pass checkpoint. Since the passengers enter this area only from one direction, we label the passenger according to the sequence they enter. We label the bins in the same manner, i.e., by the order of entrance. Although the bins are labeled as soon as they are dropped onto the conveyor belt, we do not associate an empty bin with a passenger. When an item is divested inside the bin (determined by the BBS measure as discussed in Section 5.3), we keep track of the passenger whose left or right palm coordinates (determined by the deep learning-based pose estimation algorithm) are nearest to the centroid of the bin for 30 frames. If the bin is fully divested after that, we assign the bin to the passenger whose palm is closest for most of the frames (as illustrated in Figure 6). We do not alter this assignment until pick-up, so that passengers other than the owner can temporarily move the bin (which frequently happens in the airport environment). Moreover, we also save the order of the bin labels as they exit to the x-ray machine and the states of the bins so that they can be tracked after coming out of the x-ray machine in the pick-up area (i.e., Camera 4).



(a) (b)

Figure 6: (a) The bins are labeled but not associated. (b) Person P1 divests item in bin B2; thus B2 is assigned to P1.

6.3 Pick-Up Area

The pick-up area (Camera 4) is crucial for deciding whether there is a correct pick-up or a potential theft. Here, an important point to consider is that the bins come out of the x-ray machine in the same order as they enter the machine. Initially an incoming bin is detected by intensity profile matching (discussed in Section 5.1), and we assign the corresponding label and BACF tracker to the bin.

In this area, passengers pick up their items from the bins. Passengers are tracked through Cameras 2, 1 and 3 to Camera 4; thus we know the labels of each passenger. During a pick-up event, we detect the recipient as the passenger who is closest to the bin by measuring the distance between the bin centroid and palm coordinates. We determine whether there is a potential occurrence of theft by comparing the bin owner label with the bin recipient label, as illustrated in Figure 7. After the item is picked up, we no longer track the bin.

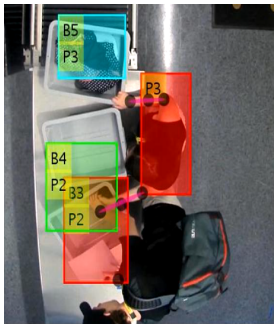


Figure 7: Potential theft: Person P3 is picking up item B3 that belongs to Person P2 (i.e., mismatch between owner and recipient labels). The arm pose points of P3 are visible here.

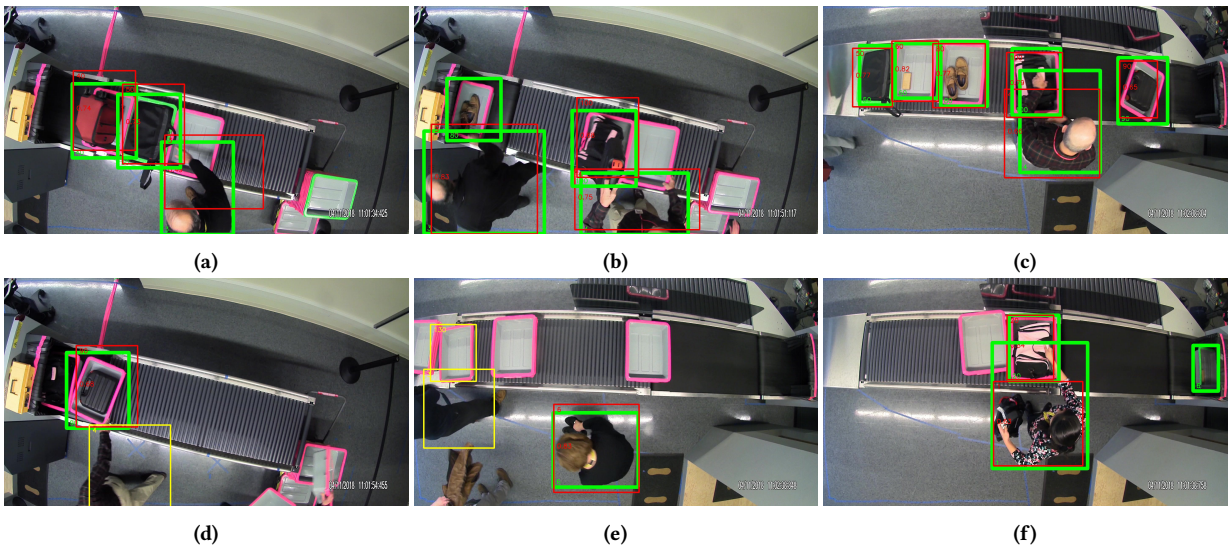


Figure 8: Sample evaluation results. Here, green symbolizes ground truth data, red symbolizes algorithm data, and yellow symbolizes false alarms for the algorithm data. (a)-(c) Sample frames with correct detections. (d), (e) Sample frames with false alarms for passenger detection, all of which are caused by partial views of the passengers. (f) A sample frame of a missed DVI detection. Here, the bin is just coming out of the x-ray machine. Our algorithm has not detected the bin yet, as it is not fully in the camera view.

7 EXPERIMENTAL RESULTS

Here, we report the results of our algorithm on two datasets that we denote "training" and "testing" (though many other videos were used in the design and tuning of the algorithms). For the training dataset, the ground-truth annotations of passenger and bin bounding boxes and timing of transfer events were known. However, for the testing dataset, the annotations were unknown to the researchers and evaluated by an independent "oracle". Both datasets consist of two full runs of passengers in the testbed; the training data includes five cameras having video length of 136 seconds each, and the testing data includes five cameras having video length of 132 seconds each.

For evaluation, we define the probability of detection, PD, as the number of true detections divided by the number of total detection events, and the probability of false alarm, PFA, as the number of false detections divided by the number of total events. The PD and PFA are evaluated for passengers, divested items, and transfer events (divestment or picking up of items from bins). A true detection for a passenger or DVI occurs when the intersection-over-union (IoU) of the detected bounding box and the ground truth bounding box is greater than a threshold; otherwise it is a false alarm. The default threshold is 0.3 for passenger (PAX) detection and 0.5 for divested item (DVI) detection. A transfer event is a "hit" (correctly detected) if it occurs within ±30 frames of the ground truth transfer. Moreover, the evaluation tool also calculates the probability of switch (both for PAX and DVI) and probability of mismatch (for transfer events). A switch is registered when there is any change of label of a DVI or PAX, and a mismatch is registered if a PAX-DVI association disagrees with ground truth on divestment or pick-up. Clearly, we want the PD to be high and PFA, switch and mismatch to be low.

The evaluation is performed only for Cameras 2 and 4, since the divestment and pick-up occur in these two cameras respectively. We use the other cameras for tracking the passengers so that there are few switch or mismatch events in the final results. The results on the training and testing datasets are tabulated in Table 1 below. All of the values are converted to percentages. The top row of Figure 8 shows example frames with correct passenger and bin detections.

Table 1: Experimental results

	training		testing	
Camera	2	4	2	4
PD(PAX)	81.5	88.6	95.0	100.0
PD(DVI)	87.5	91.4	91.4	92.0
PD(XFR)	88.9	75.0	87.5	62.5
PFA(PAX)	34.1	14.6	27.5	7.5
PFA(DVI)	7.3	12.2	25.0	25.0
PFA(XFR)	0.3	0.0	0.1	0.1
P(PAX Switch)	7.4	0.0	0.0	3.3
P(DVI Switch)	7.5	0.0	0.0	2.0
P(Mismatch)	0.0	12.5	0.0	6.2

The bottom row of Figure 8 shows several example frames where our algorithm disagrees with the ground truth annotations. We can see that there are subtle evaluation issues for which the algorithm results in "false alarm" or "miss". One issue is that the ground truth denotes someone as a passenger only when his/her head appears. However, our algorithm detects passengers even

when some body parts are partially visible. The high false alarm in the passenger tracking is mainly due to this evaluation issue. The same goes for DVI detection; there are disagreements between the ground truth and our algorithm with respect to when should we start or stop labeling a DVI, which are difficult to resolve automatically or with a heuristic rule. The video at https://www.youtube.com/watch?v=CJqeqhKfNFk shows an example of the algorithm, which additionally includes a "news feed" visualization of the automatically detected, time-stamped video events.

Figure 9 explores the algorithm performance as the intersection-over-union (IoU) threshold that determines a "hit" is varied for the training dataset. As expected, decreasing the IoU threshold increases the detection rates, though we note that the DVI detection rate is robust to changing the IoU threshold in the range between 0.2 and 0.6.

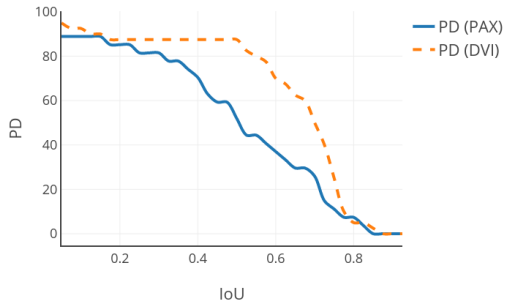


Figure 9: PD vs. IoU threshold

We also found a subtle ground-truthing issue with respect to transfer events. A transfer event occurs when a passenger divests an item into a bin or picks up an item. We found that the divestment event is triggered in the ground truth at the instant the item is placed in the bin. However, our algorithm waits several frames to be sure that it is a divestment rather than some illumination change or waving of passengers' hands. Though a transfer event is a "hit" within ±30 frames (1 second) of the ground truth event, we observed that a 1 second window is not enough to capture all transfer issues.

8 CONCLUSIONS

We presented an airport surveillance system for security checkpoints based on a highly accurate testbed and realistic passenger/divestment behavior. The proposed algorithms can accurately detect, track, and associate passengers and divested items, as well as detect complex activities like theft or person-to-bin mismatches. In addition, our algorithm is robust to changes in camera position, lighting and shadows. There are several frames in which the lighting changes dramatically but our algorithm is unaffected.

In order to use deep learning architectures for the problem, we had to manually label data to fine-tune the networks. While this is a tedious process, it should be noted that we were able to achieve very good performance with a very small labeled dataset. We are currently working on domain adaptation techniques with the goal of further reducing the number of labels needed to fine-tune or even eliminating the need for them altogether.

Despite the ground-truth annotation issues mentioned above, there is still room to improve the passenger detection algorithm, in particular by by decreasing false detections. Estimating accurate transformations to convert coordinates between cameras would make the tracking system more robust.

In short-term future work, we can use the described system to automatically produce statistics from the event log, such as histograms of the number of bags per passenger, time taken by each passenger/bag to clear the screening process, number of seconds a passenger is physically separated/out of the line-of-sight of their belongings, and so on. Longer-term work would involve making the system robust to common "corner cases" such as families with children, passengers in wheelchairs, and oddly shaped items that do not fit into standard bins. Families present a particular challenge (that may be too difficult to solve by video alone) in that it should be "legal" for one family member to pick up an object that was divested by another family member.

REFERENCES

[1] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. 2005. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision* 61, 3 (2005), 211–231.
[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
[3] Xiaotang Chen, Kaiqi Huang, and Tieniu Tan. 2014. Object tracking across non-overlapping views by learning inter-camera transfer models. *Pattern Recognition* 47, 3 (2014), 1126–1137.
[4] cvhciKIT. 2017. Sloth labeling tool. (Jun 2017). https://github.com/cvhciKIT/sloth
[5] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T Freeman. 2015. Best-buddies similarity for robust template matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
[7] H Kiani Galoogahi, Ashton Fagg, and Simon Lucey. 2017. Learning background-aware correlation filters for visual tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
[8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*.
[9] Kevin Lin, Shen-Chi Chen, Chu-Song Chen, Daw-Tung Lin, and Yi-Ping Hung. 2015. Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. *IEEE Transactions on Information Forensics and Security* (2015).
[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*.
[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*.
[12] Juan Carlos San Miguel and José M Martínez. 2008. Robust unattended and stolen object detection by fusing simple algorithms. In *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*.
[13] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).
[14] A Singh, S Sawan, Madasu Hanmandlu, Vamsi Krishna Madasu, and Brian C Lovell. 2009. An abandoned object detection system based on dual background segmentation. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*.
[15] Chris Stauffer and Kinh Tieu. 2003. Automated multi-camera planar tracking correspondence modeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
[16] Ziyan Wu and Richard J Radke. 2011. Real-time airport security checkpoint surveillance using a camera network. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.