Real-time Implementation of Automated Counter-flow Detection in Airport Security Exits

Mohamed A. Elgharib, Venkatesh Saligrama
Boston University

1. Objective

- Airports contain single-direction exit corridors.
- A Major security requirement : **No person can enter the secured area of the airport through the exit corridors (see Fig. 1).**
- A breach of this requirement means : (1) Airport is no longer safe (2) The entire airport could be evacuated.
- **The evacuation process costs millions of dollars.**
- Current solution : 24/7 manual monitoring of all exit corridors.
- Computer Vision Analysis can : (1) reduce the cost of manual monitoring (2) reduce the risk of a security breach.
- **Our Contribution : A real-time implementation of an automated system that analysis camera signals and detects people moving in the counter-flow direction in the exit corridors.**
- This work is carried out in collaboration with DHS, TSA Cleveland Hopkins International Airport and SIEMENS.

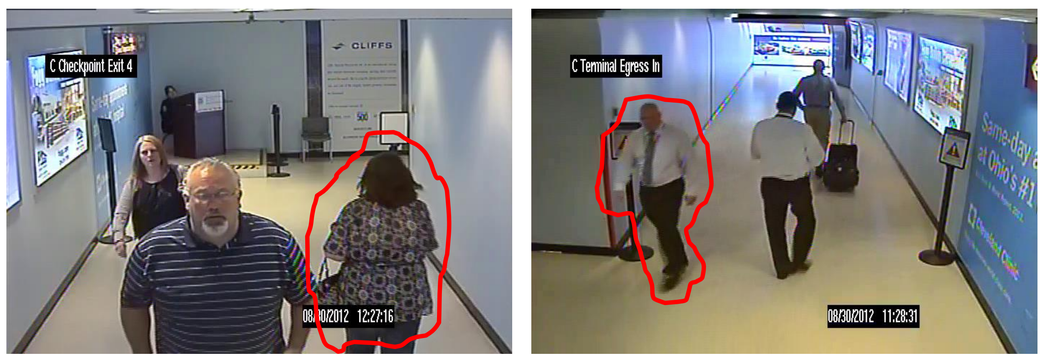

Fig.1 *Examples of people moving in the counter-flow direction (shown in red)*

2. System Overview

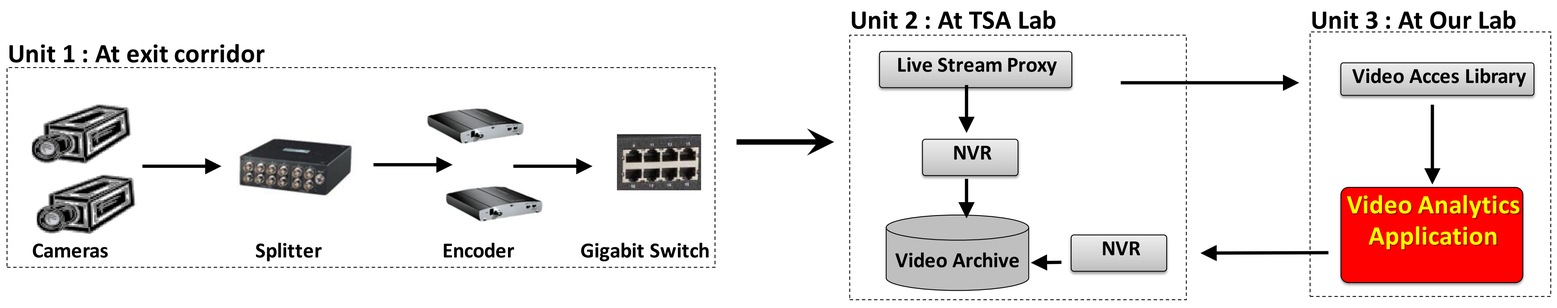

Fig.2 Illustrating the components of the coutner-flow detection system. Unit 1 captures analog video data and converts it to a digital form. Unit 2 feeds the live stream to Unit 3 and stores data for future retrieval through NVR. Unit 3 contains the Video Access library and the Video Analytics Application. The Video Access Library is written in C++ and enables the video data to be accessed by the video analytics application. Our contribution is designing and developing the Video Analytics Application.

3. Video Analytics Algorithm Overview

- Image features points are detected [1] (see green in Fig. 3)
- Feature points trajectories are calculated by estimating correspondence between points at the current and next frames. (see red in Fig. 3)
- Here the KLT feature point trajectory estimator is used [1].
- Only regions undergoing motion are considered .
- **Counter-Flow Inference:**

Counter-flow Displacement Threshold set to 50 pels

$$d_n(\mathbf{i}) = \mathbf{Dir}(F_n(\mathbf{i}) - F_o(\mathbf{i})) > W \qquad (1)$$

Binary indicator with value 1 for counter-flow

Feature point trajectory i at frame n

Feature point trajectory i at its first frame

- $\mathbf{Dir}(.)$ estimates how many pels did the examined trajectory travel in the counter-flow direction.
- Eq. 1 is evaluated every 10 frames for every trajectory.
- The final detection score D_n at frame n is evaluated every 50 frames as follows :

$$D_n = \sum_{k=n-40}^{k=n} \sum_{i \in \text{all trajectories}} d_k(\mathbf{i}) \qquad (2)$$

Fig.3 First row: 2 consecuitve frames. Middle row: Feature points deteciton in green. Last row, from left: Motion mask, feature points trajectories shown in red.

4. Implementation Details

- Three cameras are processed in real-time (see Fig. 4), 30 frames per second per camera, 480x704 pels per frame.
- **An event is flagged as counter-flow if detection occurs in camera 1 followed by a detection in at least one of the remaining cameras.**
- Detection score is set to 280, 10 and 5 for the first, second and third cameras respectively.
- Detected events are saved on to a .txt log file with snapshots.
- Program written in C++ on a Quad Core i7-390 @ 3.2 GB RAM, GeForce GTX 580 GPU.

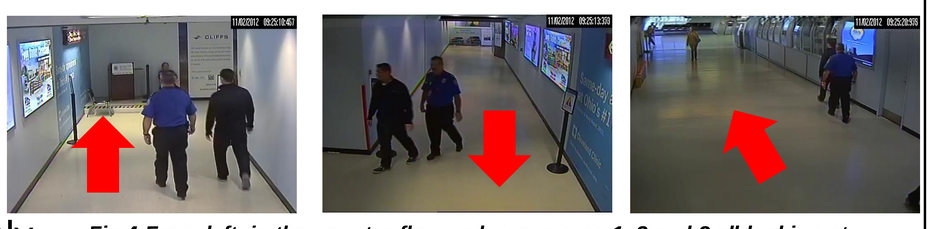

Fig.4 From left, in the counter-flow order: cameras 1, 2 and 3 all looking at the same exit corridor. Counter-flow directions are shown in red.

5. Results

- Program tested in Cleveland Hopkins International Airport for 3 different weeks.
- 10 counter-flow drills were performed by TSA officers every day.
- Program robust to occlusion, waving objects, fast and slow motion & zigzag displacements.

Fig.5 Counter-flow events detected by our technique shown in blue. From top left, in clockwise direction, we have: strong occlusion, fast motion with occlusion, slow motion of 1 person, slow motion of 2 people, an oversized object and a person with a bright coloured jacket.

	Oct. 25-Oct. 29 *~96 hours*	**Nov. 1-Nov. 7** *~168 hours*	**Nov. 29-Nov. 26** *~168 hours*
Correct Detections	100%	100%	100%
Missed Detections	0	0	0
False Detections	1	3	2

Table 1. Results generated by our counter-flow detection program

Current Work, ***Tag and Track*** **:**

- Manual annotation and automated tracking of suspects through out the whole terminal.
- The feature point tracker developed here will be incorporated with spatial information between different cameras.

[1] Jianbo Shi and Tomasi, C., Good Features to Track, in IEEE Conference of Computer Vision and Pattern Recognition, WA, USA 1994