

A system for airport surveillance: detection of people running, abandoned objects and pointing gestures

Samuel Foucher, Marc Lalonde, Langis Gagnon¹

^aR&D Dept., CRIM, 405 Ogilvy Avenue, Suite 101, Montreal, QC, Canada, H3N 1M3

ABSTRACT

The proposed system is focusing on the detection of three events in airport videos: a person running, a person putting down an object and a person pointing with his/her hand. The system was part of the NIST-TRECVID 2010 campaign, the training dataset consists in 100 hours of video from the Gatwick airport from five different cameras. For the detection of a person running, a non-parametric approach was adopted where statistics about tracked object velocities were accumulated over a long period of time using a Gaussian kernel. Outliers were then detected with the help of a kind of t-student test taking into account the local statistics and the number of observations. For the detection of “object put” events, we follow a dual background segmentation approach where the difference in response between a short term and a long term background model (Mixture of Gaussians) triggers alerts. False alerts are excluded based on a simple modeling of the camera geometry in order to reject objects that are too large or too small given their positions in the image. The detection of pointing gesture events is based on the grouping of significant spatio-temporal corners (Harris) in a 3x3x3 cell called compound features as proposed recently by Andrew Gilbert *et al.* [10]. A hierarchical codebook is then derived from the training set based on a data mining algorithm looking for frequent items (called transactions). The algorithm was modified in order to deal with the large number of potential transactions (several millions) during the training step.

Keywords: video surveillance, action recognition, event detection

1. INTRODUCTION

The automatic detection of highly semantic events from security cameras in a public places remains challenging because of the high complexity of the data: many people can be seen in different contexts, such as isolated vs. in group, carrying objects, etc. The problem of understanding the actions and behavior of these people is very difficult and it gets even worse if the detection system is equipped with low-cost analog cameras. Over the last three years, this scenario has been proposed during the NIST-TRECVID campaign [1], where images from five cameras inside an airport terminal are supplied to an automatic surveillance system, whose task is to locate specific events in the video streams such as people putting down objects (typically luggage), meeting/hugging other people, using a cell phone, etc. Most systems evaluated at the end of the campaign show a very high level of missed events (miss rates above 95%) while trying to limit the number of false events per hour. This paper reports on some results obtained by the our team with the TRECVID video data on three detection problems: 1) detection of people running (*PersonRuns*); 2) detection of object put on the floor (*ObjectPut*) and 3) detection of the action of someone pointing (*Pointing*). For the people running detection, we propose a non-parametric approach where the statistics of the observed object velocity field are learned from the video data from which we detect velocity outliers. The detection of ObjectPut is based on a dual-foreground algorithm where the difference in response is analyzed. Finally, a recent technique for action recognition is implemented based on the mining of a dense and overcomplete set of low-level spatio-temporal features [10].

The system was tested on a 144 hours video corpus as part of the TRECVID’2010 competition [2].

2. TRACKING SYSTEM

The tracking system is a basic blob tracker described in [3]. It is composed of two main components: (1) background modeling, and (2) track management. First, a foreground/background segmentation is performed and then, objects are tracked with a combination of blob matching and particle filtering techniques. Blobs resulting from the

¹ langis.gagnon@crim.ca; phone 514-840-1235; fax 514-840-1244; crim.ca

background/foreground segmentation in successive frames are matched based on the similarity between their shape moments. In case the blob matching fails, we apply a simple particle filter tracker (bootstrap filter) based on the color histogram observed on the last object occurrence. Foreground/background segmentation is based on a SOM approach [4].

3. SCENE MODELING AND UNDERSTANDING

3.1 Pedestrian occurrence

The goal here is to produce processing masks for the various detection tasks as well as build simple camera geometry models in order to reduce false alarms. Pedestrian detections were performed on the entire development set (100 hours) using the Dalal and Triggs detector [5]. For each position within the scene, we estimated also the average pedestrian height. The height measurements are also exploited below for the geometric modeling of each camera view. A pedestrian probability map is also computed as shown in Figure 1. Those probability maps, once thresholded, will define the processing masks for the *PersonRuns* event detection.

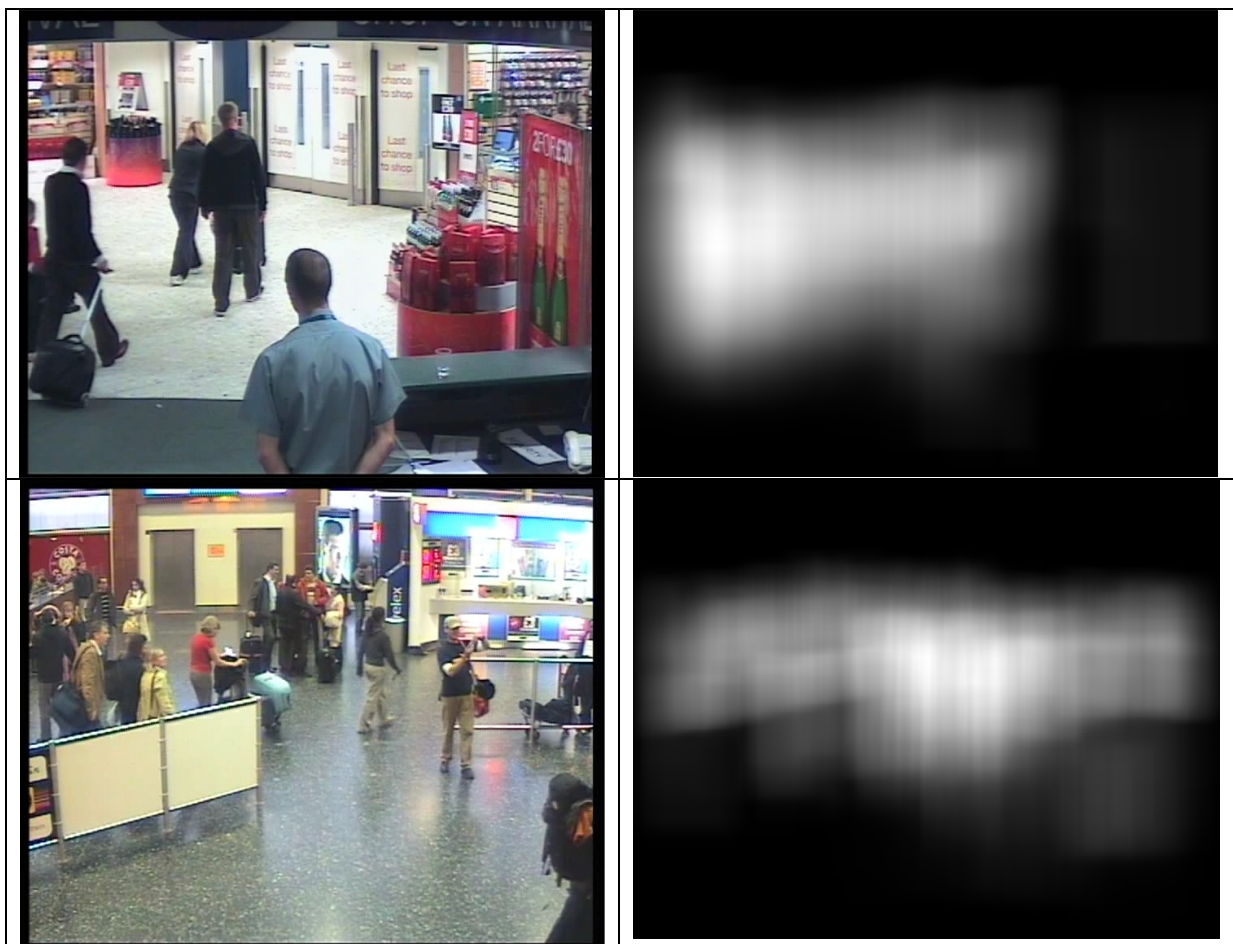


Figure 1. Top row: a frame from Camera 1 and the corresponding pedestrian probability map (white= high number of detection occurrences); bottom row: same for Camera 3.

3.2 Camera geometry

A similar approach was used by the SFU team at TRECVID 2009 [7]. Assuming a simple projective geometry, a camera parallel to the ground plane and objects only on the ground plane, we can exploit the following relationship between the real world object height h and the observed image height Δy [6]

$$h = h_c \frac{\Delta y}{y_b - y_0} \quad (1)$$

where y_0 is the row position of the horizon line and y_b is the bottom image coordinate for the object. Therefore a simple automatic camera calibration can be performed by regression from all the pedestrian measurements given an average person height h_p and standard deviation σ_p (here we chose $h_p = 1.8m$ and $\sigma_p = 0.15m$). The likelihood of the estimated height h is assumed to be Gaussian

$$p(h|h_p, \sigma_p) = \frac{1}{\sqrt{2\pi\sigma_p^2}} e^{-\frac{(h-h_p)^2}{2\sigma_p^2}} \quad (2)$$

Figure 2 gives an example of the object height model derivation for one of the camera. A mask for the ground plane is then derived from the convex hull of the pedestrian bottom positions that are in good agreement with the height model (2). This mask will be used to validate objects put on the floor in Section 5.

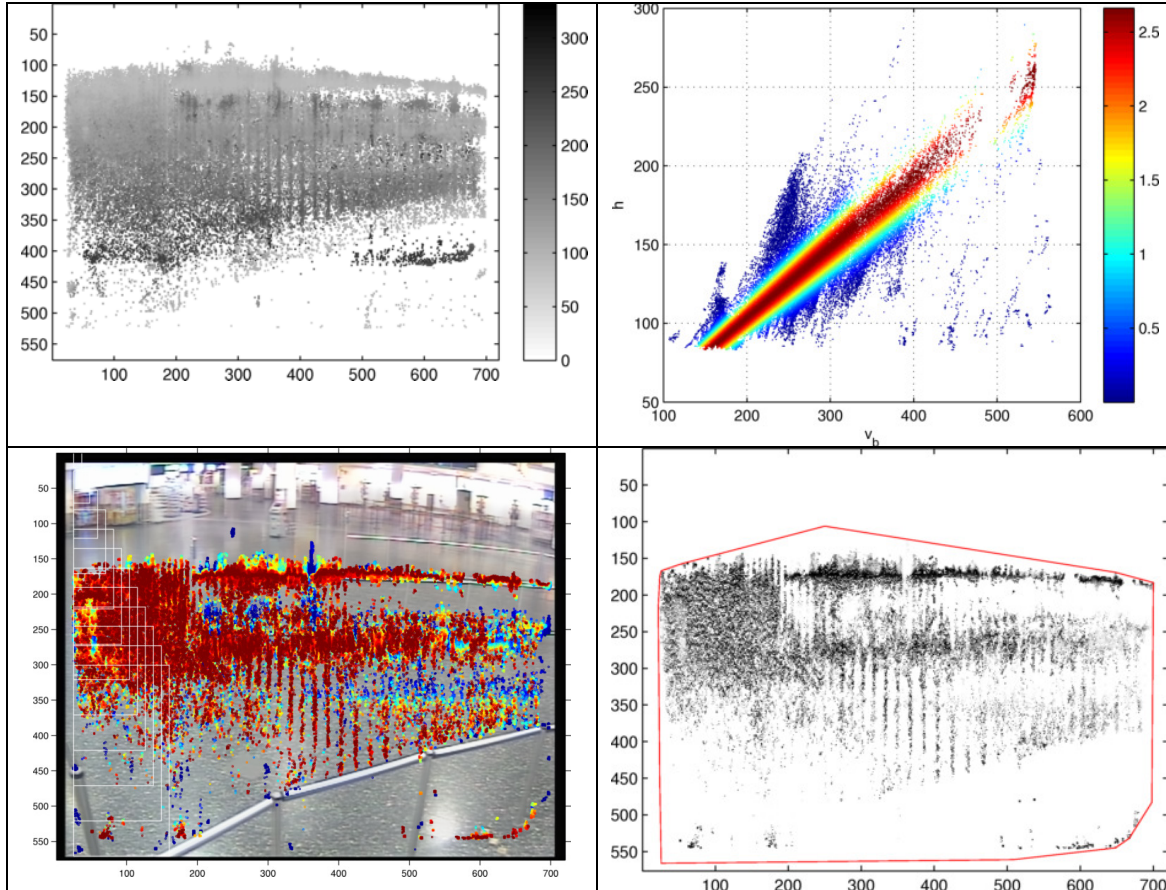


Figure 2. From left to right and top to bottom: average pedestrian height map (dark=tall); object height distribution function of the object vertical position with color function of likelihood (2), most likely object positions with white boxes on the left showing predicted heights by the model; convex hull (in red) of the most likely area for the *ObjectPut* detection.

3.3 Object velocity

Our object tracker was run on the training set. From the object tracks, we derive velocity measurements in pixels/frames. We are considering only the trajectories that are “clean” enough with a minimum number of observations and close to a linear trajectory. At each image location \mathbf{p} , we estimated the velocity moments of order r from all the observed velocities

$$M_r(\mathbf{p}) = \frac{1}{C} \sum_{i=1}^n v_i^r K\left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{s}\right), \text{ with } C = \sum_{i=1}^n K\left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{s}\right) \quad (3)$$

where $K()$ is a Gaussian spatial kernel with a spatial width s . Only moments of order 1 (mean) and 2 are estimated. Those statistics will be used for the *PersonRuns* event detection explained below. In Figure 3, we show the resulting statistics for camera 5 with a kernel size value s set to 5 pixels.

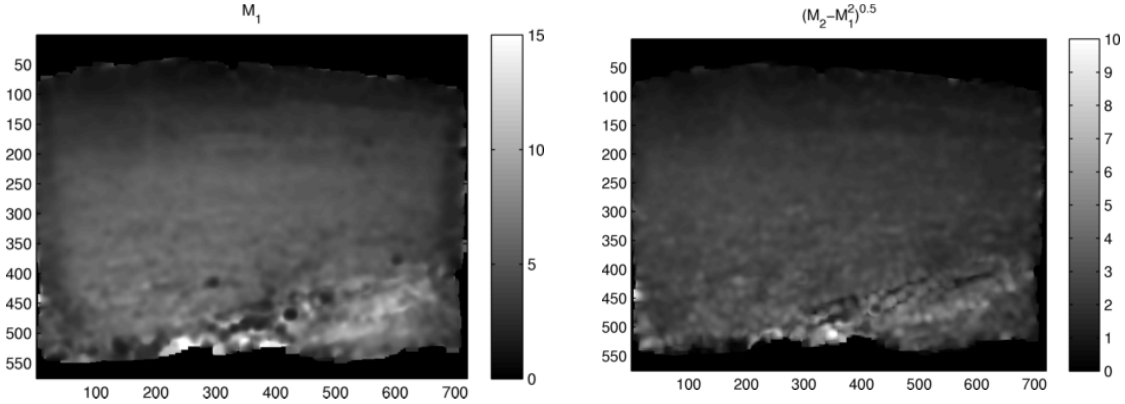


Figure 3. Average velocity map for camera 5 (left) and standard deviation (right).

4. PERSON RUNNING

Our tracking system is producing trajectories (tracks) on various detected foreground objects within the scene. For each position in the track we record the object velocity, size and compute the final distance (distance between the track starting point and ending point) as well as the total distance (total distance travelled by the object). The *PersonRuns* events were detected by assessing by how much the current velocity diverges from the learned statistics with a kind of a one-sided t-student test. Let’s say that we observe an object with velocity v , we then compute the Velocity Outlier (*VO*) score at the confidence level α based on the learned scene statistics

$$VO_{1-\alpha} = \frac{(v - M_1)}{2t_{\alpha, n-1} \sqrt{M_2 - M_1^2} / \sqrt{n}} \quad (4)$$

where $t_{\alpha, n-1}$ is the one sided t-student distribution with $n-1$ degrees of freedom where n is the number of observations involved in the computation of M_1 and M_2 . In case of a person running, $VO_{1-\alpha}$ should capture the deviation from the mean velocity and take values over 0.5. A confidence level is then computed for each observed track function of the average *VO* score and the track quality:

$$Conf_{PR} = \min(E\{VO_{\alpha, n-1}\}, 1) \times \frac{\text{Final Distance}}{\text{Total Distance}} \quad (5)$$

where $E\{VO_{\alpha, n-1}\}$ is the average outlier score observed over the duration of the track. The ratio of the track “Final Distance” over “Total Distance” penalizes tracks that are too noisy or that are significantly different from a linear trajectory. In order to further reduce the number of false alarms, only events within the learned pedestrian mask (see section 1) were processed.

5. OBJECT PUT

The *Object Put* event detection is a more general problem than “abandoned object” as it includes also all objects put down on the floor but not necessarily abandoned. Most of the time, the individual to whom the object belongs to, will generally stay near the object for a significant period of time. Also, the object will generally be occluded by passing pedestrians. The approach proposed is based on a very simple dual background model approach described in [8]. Both the long term and short-term background models were Mixture of Gaussians (MoG) [9]. The learning rate for the short term was fixed to $1/20$ and the long term rate to $1/600$. Both models use a 5 modes mixture model. The difference image between the foreground images coming from the two models is then accumulated over time. An example is shown in Figure 4 where we can observe the bag on the floor getting darker (i.e. more likely) on the alert images.

In order to reduce false alarms, the object height is validated with the camera geometric model described in section 3:

1. only alerts with a height between 10% and 60% of the expected pedestrian height are considered.
2. only the alerts within the ground mask estimated in section 3.2 are considered.

The confidence level for the event is derived from the average accumulated value within the event ROI on the cumulative difference image. This fast and simple approach gave good results in terms of false alarms.



Figure 4: Some frames for an *ObjectPut* event (left column); foreground difference image (center) and alert images (left)

6. ACTION RECOGNITION

Action recognition within an unconstrained environment is a challenging problem. One issue is that many different actions occur at once in a typical airport scene. For the *Pointing* event, we implemented a recent approach based on the learning of compound features proposed recently by Gilbert *et al.* [7][8]. This is a data mining approach where meaningful configurations of spatio-temporal salient points are mined from the video dataset. A dense and overcomplete set of simple spatio-temporal Harris corners at various scales, called compound features, are encoded to form transactions (or itemsets). The goal of the data mining algorithm is to extract frequent transactions from this dense set of simple features that may be related to the ongoing action. Transaction rules are then formed from frequent itemsets, a confidence level is associated to each rule derived from the observed transaction occurrences in both the positive and negative transaction sets. The pointing event is detected if enough rules are firing at a given point of time. The following steps are involved for the first scale:

1. build a dense and overcomplete set of Harris corners at various spatial scale and in the temporal domain.
2. group corners within a 3x3x3 neighborhood to form compound features.
3. compound features are encoded using information about cell position, scale and corner type to form transactions (or itemsets).
4. a data mining algorithm (*APriori* algorithm) is applied in order to extract frequent itemsets.
5. transaction rules and associated confidence levels are derived from the frequent itemsets.

The above steps are then repeated for a larger neighborhood but using only the filtered compound features from the previous scale. The training stage requires a region of interest centered on the action. This is not an easy task in unconstrained videos as the subjects may have different poses and are sometimes walking. We chose to select the area starting at the shoulder and covering the arm extent at the apex of the action. Only the frames from the start of the action to the apex (i.e. arm fully extended) were included. The training was very limited due to the lack of time, only the pointing events in the first video of camera 1 in the development set were used (about 20 events). The positive compound features were taken within the region of interest. Negative samples were taken on the same frames but outside the region of interest.

For the *APriori* data mining algorithm we used the implementation of Christian Borgelt [12]. From our limited training sets, we extract about 300,000 transactions at level 1. When *APriori* is applied, we get about 3.0 million frequent itemsets. Rules are then derived from the frequent itemsets. There are two phenomena that can be observed:

- 1) The number of rules increases exponentially with the size of the itemsets. For instance, we get only 256 rules for 1-itemset but 90 thousands rules for 2-itemsets, 500 thousands for 3-itemsets, etc.
- 2) The more items in the transaction, the less frequent it is and therefore the less chance to find an intersection between the negative and positive sets. One consequence is that a rare positive itemset that is not observed in the negative set will be assigned a 100% confidence level (this is the case for 75% of the 2-itemset rules). One possible way to limit this problem is to impose a minimal support value (i.e. number of observations) to a frequent itemset.

One issue we need to tackle is the large number of transactions generated during the training step (over 1 million transactions were generated here). Another issue is how to take a reliable decision on the presence of *Pointing* events in a scene where many other actions are taking place (e.g. people walking). Figure 5 shows the detection results when transaction rules are applied. We can see that compound features corresponding to high-confidence level rules (in red) concentrate around the pointing action. The corresponding probability map is shown but because the compound features are rare and sparsely distributed the resulting map is not very informative. Looking only at performance on the development set for camera 1, we get a miss rate of 78% and a false alarm rate of 580 events/hour which is too high.

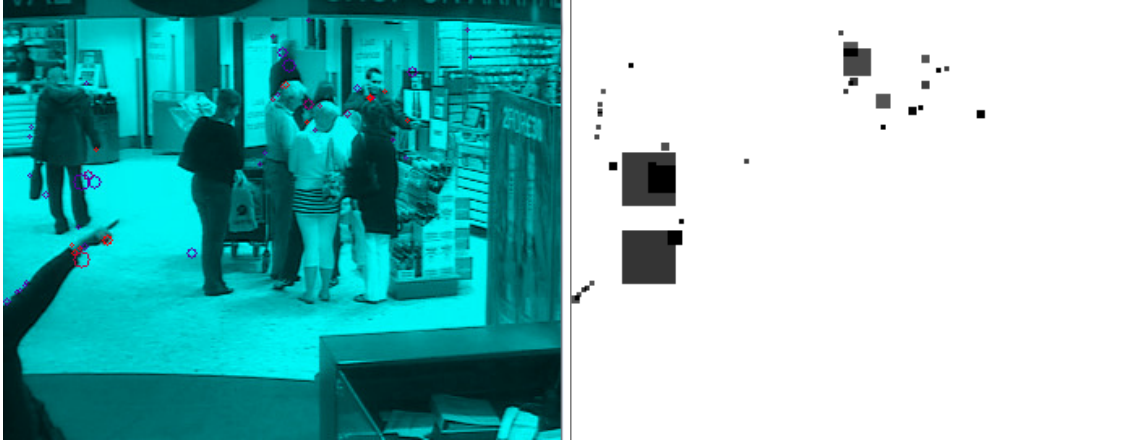


Figure 5: Example of a Pointing frame with learned compound features shown as circles (left) and corresponding probability map (right) where dark means high probability.

7. RESULTS

The system was tested on a 144 hours video corpus as part of the TRECVID'2010 campaign [2]. Videos were acquired by 5 different indoors cameras at the Gatwick Airport (courtesy of the UK Home Office) and compressed in MPEG-2 format. The corpus is split between 100 hours of development data (10 hours x 2 hours/day x 5 cameras) and 44 hours of test data. The development set is annotated in terms of time positions of events only (the spatial location of the events is not provided). The results are evaluated in terms of miss event rates, false alarms rates per hour (RFA) and Normalized Detection Cost Rates (NDCR) using the F4DE toolkit [1]. Note that this framework evaluate detected events only in terms of temporal overlap with the ground truth and does not take into account its spatial location. Overall detection results (see Table 1 and 2) present too many false alarms especially for *PersonRuns* and *Pointing*.

Table 1. Actual Miss rate and False Alarm rate for each event

Event	Person Runs	Object Put	Pointing
Act. Miss	0.196	0.839	0.964
Act. RFA (in Events/Hour)	2110	232	440
Act. DCR	10.745	1.999	3.166

Table 2. Minimum Miss rate and False Alarm rate for each event

Event	Person Runs	Object Put	Pointing
Min Miss	0.944	0.955	0.988
Min RFA (in Events/Hour)	68	0.394	228
Min DCR	1.285	0.997	2.127

8. CONCLUSIONS

The proposed *PersonRuns* detector is based on a non-parameteric estimation of the object velocity field for each camera learned on the development set. The method is fairly simple with a processing time close to real time. However, results show a large number of false alarms mainly because our tracker is too noisy and track fragmented blobs.

The *ObjectPut* detector despite its simplicity gave surprisingly good results and was one of the top algorithms in the TRECVID'2010 campaign. As most of the false alarms are stationary people (bystanders), the results could be improved when combined with a pedestrian detection.

The action recognition algorithm needs to be improved, in particular the training stage which was too limited. This detection would greatly benefit also from a person detector in order to limit the number of false alarms and improve the processing time.

ACKNOWLEDGMENTS

This work has been supported by MDEIE of the "Gouvernement du Québec". The authors wish to thank the RQCHP (Compute Canada) for giving access and support to their high performance computing resources.

REFERENCES

- [1] Smeaton, A. F., Over, P., and Kraaij, W., "Evaluation campaigns and TRECVID". In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval (Santa Barbara, California, USA, October 26 - 27). MIR '06. ACM Press, New York, NY, 2006.
- [2] TRECVID 2010 Evaluation for Surveillance Detection, <http://www.itl.nist.gov/iad/mig/tests/trecvid/2010/>
- [3] Foucher, S., Lalonde, M. and Gagnon, L., "Automatic Scene Modeling for Improving Object Classification", Proc. SPIE Defense and Security Symposium: Visual Information Processing 2010, Proc. SPIE 7701, 770104 (2010); doi:10.1117/12.850273.
- [4] Maddalena, L. and Petrosino, A., "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", IEEE Trans. on Image Processing, Vol. 17, No. 7, 2008, pp. 1168-1177.
- [5] Dalal, N. and Triggs, B., "Histograms of Oriented Gradients for Human Detection", CVPR 2005, vol. 1, pp. 886 - 893.
- [6] Hoiem, D., Efros, A. A. and Hebert, M. "Putting Objects in Perspective", CVPR 2006, pp. 2137 - 2144.
- [7] Yang, W., Lan, T., and Mori, G., "SFU at TRECVID 2009: Event Detection", School of Computer Science, Simon Fraser University.
- [8] Porikli, F., Ivanov, Y. and Haga, T., "Robust Abandoned Object Detection Using Dual Foregrounds", EURASIP Journal on Advances in Signal Processing, vol. 2008, Jan. 2008.
- [9] KaewTraKulPong P. and Bowden, R., "An improved adaptive background mixture model for real-time tracking with shadow detection", Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems, 2001.
- [10] Gilbert, A., Illingworth, J. and Bowden, R., "Action Recognition using Mined Hierarchical Compound Features" IEEE Trans Pattern Analysis and Machine Learning. 2011, vol. 33, no. 5, pp. 883 - 897.
- [11] Gilbert, A., Illingworth, J. and Bowden, R., "Fast Realistic Multi-Action Recognition using Mined Dense Spatio-temporal Features", In Proc. Int. Conference Computer Vision (ICCV09), Kyoto, Japan, pp. 925 - 931.
- [12] Borgelt, C., "Recursion Pruning for the Apriori Algorithm", 2nd Workshop of Frequent Item Set Mining Implementations (FIMI 2004, Brighton, UK).