Improving Counterflow Detection in Dense Crowds with Scene Features

Ziyan Wu*, Richard J. Radke

Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, USA 12180

Abstract

This paper addresses the problem of detecting counterflow motion in videos of highly dense crowds. We focus on improving the detection performance by identifying scene features — that is, features on motionless background surfaces. We propose a three-way classifier to differentiate counterflow from normal flow, simultaneously identifying scene features based on statistics of low-level feature point tracks. By monitoring scene features, we can reduce the likelihood that moving features' point tracks mix with scene feature point tracks, as well as detect and discard frames with periodic jitter. We also construct a Scene Feature Heat Map, which reflects the space-varying probability that object trajectories might mix with scene features. When an object trajectory nears a high-probability region of this map, we switch to a more time-consuming and robust joint Lucas-Kanade tracking algorithm to improve performance. We evaluate the algorithms with extensive

*Phone: +1 (518) 961-1754, Fax: +1 (518) 276-8715

Email addresses: wuz5@rpi.edu (Ziyan Wu), rjradke@ecse.rpi.edu (Richard J. Radke)

URL: www.rpi.edu/~wuz5 (Ziyan Wu), www.ecse.rpi.edu/~rjradke (Richard J. Radke)

experiments on several datasets, including almost three weeks of data from an airport surveillance camera network. The experiments demonstrate the feasibility of the proposed algorithms and their significant improvements for counterflow detection.
Keywords: counterflow, scene feature, tracking, false alarm, video surveillance

1. Introduction

Counterflow detection is a critical problem in security-related surveillance. For example, a person moving the wrong way through the exit corridor of an airport can prompt an entire terminal to be "dumped", resulting in hundreds of delayed flights and inconvenienced passengers. By tracking low-level feature points, the typical flow direction can easily be determined. However, most of the cameras deployed in security surveillance networks have poor resolution and quality compared to a consumer digital camera, which can negatively affect tracking algorithms, especially during long-term operation. Another issue preventing automatic video analytic algorithms from replacing manual monitoring is that the false positive rate is likely to be very high compared to the small number of true positives in 24/7 continuous operation.

This paper presents three contributions. First, we demonstrate that counterflow detection can be significantly improved by introducing a novel classifier to identify scene features in the image, which are then used to mitigate cases in which foreground and background features are mixed in the same point trajectory. Second, by monitoring the statistics of scene features, we identify jitter frames that should not play a role in tracking. Third, we con-

struct a Scene Feature Heat Map that enables the automatic selection of a suitable tracking scheme for point tracks in different locations of the image to achieve more robust performance. We conducted extensive experiments on both a standard dataset (CAVIAR) and several real-world video datasets acquired from an airport surveillance camera network, demonstrating that our counterflow detection algorithm is significantly improved by using the scene-feature-based algorithms. The resulting framework was in continuous operation for three weeks at a major airport, successfully detecting hundreds of counterflow events with no misses and only three false alarms.

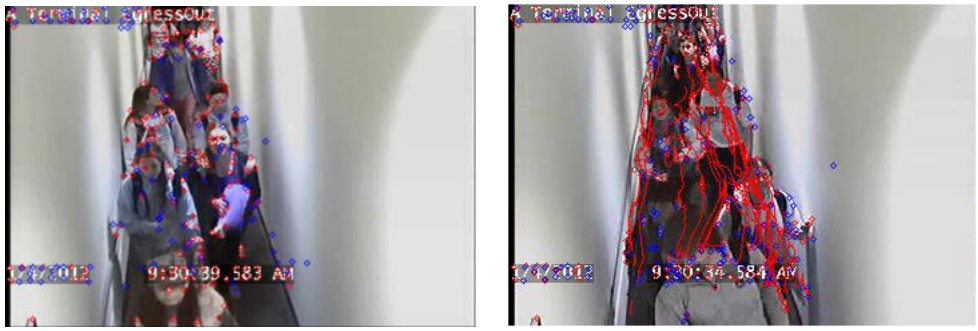
2. Related Work

The problem of detecting dominant motions in crowded video and classifying outlying motions has been widely studied [8, 13, 2, 4]. Tu and Rittscher [17] introduced a crowd segmentation algorithm by clustering interest points into groups by determining maximal cliques in a graph. However, both the algorithm and experiments are based on videos from overhead views only, which is the easiest case for counterflow detection. Andrade et al. [3] proposed an algorithm for detecting abnormal movements in crowds by applying principal component analysis to optical flow maps and spectral clustering to hidden Markov models, but did not perform any real-world experiments. This algorithm identifies abnormal motion based on a trained flow map, which is sensitive to noise and may cause false positives for normal motions not covered by the training set. Brostow and Cipolla [7] used an unsupervised Bayesian detection algorithm to segment low-level feature tracklets based on a spatial prior and a likelihood model of coherent motion. Ali and Shah [1]

modeled a highly dense crowd as an aperiodic dynamical system that can be studied with Lagrangian particle dynamics techniques. Antonini and Thiran [4] introduced a trajectory clustering method based on independent component analysis. Junejo et al. [13] applied graph cuts to segmenting tracklets. Cheriyadat and Radke [10] proposed a trajectory clustering algorithm based on non-negative matrix factorization.

Cheriyadat and Radke [9] proposed an automatic dominant motion detection method by clustering trajectories based on longest common subsequences. Since individual people are difficult to segment, the inputs to the algorithm are tracked low-level features obtained using optical flow. Our algorithm takes a similar approach. However, these types of algorithms might not yield good results in situations involving low-resolution cameras and poor image quality. Marcenaro and Vernazza [15] proposed an image stabilization algorithm based on feature tracking in which scene features are used as references to compensate for the motion of the camera. In this paper, we propose a classifier to identify scene features in the context of detecting counterflow motion. We show that by using information from the scene features, the performance and accuracy of foreground object point tracking can be improved under low-quality, complex-background conditions.

An earlier version of this paper appeared in [18]. Here, a new concept, the Scene Feature Heat Map, and a joint processing mechanism within a camera network are proposed in order to further reduce the false alarm rate. A new experiment on the CAVIAR dataset and a more extensive long-term experiment using a camera network at an airport are presented, demonstrating the effectiveness of the proposed algorithm.



(a) (b)

Figure 1: Results of feature tracking. (a) Features detected in the image. (b) Point tracks extracted from a video sequence.

3. Feature Tracking

Even in the age of high-quality consumer digital cameras, videos from surveillance camera networks are frequently low-resolution (e.g., 352×240). Since we want the system to process video streams from tens of cameras in real time, and the dominant (or allowable) direction of motion is all we need to know, we use low-level features to track the flow. We first identify low-level features in the initial frame using the FAST corner detector [16]. The features are then tracked over time using the Kanade-Lucas-Tomasi (KLT) optical flow algorithm [14], adapting the pyramid representation in [6], which can track large pixel motions while keeping the size of the integration window relatively small.

The results of feature tracking are shown in Figure 1, in which red circles indicate all the features detected in the current frame (up to a maximum number, e.g., 300) and blue circles indicate reliably-trackable features. New features are added to the tracker every 5-10 frames, discarding those too close

to current tracks. These feature tracks form a large trajectory set.

4. Improving Robustness with a Scene Feature Classifier

This low-level feature point tracking is often inaccurate, due to both the low resolution and quality of the input videos and periodic jitter. Consequently, it is common for features on foreground objects (corresponding to the allowable/counter flow) to mix or merge with stationary scene features, as illustrated in Figure 2.

Our solution to this problem is to build a three-way classifier to identify normal flow, counterflow, and scene features. The point tracks are classified at a specified interval (e.g., every 300 frames). The recognized scene features can also be used to compensate for location drift caused by jitter. Here we assume flow goes roughly up-and-down on the image.

Let $\mathbf{L}^j = \{(x_j(1), y_j(1)), \ldots, (x_j(n_j), y_j(n_j))\}$ be the data of the j^{th} point track. We define two features (d_1^j, d_2^j) for each trajectory $\mathbf{L}^j$ as

$$d_1^j = \frac{1}{n_j^2} \sum_{i=1}^{\lfloor \frac{n_j}{3} \rfloor} y_j(n_j - i) - y_j(i)$$

$$d_2^j = \frac{1}{n_j} \sqrt{\sum_{i=2}^{n_j} (x_j(i) - x_j(1))^2 + (y_j(i) - y_j(1))^2}$$

That is, d_1^j represents the difference in sum on y between the first third and last third of the trajectory, and d_2^j represents the variance of the points on the trajectory from their initial position.

As Figure 3 illustrates, the three-way classifier separates trajectories corresponding to normal flow, counterflow, and scene features based on the rule

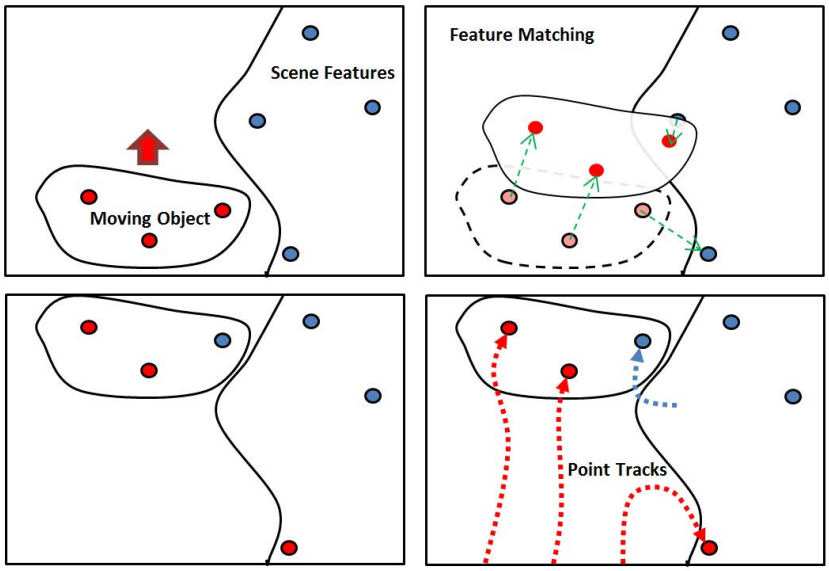
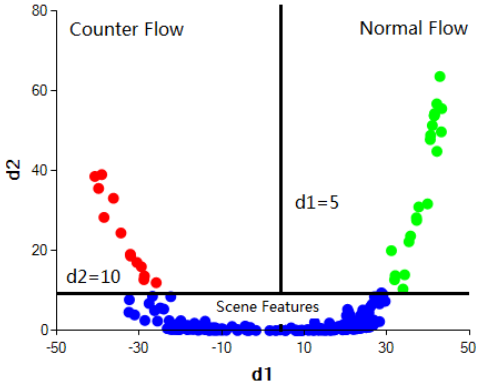


Figure 2: Foreground points mixing with scene points.

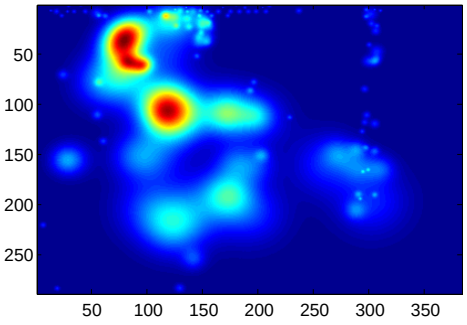
$$\mathbf{L}^j = \begin{cases} \text{normal flow} & d_1^j > a, d_2^j > b \\ \text{counterflow} & d_1^j \leq a, d_2^j > b \\ \text{scene feature} & d_2^j \leq b \end{cases} \tag{1}$$

The value of b in the classifier to separate scene points can be obtained by learning the standard deviation in the image location of features from an image sequence containing only the background. We used $b = 10$ in our experiments. The value of a is trained on a short sequence based on user editing of missed detections and false alarms. That is, a is set to an initial value (e.g., 5) and is adjusted based on user edits to the smallest number such that the classifier has no missed detections (which are operationally extremely costly).

(a)



(b)



(c)

Figure 3: Result of the three-way classifier and Scene Feature Heat Map. (a) Feature tracks. (b) Classifier result corresponding to (a). (c) Scene Feature Heat Map corresponding to (a).

After the scene features are classified, they can be used to deal with two issues. First, point tracks that were classified as scene points in the previous decision are matched and tracked only after all of the other (flow) features are matched and tracked for each frame. This step significantly reduces the probability that scene features mix with moving features and confuse the tracker/classifier, as we show in Section 7. Second, the statistics of scene features provide an easy way to detect frames with jitter, as shown in Figure 4. We can easily learn a threshold on the change in x values along the trajectories for scene points that detects jitter frames. These frames are then ignored for the purposes of tracking and classification, which substantially improves robustness.

5. Scene Feature Heat Map

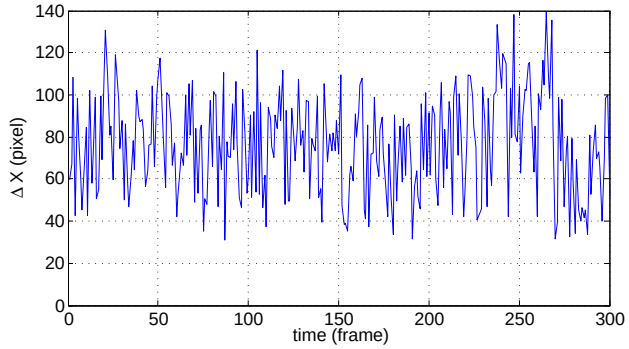
Using only the feature tracks classified as scene features $\mathcal{S}$, we generate a Scene Feature Heat Map (SFHM) to further reduce false alarms, defined as:

$$h(u,v) = \sum_{\mathbf{L}^j \in \mathcal{S}} \exp\left\{ \frac{-1}{2\sigma^2} \begin{bmatrix} u - x_j(n_j) \\ v - y_j(n_j) \end{bmatrix}^\top \Sigma^{-1} \begin{bmatrix} u - x_j(n_j) \\ v - y_j(n_j) \end{bmatrix} \right\} \tag{2}$$

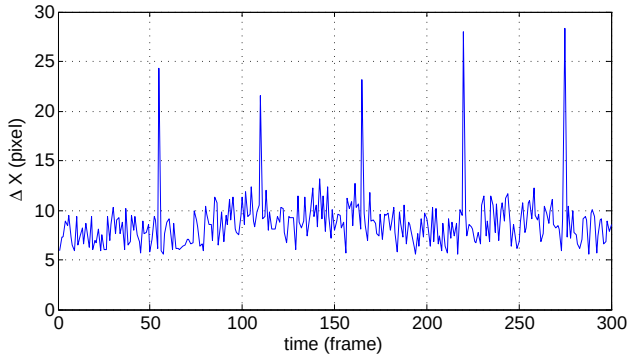
in which Σ is the 2×2 covariance matrix of point track data $\mathbf{L}^j$. σ is a scale factor depending on the image size, defined as

$$\sigma = b^{-2}\sqrt{(W^2 + H^2)} \tag{3}$$

in which W and H are the width and height of the image respectively. Figure 3(c) shows a Scene Feature Heat Map generated from the results shown in Figure 3(a-b). The SFHM is basically a visualization of the probability that a feature track at the given pixel contains a scene feature.



(a)



(b)

Figure 4: Statistics of (a) moving points and (b) scene feature points. Jitter frames are clearly visible as spikes in (b).

When a tracked feature moves close to a "high-heat" region on the SFHM, it is more likely to mix with scene features. Hence, in this case, we use a Pyramidal Joint Lucas-Kanade (JLK) algorithm [5] for feature tracking, which combines the Kanade-Lucas [14] and Horn-Schunck [12] optical flow algorithms. That is, we define the overall tracking problem at a pixel as

minimizing C by finding the displacement $\mathbf{t} = (t_x, t_y)^\top$, in which

$$C = \begin{cases} C_{LK} & h(x,y) < \beta\sigma \\ C_{JLK} = C_{LK} + \lambda C_{HS} & \text{otherwise} \end{cases} \tag{4}$$

The two cost functions are defined as

$$\begin{aligned} C_{LK}(u,v) &= \sum_{(x,y)} w(x,y) \left(I(x+u, y+v, t+1) - I(x,y,t)\right)^2 \\ C_{HS}(u,v) &= \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} \right\|^2 \end{aligned} \tag{5}$$

in which $w(x,y)$ is a windowing function and $(\hat{u}, \hat{v})^\top$ is the expected displacement computed by fitting an affine motion model to the displacements of N_e neighboring moving features weighted by their distance to the feature (we used $N_e = 5$ in our experiment).

In (4), β is a positive constant (we use 1.0 in our experiments.) This number can be set according to computational load and computational power; that is, the higher the available computational power, the smaller β should be set. λ is a smoothing term which we set to $\lambda = \sigma^2$.

The problem of minimizing $C = C_{LK} + \lambda C_{HS}$ can be solved using Jacobi iterations [5]. The JLK algorithm is more time consuming (usually 5-10 times slower than KLT), but with the help from neighboring moving features, the tracked feature points are less likely to merge or mix with scene features. The proposed SFHM method makes the tracking algorithm more robust while maintaining similar processing speed compared to KLT. More sample frames and their corresponding SFHMs and outputs from the classifier are shown in Figure 5.

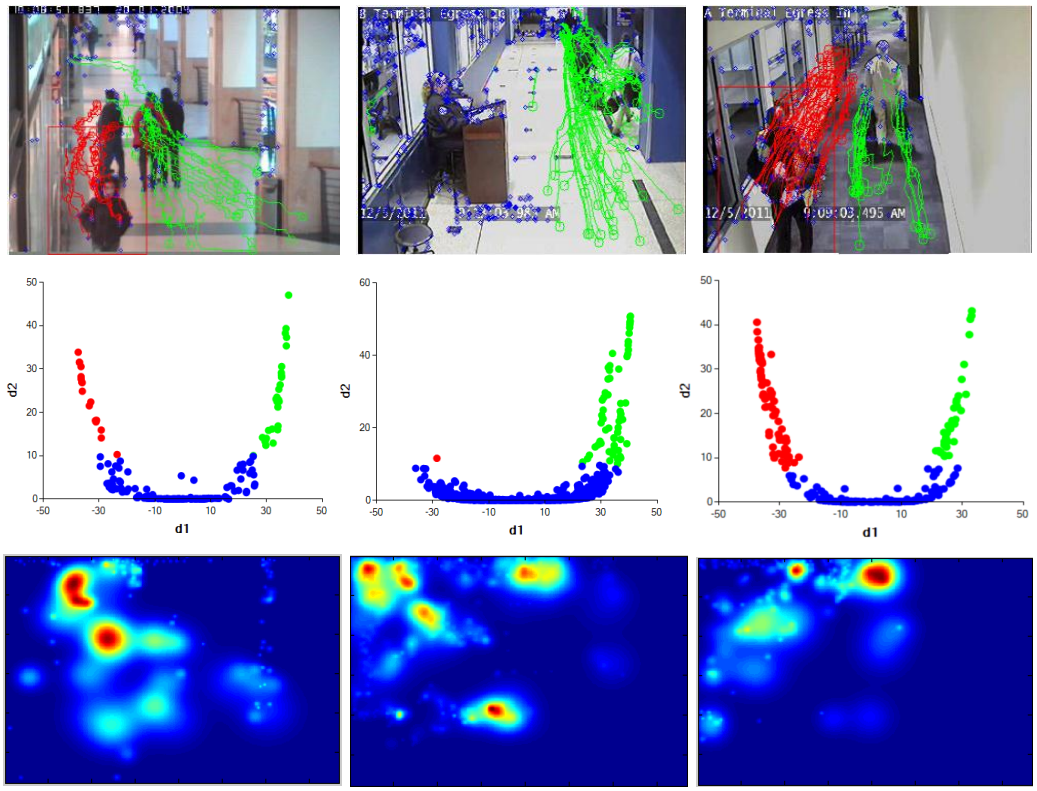


Figure 5: Classifier output and Scene Feature Heat Map for typical frames. The first row shows the frames processed with tracklets. The second row shows the output of the three-way classifier. The third row shows the corresponding Scene Feature Heat Map.

Features are tracked for every frame and are classified after a certain interval depending on the frame rate of the camera. Tracklets classified as counterflow are considered as potential alarms. Detection is finalized after an additional criterion. We found that most of the false alarms are caused by mixing tracks. Even with the three-way classifier, some of the mixed tracks are hard to filter since some regions contain both scene features and humans. However, the trajectories of mixed tracks are usually highly random. The

correlations between x and y coordinates are significantly lower than those of other trajectories. By setting a threshold (we used 0.5 in our experiments) on the correlation for trajectories, most false alarms caused by mixing point tracks can be eliminated.

6. Joint Processing in a Camera Network

In most cases, the configuration/floor plan of the camera surveillance network is known, which can be leveraged to further reduce the false positive rate. Even though all the cameras may not share overlapping fields of view, they generally cover the same path, which usually means that counterflow should be detected in more than one camera. The design of the joint decision mechanism depends on the physical setup of the camera network, as discussed further in the next section.

7. Experiments

We evaluated our algorithms on several datasets, both standard ones and custom-collected video at a large US airport (Cleveland-Hopkins International).

7.1. CAVIAR

First, we tested our algorithms on the standard CAVIAR dataset (clips from a shopping center in Portugal) [11]. 27 video sequences from the corridor view were used in the experiments, in which *WalkByShop1cor*, *2LeaveShop1cor* and *2LeaveShop2cor* were used to train the classifier.

The other 24 sequences were tested with and without our scene-feature based algorithm. As a baseline, we compare the results against a classifier

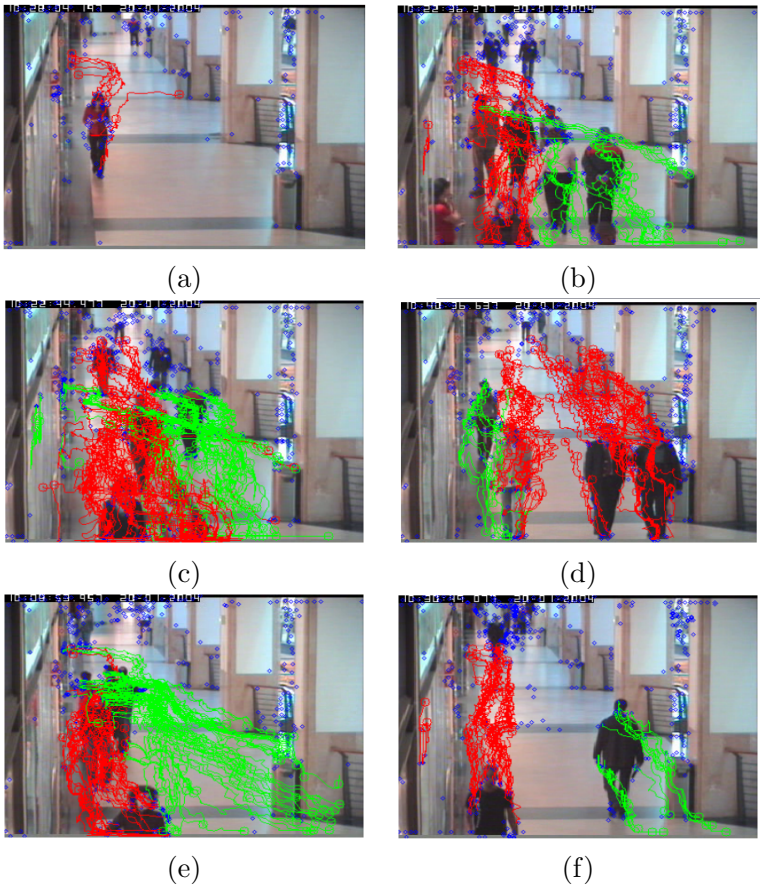


Figure 6: Sample images from CAVIAR with annotated results.

that only discriminates between normal flow and counterflow using the d_1 feature (i.e., not taking into account scene features). We also compared the results against the algorithms of LPD [1] and LCSS [9]. The results are shown in Figure 7. Several examples are shown in Figure 6. Normal trajectories are displayed in green while counterflow trajectories are displayed in red.

Since LCSS is based on a similar feature tracking approach, its performance is similar to the 2-class baseline approach; its false alarm rate is

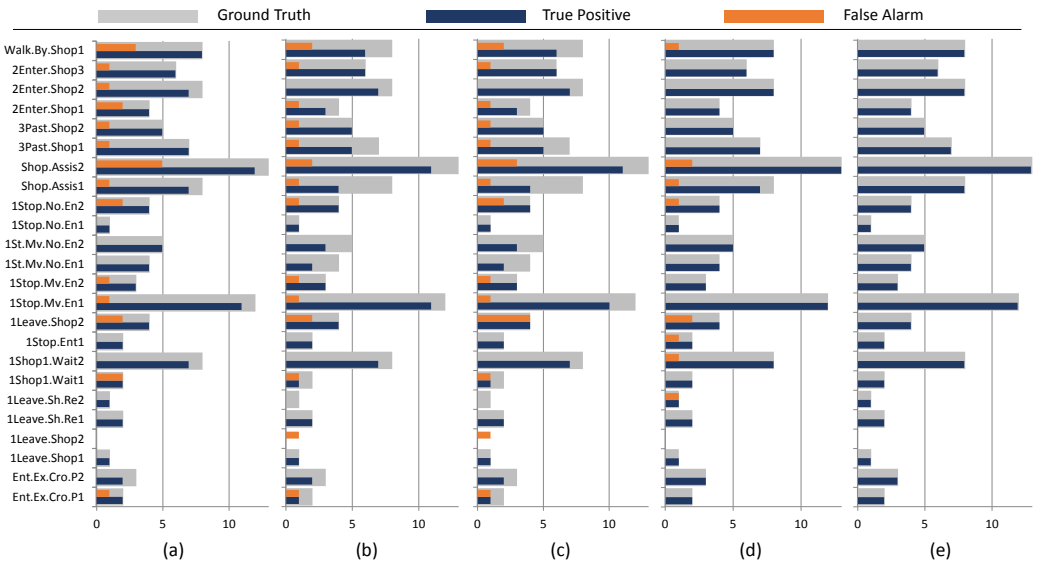


Figure 7: Results of the counterflow experiment on CAVIAR. (a) LPD[1]. (b) LCSS[9]. (c) 2-class classifier. (d) 3-class classifier. (e) 3-class classifier plus SFHM.

slightly better since the trajectory clustering in LCSS can bypass some mix-of-track trajectories. The LPD algorithm is based on a flow field that does not rely on trajectories. Its true positive rate is significantly higher than LCSS and the 2-class baseline approach. On the other hand, it is more sensitive to noise, resulting in a higher false positive rate. From the results we can see that with the 3-way classifier, the detection rate has been improved over the 2-class baseline from 79% to 99%, while the false positive rate has been reduced from 19% to 8%. The performance of the 3-way classifier surpasses LPD and LCSS on both the true positive rate and false alarm rate. When adding the SFHM, the detection rate reaches 100% without any false positives.

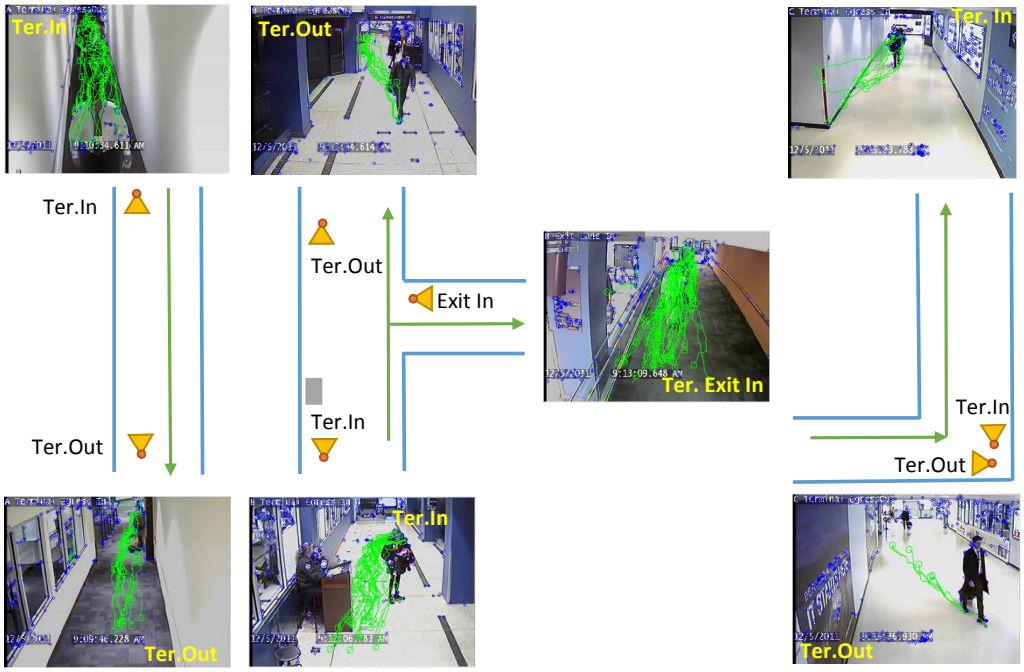
Table 1: Results of the counterflow experiment on the short airport videos. **GT** denotes the number of ground truth counterflows, **TP** the number of true positives and **FA** the number of false alarms.

Video	**Len** min	**GT**	LPD[1]		LCSS[9]		**2-Class**		**3-Class**		**+SFHM**	
			TP	**FA**	**TP**	**FA**	**TP**	**FA**	**TP**	**FA**	**TP**	**FA**
TA Eg.In	40	1	1	2	1	2	1	2	1	0	1	0
TA Eg.Out	32	2	2	6	2	6	2	6	2	1	2	0
TB Eg.In	40	10	8	4	9	4	8	4	10	1	10	0
TB Eg.Out	32	10	10	12	7	10	6	12	10	6	10	0
TC Eg.In	5	0	0	5	0	3	0	3	0	0	0	0
TC Eg.Out	5	2	2	6	0	5	0	5	2	3	2	0
Total	154	25	23	35	19	30	18	32	25	11	25	0

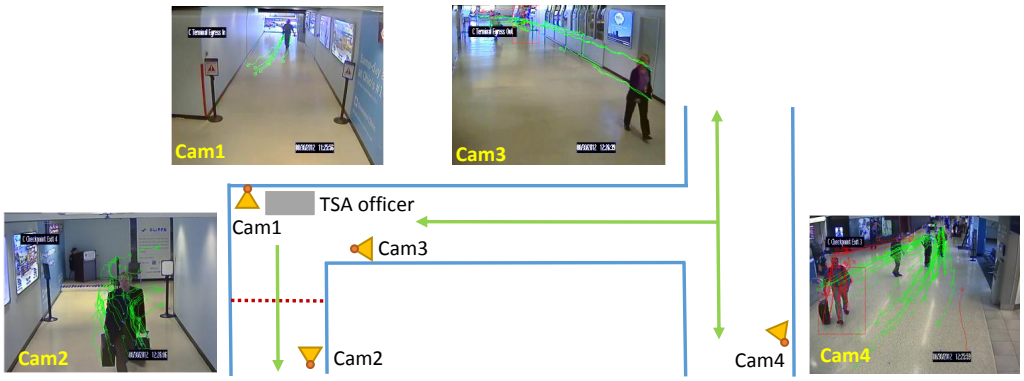
7.2. Shorter Airport Videos

We next tested the counterflow detection algorithm on six video sequences from cameras overlooking exit lanes at an airport. Figure 8(a) illustrates the configuration of the cameras and sample images. The classifier is first trained with a one-minute portion of the video and tested with the rest. The results for the six sequences are collected in Table 1, and several examples of normal flow and counterflow are shown in Figure 9. A bounding box corresponding to each suspicious target is also created.

These videos are all at low resolution (320×240) and contain periodic jitter. It can be seen that with the help of the 3-way classifier, the detection rate improved from 72% to 100% while the false positive rate was reduced from 64% to 30%. The performance of LPD and LCSS reflects the same conclusions from the CAVIAR experiment. Again, the performance of 3-way classifier surpasses both LPD and LCSS. When adding the SFHM, the algorithm achieved a 100% detection rate without any false alarms. No



(a) Floor plan and sample frames for Shorter Airport Videos



(b) Floor plan and sample frames for Long Airport Videos

Figure 8: Floor plans and sample frames from each camera.

(a) (b)

Figure 9: Sample flow classification results. (a) Normal flow. (b) Counterflow is detected and the target is located.

counterflow events were missed using the 3-way classifier.

7.3. Long Airport Videos

Finally, we conducted a long-term experiment at the airport with a camera network consisting of 4 cameras, as illustrated in Figure 8(b).

These cameras are at higher resolution (640×480). However, flicker and artifacts due to compression and illumination problems still make the task challenging, especially for 24/7 continuous processing. We trained the classifier for each camera for 10 minutes. For this experiment, we also evaluated

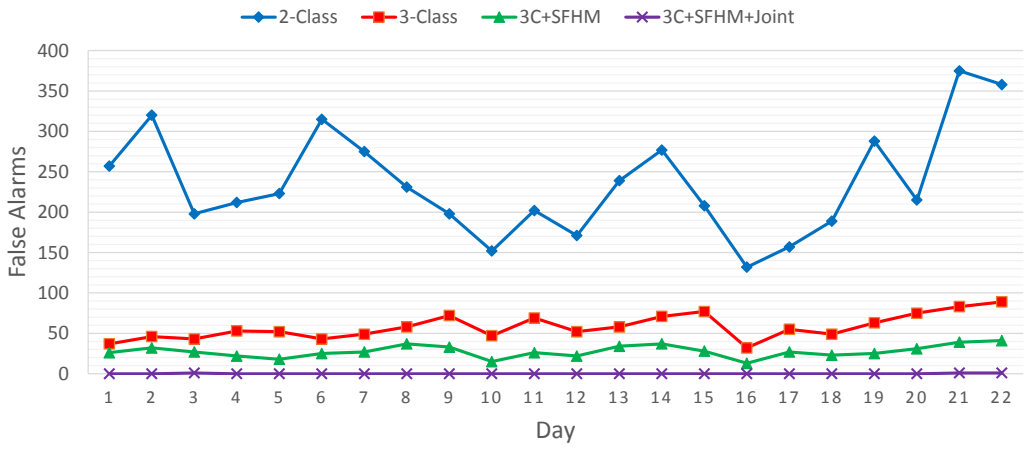


Figure 10: False alarms for the counterflow experiments on the long airport videos. There were 249 ground truth events, 234 of which were detected by the 2-class classifier, and all of which were detected by the other variants of the algorithm.

the method of relating results from multiple cameras mentioned in Section 6. We consider a counterflow event to be detected only if detections are found in both camera 1 and 2 or in both camera 1 and 3 within a 30 second time span (without requiring the detections to be found at exactly the same time). This time span can be set by the expected or statistically longest time for a person to walk from the FOV of camera 2 to the FOV of camera 3. However, a detection found in camera 1 with high confidence (i.e., more than 3 counterflow point tracks are found) is directly considered as a true detection since this viewpoint is most reliable. We processed video from 22 straight days from these 4 cameras in real time. Each day, about 10 counterflow events were generated by airport security officers to test the algorithm. The results are summarized in Figure 10.

From the results we can see that with the help of the 3-way classifier, the

detection rate improved from 93% to 100% while the false positive rate was reduced from 96% to 83%. The addition of the SFHM further reduces the false positive rate to 71%, which is still much too high. By jointly relating the results from three cameras, the false positive rate was reduced to about 1%, which is considered to be operationally acceptable from the perspective of airport security officers (less than 1 false alarm per day). No counterflow events were missed using the 3-way classifier.

7.4. Failure Cases

The proposed algorithms may fail when processing videos with a highly dynamic background, or that contain serious ghosting or compression artifacts. In our experiments, most of the false alarms are due to noise, jitter or mixing point tracks. Since false alarms are easy to assess and discard, the level of performance seems acceptable in practical applications given the lengths of the videos involved. Some videos with false alarms are particularly challenging due to unusual walking patterns of passengers, "acceptable" counterflow caused by security officers, mixing feature tracks, a complex background, and passengers coming from an exit far from the camera.

8. Conclusion

As desired, the algorithms successfully detected all the counterflow occurrences in all the sequences without error. More importantly, the number of false alarms has been significantly reduced by successive refinements to the algorithm.

Both the false alarm rate and true positive rate are improved by the proposed scene-feature-based algorithms. Experimental results show that the

proposed algorithms outperform LPD [1] and LCSS [9] on both a standard dataset and real-world video sequences, suggesting that with the help of scene features, trajectory-based counterflow detection approach can be significantly improved. Future work includes improving the 3-way classifier so that it can be trained without supervision, adding robust broken tracklet re-connection to the algorithm, and identifying security officers and ignoring counterflows caused by them.

9. Acknowledgment

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. Thanks to Edward Hertelendy and Michael Young for supplying the data in Section 7.

References

[1] S. Ali, M. Shah, A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis, in: IEEE Conference on Computer Vision and Pattern Recognition(2007).

[2] J. Alon, S. Sclaroff, G. Kollios, V. Pavlovic, Discovering clusters in motion time-series data, in: IEEE Conference on Computer Vision and Pattern Recognition(2003).

[3] E. Andrade, S. Blunsden, R. Fisher, Modelling Crowd Scenes for Event Detection, in: 18th International Conference on Pattern Recognition(2006).

[4] G. Antonini, J. Thiran, Counting Pedestrians in Video Sequences Using Trajectory Clustering, IEEE Transactions on Circuits and Systems for Video Technology 16 (2006) 1008–1020.

[5] S.T. Birchfield, S.J. Pundlik, Joint tracking of features and edges, in: IEEE Conference on Computer Vision and Pattern Recognition(2008).

[6] J.Y. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, Technical Report, Microprocessor Research Labs, Intel Corporation, 2002.

[7] G. Brostow, R. Cipolla, Unsupervised Bayesian Detection of Independent Motion in Crowds, in: IEEE Conference on Computer Vision and Pattern Recognition(2006).

[8] D. Buzan, S. Sclaroff, G. Kollios, Extraction and clustering of motion trajectories in video, in: Proceedings of the 17th International Conference on Pattern Recognition(2004).

[9] A. Cheriyadat, R. Radke, Detecting Dominant Motions in Dense Crowds, IEEE Journal of Selected Topics in Signal Processing 2 (2008) 568–581.

[10] A.M. Cheriyadat, R.J. Radke, Non-negative matrix factorization of partial track data for motion segmentation, in: IEEE International Conference on Computer Vision(2009).

[11] EC Funded CAVIAR project/IST 2001 37540, Clips from Shopping Center in Portugal, 2004.

[12] B.K. Horn, B.G. Schunck, Determining optical flow, Artificial Intelligence 17 (1981) 185–203.

[13] I. Junejo, O. Javed, M. Shah, Multi feature path modeling for video surveillance, in: 17th International Conference on Pattern Recognition(2004).

[14] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, Proceedings of the 7th International Joint Conference on Artificial Intelligence (1981) 674–679.

[15] L. Marcenaro, G. Vernazza, Image stabilization algorithms for video-surveillance applications, International Conference on Image Processing (2001).

[16] E. Rosten, R. Porter, T. Drummond, Faster and better: a machine learning approach to corner detection., IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 105–19.

[17] P.H. Tu, J. Rittscher, Crowd Segmentation Through Emergent Labeling, Lecture Notes in Computer Science 3247 (2004) 187–198.

[18] Z. Wu, R.J. Radke, Using scene features to improve wide-area video surveillance, in: IEEE International Workshop on Camera Networks and Wide Area Scene Analysis (2012).