Detecting Single-Actor Events in Video Streams for TRECVid 2008

Andreas Stergiou, Aristodemos Pnevmatikakis, Lazaros Polymenakos
Athens Information Technology
0.8km Markopoulo Av., Peania 19002, Greece
{aste,apne,lcp}@ait.edu.gr

Nikos Katsarakis
Athens Information Technology and
Aalborg University, CTiF
nkat@ait.edu.gr

October 22, 2008

Abstract

This paper presents the systems and results for the Event Detection task of the TRECVid 2008 evaluation campaign. The kind of events addressed are single-actor, without requiring detailed tracking of body parts; rather the body as a whole is considered and the interaction between people is not addressed. The events thus considered are the OpposingFlow, PersonRuns and ElevatorNoEntry. For the first two, motion vectors of areas with large frame-by-frame activity are analyzed. Groups of motion vectors pointing opposite at key door-frames indicate opposing flow events, while other groups with large magnitudes that persist across frames indicate running. For the ElevatorNoEntry event, the state of the elevator doors and the number of people present in the camera field of view are tracked to decide if an elevator arrives and none of the people present enter it. People and elevator doors are tracked using blob-based tracking on top of adaptive foreground segmentation.

1 Introduction

While tracking and recognition can provide a lower-level understanding of a video sequence by answering the 'Where?' and 'Who?' questions, a higher level of understanding is sought in order to answer the 'What?' question. This third question relates to what is happening in a video stream, the focus being the detection of events of interest. The Video Retrieval Evaluation conference of the Text REtrieval Conference series (TRECVid 2008) [1] initiated an Event Detection evaluation campaign [2] to assess performance of video event detection

systems in a variety of events occurring in the Gatwick airport. 17 events have been defined [3]. These can be categorized based on the required level of body modeling as:

- Whole body: The whole human body as a blob suffices to detect the event.
- Body parts: It is essential to model the various body parts, usually hands, heads or torsos.

A second categorization involves the requirement to understand interaction with other objects or people.

- None: The observation of the single actor suffices.
- People: More than one person are involved. Their interaction needs to be modeled.
- Fixed object: The person interacts with a fixed (usually large) object whose position and visual characteristics can be assumed known by the system.
- Arbitrary object: The person interacts with a (usually small) object that he or she at some times carries and that is not always visible. The object might appear anywhere in a frame and its visual characteristics (color, exact shape and size) are not known.

The submitted AIT systems for event detection address events that require only whole body modeling and either no interaction, or interaction with fixed objects. Three events are chosen: the OpposingFlow and PersonRuns events involve a single person without any interaction and the ElevatorNoEntry involve people interacting with two elevators. This interaction is to ignore the elevator, step out from it or step into it. The camera setup is such that for the first two events, the scene is mostly very crowded, making 2D person tracking very difficult. Since the cameras view mostly disjoint spaces, 3D tracking is not possible. As these two events imply motion, their detection is based on the motion vectors extracted from areas of the frames with large frame-by-frame activity, as detailed in Section 2. The ElevatorNoEntry events are detected on a near-filed camera view resulting into sparsely occupied frames with little activity. It is relatively straightforward to build a blob-based tracker for such a setup. The detection of the ElevatorNoEntry events based on a blob tracker is detailed in Section 3, followed by the experiments and results in Section 4 and the conclusions in Section 5.

2 Event detection using motion vectors: Running and opposing flow

2.1 Motion vectors

Events involving the way people move, where their interaction with other people or objects is irrelevant can be detected using motion information. We estimate motion using block matching and logarithmic search [4]. The block size is 8×8 pixels, but a motion vector is calculated for a block only is there is activity in that part of the frame and there is significant texture present. We require activity since its absence implies no motion, and contrary to estimating motion vectors for compression, in this case we are not interested in residual error minimization [4], but on finding how image blocks move. Activity is measured as the mean of the frame-by-frame difference in the specific block location. We require the block to have texture, because in textureless areas there is ambiguity in the motion vector estimation. Textureless blocks are those whose standard deviation is small. We estimate motion vectors both using the previous and the next frame. The final motion vector is the mean of the two, if they do not differ too much. In the latter case, the motion vector of either the forward or the backward prediction is chosen, based on which prediction gave the smallest matching error.

2.2 Person runs

We search for PersonRuns events in all cameras and the complete frame. Running is indicated by some motion vectors with large magnitude. Such large vectors should be present for a few frames, as the person runs for some time. What magnitude can be considered large is not straightforward though. Pixels do not correspond to the same distance across the frame. Pixels depicting people or objects closer to the camera correspond to smaller distance than others depicting far-away people or objects. As all the cameras are positioned higher than head level, people closer to the bottom edge of the frame are closer to the camera. Hence the motion vector magnitude is weighted proportional to the distance from the bottom frame edge. The proportionality constant is estimated using the development set.

Evidence of running is collected as follows: Every weighted motion vector magnitude exceeding a threshold increments the evidence for running for the particular frame. The decision for running is taken with a delay. The evidence for running is first smoothed by a running average filter of 1 second long window. Each time the smoothed evidence for running exceeds a threshold, the local peak in this temporal neighborhood is found, together with the times the smoothed evidence for running begun to grow and ceased again. This temporal window is marked to contain running if the sum of the smoothed evidences for running in it exceeds a threshold.

An example of a peak in the smoothed evidence for running and some of the associated frames with the motion vectors superimposed on them is depicted in

Figure 1. The particular peak is very strong, but also results to an obvious false positive, since the runner is the airport taxi, not a person.

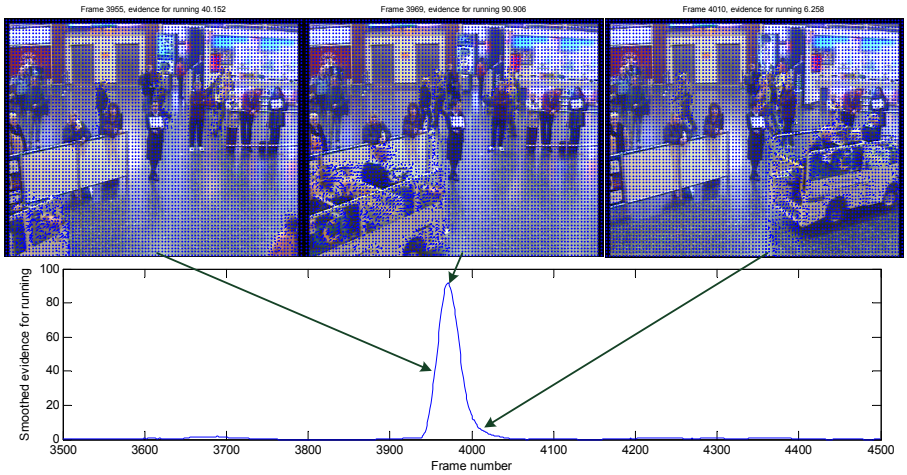

Figure 1: A peak in the smoothed evidence for running and some of the associated frames with the motion vectors superimposed on them.

2.3 Opposing flow

Detecting the OpposingFlow event utilizes the motion vectors in the two doorways in camera 1, and the area immediately below them. The algorithm collects evidence supporting and contradicting the existence of the event. Event supporting evidence ev_{oppose} is accumulated from motion vectors in any of the doorways pointing to the left of the image, as long as their magnitude is larger than unity. There are two types of contradicting evidence:

- Obeying the flow in the doorways ev_{obey}. This evidence is accumulated from motion vectors not pointing towards the right of the image.
- Opposing the flow below the doorways ev_{below}. It is usually the case that people, especially airport personnel, move towards the left, but not through the doorways. Their heads can occlude the doorways, adding false evidence supporting the event. Their torsos on the other hand will lie immediately below the doorways, giving similar motion vectors that point to the left. These are collected as evidence contradicting the event, to balance the heads giving false supporting evidence.

Hence the evidence suggesting the event is given by:

$$ev_{opposingflow} = ev_{oppose} - (ev_{obey} + ev_{below}) \quad (1)$$

The existence and the duration of the event are deduced using the same peak detection mechanism of the PersonRuns event, only now the smoothing window is 0.5 seconds.

3 Event detection using state tracking: Elevators

For the detection of the ElevatorNoEntry events, people and elevator open doors are tracked using blob-based tracking [5] on top of adaptive foreground segmentation [6]. The background image is adapted to account for lighting changes that occur during the 2 hour long recordings. The number of body blobs is stored. Door open/close events are signaled when blobs of significant height and width appear/disappear at the known positions of the two elevator doors. Such blobs are very unlikely to be caused by a person. Hence the state of the system comprises of a three element vector. The first element is integer, corresponding to the number of people present; the other two are binary, indicating if any of the two elevator doors are open. The event is detected when any of the doors opens with a given number of people present, to close later with the same number of people still being there.

Examples of the states during elevator door openings and closings are given in Figure 2. The states corresponding to the upper images lead to an ElevatorNoEntry event, as the person is waiting by the right elevator but the left one opens. The number of people remains the same before, during and after the elevator doors open and close. The states corresponding to the lower images do not indicate an event, as people clearly enter the right elevator. This is indicated by the decrease of the number of people after the doors close.

4 Experiments

All the thresholds for the event detection algorithms have been estimated using the development data and the associated annotations [7]. The evaluation data are 50 hours long for the PersonRuns events (all five cameras are involved) and 10 hours long for the other two (OpposingFlow is defined only for camera 1 and ElevatorNoEntry only for camera 4).

Note that the submission for the ElevatorNoEntry events has been wrong: The tracker operates every five frames, but the resulting frame numbers have not been casted back to the original frame rate.

5 Conclusions

Detecting events related to single person, without interacting with objects at unknown positions has been proven very difficult in the crowded Gatwick airport. Other events are even more difficult to detect.

In cases where the scene is not crowded and tracking is feasible, event detection is more reliable, especially if no modeling of body parts is needed, and any interaction of the people to be detected is with objects at known locations and of known appearance, like elevators or ATMs.

Acknowledgment

This work has been partly sponsored by the European Union, under the FP7 Specific Targeted Research Project HERMES (Cognitive Care and Guidance for Active Aging).

References

[1] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR'06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[2] TRECVid 2008 evaluation for event detection. www.nist.gov/speech/tests/trecvid/2008/.

[3] TRECVid 2008 event annotation guidelines. www.nist.gov/speech/tests/trecvid/2008/doc/TRECVid08_Guidelines_v1.6.pdf.

[4] A. Murat Tekalp. *Digital video processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[5] Ghassan Karame, Andreas Stergiou, Nikos Katsarakis, Panos Papageorgiou, and Aristodemos Pnevmatikakis. 2D and 3D face localization for complex scenes. In *IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS 2007)*, London, UK, September 2007.

[6] Aristodemos Pnevmatikakis and Lazaros Polymenakos. Robust estimation of background for fixed cameras. In *15th International Conference on Computing (CIC '06)*, pages 37–42, Mexico City, Mexico, November 2006. IEEE Computer Society.

[7] Stéphane Ayache and Georges Quénot. Video corpus annotation using active learning. In *30h European Conference on Information Retrieval (ECIR'08)*, pages 187–198, March 2008.

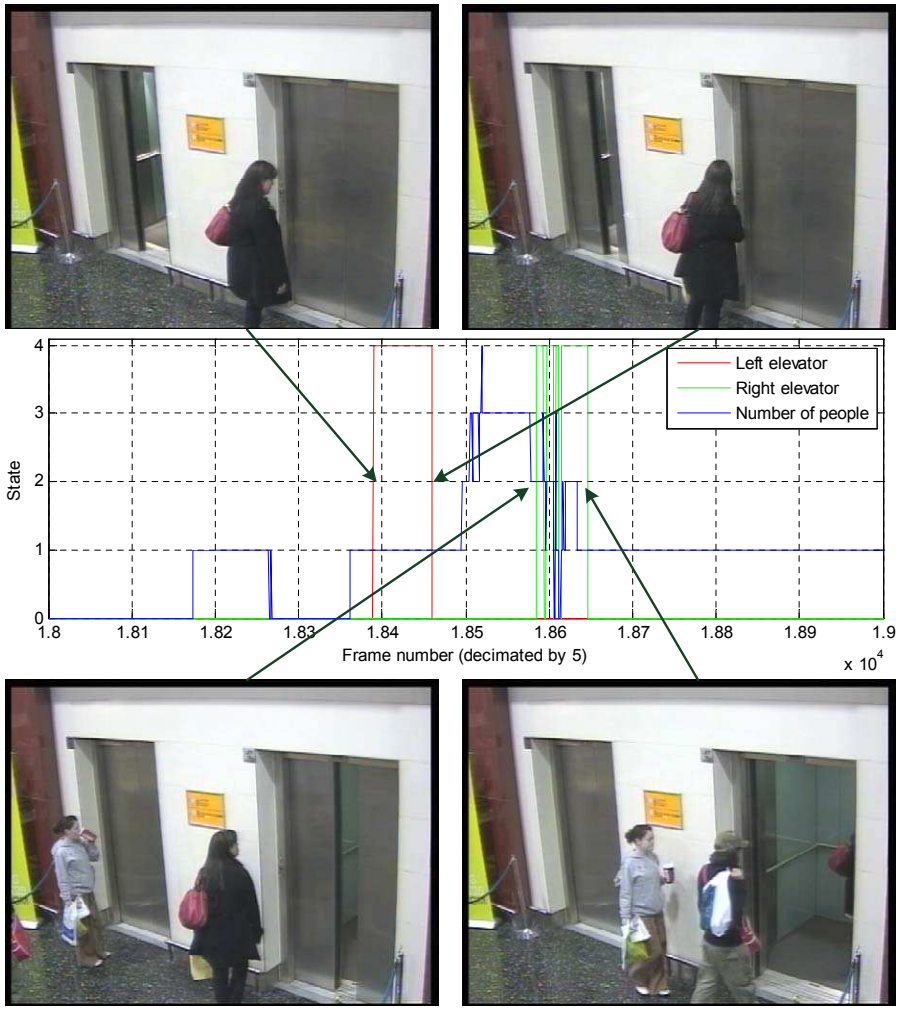

Figure 2: Examples of the states during elevator door openings and closings and the associated frames.