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.

2. System Overview

Unit 1: At exit corridor
- Camera
- Splitter
- Encoder
- Gigabit Switch

Unit 2: At TSA Lab
- Live Stream Proxy
- Video Access Library
- NVR
- Video Analytics Application

Unit 3: At Our Lab
- Video Archive
- Live Stream Proxy
- Video Access Library
- NVR
- 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:
  \[ d_n(i) = \text{Dir} \left( F_n(i) - F_1(i) \right) \geq W \]  
  (1)

  \[ D_n = \sum_{k=1}^{40} \sum_{i \text{all trajectories}} d_n(i) \]  
  (2)

- 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:

- 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 17-390 @ 3.2 GB RAM, GeForce GTX 580 GPU.

4. Implementation Details

- 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.

5. Results

- 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.

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Correct Detections</th>
<th>Missed Detections</th>
<th>False Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct. 25-Oct. 29</td>
<td>100%</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Nov. 1-Nov. 7</td>
<td>100%</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Nov. 29-Nov. 26</td>
<td>100%</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>