Crowd Behavior Recognition for Video Surveillance

Shobhit Saxena[1] and François Brémond[2], Monnique Thonnat[2], Ruihua Ma[2]

[1] Department of Computer Sc. and Engg, Indian Institute of Technology,
Hauz Khas, New Delhi, India
csd02407@cse.iitd.ernet.in
[2] INRIA Sophia Antipolis, Pulsar Team
2004 route des Lucioles - BP 93, 06902 Sophia Antipolis Cedex, France
firstname.surname@sophia.inria.fr
http://www-sop.inria.fr/pulsar

Abstract. Crowd behavior recognition is becoming an important research topic in video surveillance for public places. In this paper, we first discuss the crowd feature selection and extraction and propose a multiple-frame feature point detection and tracking based on the KLT tracker. We state that behavior modelling of crowd is usually coarse compared to that for individuals. Instead of developing general crowd behavior models, we propose to model crowd events for specific end-user scenarios. As a result, a same type of event may be modelled slightly differently from one scenario to another and several models are to be defined. Consequently, fast modelling is required and this is enabled by the use of an extended Scenario Recognition Engine (SRE) in our approach. Crowd event models are defined; particularly, composite events accommodating evidence accumulation allow to increase detection reliability. Tests have been conducted on real surveillance video sequences containing crowd scenes. The crowd tracking algorithm proves to be robust and gives reliable crowd motion vectors. The crowd event detection on real sequences gives reliable results of a few common crowd behaviors by simple dedicated models.

Key words: Automatic video-based surveillance, crowd tracking, dedicated modelling, crowd behavior recognition

1 Introduction

Video surveillance has become a topic more and more important with the advent of technology and because of the increasing need for security. Central to the topic is automatic analysis and detection of abnormal events in public places or during public events. One particular class of public security issues is that involving a large number of people gathering together (crowding), such as public assemblies, sport competitions, demonstrations (eg., strikes, protests), etc. Because of the high level of degeneration risk, the security of public events involving large crowd has always been of high concern to relevant authorities. In recent years, a number of security agencies specialized in crowd management have emerged to respond to the need. Especially, this problem has started to draw attention of the research community for automatic detection of abnormal crowd behaviors during public events [17]. Technically speaking, crowd behavior analysis can be divided into two tasks: (1) motion information extraction and (2) abnormal behavior modelling. The former usually amounts to crowd tracking. It is a process by which we estimate the speed, direction and location of crowd in a video sequence. Higher-level models of crowd behavior can be used to detect anomalous events. A *crowd element* can be defined as a region corresponding to more than one person which has coherent and homogeneous motion. We are looking for crowd behavior such as flow divergence and convergence in the region of interest, emergence of a new crowd flow from an existing crowd flow having different characteristics, crowd flow divergence caused by falling person(s), people moving in counter direction, erratic motion beside the main crowd flow (as in the case of fighting), etc. All such scenarios can be defined in terms of crowd elements having specific motion parameters and characteristics.

Crowd movement tracking is quite different from tracking individuals in the crowd. When individuals are being tracked, the information is computed at the level of each individual. But when we talk about crowd tracking, the focus is on the motion of a crowd as a set of small elements whose structure changes continuously, instead of tracking the same entity throughout the video. One application of crowd tracking is the use of crowd tracking to build models of crowd behavior and to detect anomalous behaviors at the crowd level rather than at the individual level. Crowd analysis also finds applications in crowd simulation, crowd management, disaster management, egress planning as well as other related areas.

There are different research topics in video surveillance related to crowds: crowd density estimation, face detection and recognition in crowds, crowd behavior monitoring. In this paper, we focus on the problem of crowd behavior and abnormal event detection in crowd flows. Our approach does not look for a general model usable for all the scenes and all the scenarios. Instead, for each end-user scenario, dedicated models of crowd events are defined. This leads to many simple but robust models. It requires in turn appropriate tools for fast modelling. In our work, we use a Scenario Recognition Engine (SRE) [14], extended to account for crowd objects. In the remaining sections, we first give a short review of previous works related to crowd video surveillance (Section 2). Then in Section 3 we describe our crowd tracking method, in particular, the motion tracker we adopt and how it is extended to meet our needs. In Section 4 we discuss our approach to crowd behavior detection, including the use of a SRE and crowd behavior modelling. This is followed by our results and observations in Section 5. Finally we provide a conclusion and discuss avenues for future work (Section 6).

2 Previous works

Compared to other video surveillance problems such as segmentation, abandoned object detection, fighting etc., there are relatively few works related to crowds. Most previous works are concerned with two tasks: crowd density estimation and crowd velocity estimation.

2.1 Crowd density estimation

Crowd density estimation techniques can be classified into two categories: counting based methods and 2-D empirical methods. The most basic set of approaches for counting based methods rely on detecting humanoids or identifiable features of humanoids (like faces) to count the number of people in the image. Another approach to humanoid detection was illustrated in [1] where the motion pattern of humans was utilized to better identify pedestrians in a scene. These methods however do not perform well under conditions of high occlusion and in situations where the camera resolution is not good enough for identifying these features. Use of several cameras can leverage this problem [2] but overlapping cameras are not always available in public places.

Other methods do not directly measure the crowd density by counting people. Ma *et al.*[12] compute the crowd density based on a linear relationship learned beforehand, especially taking into account geometrical distortion from the ground plane to the image plane. Other methods rate the crowd density on an ordinal scale, categorizing it into highly dense, somewhat dense, sparsely dense etc. For example, Marana *et al.* use the statistical texture analysis where a GLDM (Gray level dependence matrix) is computed [3]. A FLD-type neural network is then trained on features like contrast, homogeneity, entropy and energy derived from this GLDM to classify various densities.

2.2 Crowd motion and behavior

Some approaches have been attempted in the past to estimate the speed and direction of crowd motion. The first approach is based on block matching across consecutive frames to determine how the blocks have displaced in the next frame. The criterion for block matching is correlation. Another criterion which can be used is the optical flow [4]. Both of these approaches basically search for similarities in the neighborhood of a block and rely on matching the intensity levels across frames. They suffer from difficulties such as inhomogeneous background, constantly changing crowd appearance and heavy computational costs.

Another approach which has been tried is to track multiple people across frames and from that determine the motion information. One such approach uses the BraMBLe [6] (Bayesian Multiple Blob Tracker) to track individuals across frames and then collate this information to determine multiple people velocity information. Similar approaches such as detecting and tracking faces across frames have been tried as well. These methods suffer from problems such as ID-swapping, occlusions (resulting in missing an object during detection phase), etc. More importantly, this approach cannot handle situations whereby crowd is dense or cameras are far away from the crowd.

There are many works on crowd movement simulation in the context of crowd management. However little research has been done on crowd behavior modelling for abnormal event detection in the context of security surveillance. In [13], Andrade *et al.* proposed a method for crowd event modelling in a constrained environment (*eg.*, corridor). The key idea is to create normal crowd behavior models by learning, using HMM to model crowd motion based on optical flow. Its effectiveness is shown with simulated data. Yet the computational load tends to be high and it is not sure that the method could be applied to much less constrained cases such as outdoor demonstrations.

3 Crowd tracking

Crowd tracking generally is often a harder problem to solve as compared to tracking individuals. There are issues such as handling a larger degree of occlusion which can create problems in tracking crowd elements across frames successfully. Many methods to track crowd may not be so scalable: they may work for low density crowd and not for higher density ones, or they may work for a camera placed closer to the crowd and not in the cases where the camera is quite far away. Inhomogeneous background may also create problems for the tracking process. On the other hand, a set of stable and well distributed points in a crowd region should allow to determine consistent motion of the crowd. If some of them are lost they can be replaced by new ones and crowd motion can still be determined. Therefore we are not much concerned with problems related to occlusions, as in tracking of individuals. Our problem now becomes how to obtain reliable crowd motion.

In this section we show how to characterize crowd velocity. To estimate crowd velocity in a video sequence, we perform KLT (Kanade Lucas Tomasi) tracking on video frames, looking for interesting feature points in the scene and tracking them over time. Note that here motion detection and extraction are combined into one optimization process. We then quantize these crowd motion vectors based on their directions. This provides us with classes of similar crowd motion vectors (similarity of direction). We now cluster the vectors in each of these classes based on spatial locality. This gives us the regions where a particular direction of motion is prominent. We also compute speed statistics for each of these clusters. Thus we are able to obtain the areas where a crowd motion exists, its direction and speed along with its strength. We now elaborate each of these steps.

3.1 KLT tracking of feature points

The KLT (Kanade-Lucas-Tomasi) tracker [9] [10] is among the most robust and real-time algorithms for feature point tracking. Instead of the two-step "detection-then-matching" tracking, as with most other trackers, the KLT tracker optimizes a functional which combines feature point extraction and tracking. Only feature points good (local patch strongly textured) to track are extracted, resulting in tracked feature points which are much more reliable.

In our approach, we first select the number of good feature points we want to get for the first frame of the videos sequence. The algorithm introduces automatically new feature points to replace those lost in the subsequent frames. In this way we are able to figure out, for each feature point in the current frame: its start frame and its 2-D trajectory over last few frames, which can be used to extract crowd motion vectors (next step).

3.2 Computation of crowd motion vectors

The crowd motion vector is defined as a vector representing the displacement of a feature point over some number of frames (some temporal interval). It indicates how much a feature point has moved and in which direction. If the position of some feature point P was $\boldsymbol{r}_1$ at time t_1 and was $\boldsymbol{r}_2$ at time t_2, then the crowd motion vector representing the motion of P is $\boldsymbol{r}_2 - \boldsymbol{r}_1$.

Crowd motion vectors that span over multiple video frames, are obtained by connecting motion vectors computed by KLT on two consecutive frames.

Two types of crowd vectors are computed: *short-term* crowd motion vectors and *long-term* crowd motion vectors. These differ in the time (frame) intervals T chosen for them, namely . The long-term vector of a feature is the displacement from the $(i - T_{LT})^{th}$ frame to the current frame i. Similarly the short-term vector is the displacement between $(i - T_{ST})^{th}$ frame to the current frame i, with $T_{LT} > T_{ST}$. Note that the values of T_{LT} and T_{ST} are parameters to the algorithm. We have therefore two sets of crowd motion vectors: Ω for all long-term vectors of the current frame and Φ for all short-term vectors. Basically, the short-term crowd motion vectors are used to validate the feature point mobility.

3.3 Stationary feature point removal

While all crowd motion vectors represent feature point motion, some of these may not denote actual crowd motion. Many vectors may be of zero length because the feature point did not move at all, or may have small lengths, owing to noise in video acquisition, inhomogeneous background or even minuscule movement of the crowd. These should be filtered out so that they do not affect the computation of speed and direction. This is done based on short-term crowd motion vectors.

3.4 Direction-based quantization of motion vectors

Once the set of useful vectors are determined, we now classify them into classes, based on their directions. This is to group together the crowd motion vectors having almost same directions. In our implementation a simple quantization of 8 directions is performed, leading to 8 classes.

3.5 Clustering of crowd motion vectors

With the motion vectors quantized based on their directions, we are now ready to cluster the motion vectors with same direction on the basis of their spatial locations. By performing clustering, we can determine which directions of motion are prominent and in which areas in the region of interest. This step is essential to be able to determine the various crowd elements that exist in a scene and then to compute the motion parameters for them.

Since a crowd element represents the collective homogeneous and coherent motion of more than one person, it is highly likely that the number of crowd motion vectors corresponding to a crowd element are much more than one. In such a scenario, it becomes essential to cluster such vectors together to be able to identify the crowd elements correctly and to ascertain their spatial locations. For the purpose of clustering, the spatial location of a crowd motion vector is taken as the middle point of the indicator. We then apply the K-means clustering algorithm which minimizes the square distance of data points from the cluster centers. It starts with a random set of K centers and then iteratively improves upon the locations of these centers till a configuration is obtained which minimizes the square distance between the crowd motion vectors and the corresponding cluster centers. Here we need to specify an appropriate number of clusters per direction in advance.

After clustering, we use the crowd motion vectors belonging to a cluster to compute the mean speed associated with the cluster. We also use the center of the cluster to approximate the floor space coordinates where this motion exists. We can also compute other parameters such as speed variance, frequency/strength of this motion, etc. All this statistical information is useful for higher level analysis.

4 Crowd Behavior Recognition

Normal behaviors are usually characterized by regular motion direction, low (walking) speed, stopping, etc. Abnormal behaviors include opposite movement in crowd, bifurcation, deviation, fighting, and are characterized by corresponding motion attributes. Therefore, crowd behavior can be inferred from crowd motion information, carried by the long-term and short-term motion vectors discussed above. On the other hand, whether a crowd behavior is normal is also context-dependent. For example, sudden crowd stopping in a subway corridor may be considered as abnormal.

Compared to the behavior of individual which can have a more complex spatio-temporal dynamics depending on the scenarios of interest, crowd behavior is usually much simpler, as mentioned above. Furthermore, by developing dedicated models for crowd event of interest to specific scenarios, crowd models can be expected to be simpler and more robust. Therefore our main concerns here become the following: (1) How to model crowd behaviors such that robust results can be obtained? (2) How to design a platform which allows the end-users for quick crowd behavior modelling?

In the following, we will discuss what are crowd events we are generally interested in and how they can be modelled and detected. Then we will describe the SRE which allows to quickly create many dedicated crowd event models.

4.1 Crowd behavior modelling

As discussed before, the main information we rely on is crowd motion vectors, i.e., those motion vectors of feature points computed over multiple frames. Therefore, we derive the following motion attributes to be used in the crowd behavior models: *principal directions*, *speed*, and *crowd mobility*.

Principal directions are simply computed from the direction histogram of crowd motion vectors. In a demonstration, the majority of crowd vectors are aligned in one direction, i.e., there is one principal direction. The secondary principal direction, if existing, should involve only a small number of vectors. For instance, during a demonstration, abnormal crowd events can be declared if there are multiple principal directions. Note that because of the direction quantization, two main principal directions can also occur but these are neighboring directions.

The speed is directly related to the length of crowd vectors. In video surveillance, the camera is usually fixed. Therefore, if camera is calibrated and the normal walking speed is estimated and the video frame rate is known, the vector length corresponding to the normal crowd speed can be determined. Note that the calibration does not need to be very accurate. We use a simple calibration process that requires only two pairs of parallel lines (perpendicular to each other) on the ground and the height of a reference object (*eg.*, a person).

Crowd mobility is determined from the number of valid featured points. Within a reasonably short interval, the texture of a crowd image does not change much so the number of features should stay constant. This attribute can thus be used to determine abnormal events if the number of valid feature points changes drastically.

4.2 Scenario Recognition Engine

To quickly model crowd behavior for different scenarios, we use the platform proposed in [14]. It contains a scenario recognition engine (SRE) and an event modelling language. The SRE was initially intended for event detection in indoor scenes involving single persons. We have extended to cope with crowd, together with the crowd attributes as described above. Yet we have purposely introduced only low-level attributes to keep the SRE as generic as possible. For intermediate attributes (eg. principal direction) derived from low-level ones, they are computed using the facility functions provided within the SRE [15]. Various constants, such as threshold and relevant directions (Normal direction, Left lateral direction, Opposite direction, etc.) are also defined using the provided facility functions. With the event modelling language, scenarios can be defined in an incremental way, from simple (called 'Primitive') to complex (called 'Composite'), as can be seen in the example of the next section.

An event model is composed of five parts:

– the physical objects (including moving objects and zones of interest) involved in the event,
– the components (i.e. sub-events) that should be recognized,
– the forbidden components that should not occur during the recognition,
– the constraints between the physical objects and the components,
– alarms, including the alarm message and the alert level.

5 Results

In this section, we present results on crowd vector computation and clustering and crowd behavior detection using real video surveillance video data. The real challenge consists in collecting crowd videos illustrating events of interest. To our knowledge, no video database is available for benchmarking crowd behavior analysis algorithms.

5.1 Crowd vector computation and clustering

We have conducted extensive tests of our crowd motion computation algorithm. The results show that the algorithm delivers reliable motion vectors as required by the subsequent crowd behavior detection task. The first example video clip is an overhead video of a street in which a high density crowd is moving in one prominent direction, whereas motion of small number of persons exists elsewhere as well (Figure 1(a)(b)). In the processed video, we have been able to determine the presence of motion in a direction opposite to the major direction of crowd motion. Figure 1(c)(d) show the tracking and clustering results for an indoor staircase and a stadium corridor, respectively. In particular, the person going down the stairs, opposite to the crowd going up, is detected (Figure 1(c)).

5.2 Crowd behavior detection

To validate our approach, a mass demonstration mock-up has been conducted, simulating crowd movements and the police reactions. We have built 31 event models using 8 cameras, for monitoring the crowd and looking into police vehicle movements relating to the demonstrating crowds. In Table 1 we summarize the crowd behaviors we have modelled.

The sequence *ID_CONTROL* is about a bar terrace video which contains two abnormal events: *Erratic Movement* (fighting) and *Falling*. Each event is defined as composite event composed of a primitive event. The composite event is defined to accumulate the primitive event evidence by computing the overall event value within a temporal window (typically 5-10 frames). The accumulation of primitive events enable to increase reliability of the detection, taking advantage of the temporal inference capability of the recognition engine, such as Allen temporal algebra [14]. To reduce false detection, a *zone of interest* – the terrace area – is defined in a context file [16] and is associated with the event models. The sequence *Demonstration_Fight* is a demonstration which contains two abnormal

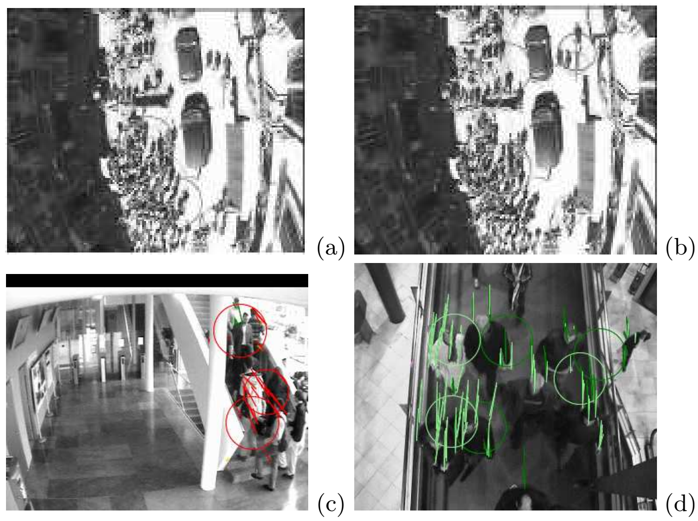

Fig. 1. Crowd motion vector computation and clustering.

events: *Opposite Movement in Crowd* and *Strong Lateral Movements* (a group of people run from the left to the right and robbed a person on the right side of the road and then run back to left). Again as above, each event is defined as a composite event composed of a primitive event in order to increase reliability of the detection. Figure 2 shows an excerpt of a model using the Scenario Description Language (SDL) of the Scenario Recognition Engine [15]. In this model a crowd element is constituted of a set of motion vectors. The constraints consist in verifying that a sufficient number of motion vectors have the direction opposite to the main crowd flow and the opposite movement is detected at least N times (duration). The events modelled in the sequence *Demonstration_Robbery* are similar to those in *Demonstration_Fight* describing interaction between the crowd and the police forces.

As discussed in Sec 4, in our approach event modelling is specific to end-user scenarios. Therefore simple event models are obtained in each specific case, leaving to relatively easy detection of the events of interest. Our experimental results show that the required events, not limited to crowd-related ones, are all successfully recognized. Also, evidence accumulation proves to be effective to increases reliability of the detection.

6 Conclusion

In this paper, we have presented our work on crowd behavior recognition. We have discussed the crowd feature selection and extraction. A multiple-frame feature point detection and tracking based on the KLT tracker has been proposed and intensive tests have been conducted on real surveillance video sequences containing crowd scenes. The algorithm proves to be robust and gives reliable crowd motion vectors. We then discussed the crowd behavior detection. Instead

Table 1. Summary of crowd behaviors modelled.

Camera/Seq.	Modelled events	Features used
ID_CONTROL	Erratic movement (fighting, Abnormal)	Number of motions directions
	Falling (Abnormal)	Principal directions, Nb of vectors
Demonstration _Fight	Crowd Moving Forward (Normal)	Principal directions
	Opposite Movement in Crowd (Abnormal)	Principal directions
	Strong Lateral Crossing to Right (robbery by crowd, Abnormal)	Principal directions
	Strong Lateral Crossing to Left (running back, Abnormal)	Principal directions
	Crowd Stopped(Normal)	Numbers of motion vectors
Demonstration _Robbery	Crowd Moving Forward (Normal) Crowd Moving to Right	Principal directions Principal directions
	Crowd Moving to Left	Principal directions
	Crowd Stopped (Normal))	Numbers of motion vectors

```
PrimitiveState(opposite_movement_in_crowd,
        PhysicalObjects((c : Crowd_Element))
        Constraints( (c's MotionVectorDirections < ThreshNbDirections)
                     (c's MotionVectorNumber >= ThreshMobility )
                     (c's PrincipalDir1 = CrowdDirection)
                     (c's PrincipalDir2 = OppDirection)
                     (c's Duration > ThreshDuration) )
        Alarm( AText("Opposite Movement in Crowd!")
               AType("URGENT") )
)
```

Fig. 2. Excerpt of a crowd event model for detecting opposite movement during a demonstration.

of developing general crowd behavior models, we proposed to model crowd events only of interest to each camera. In this approach, a same type of event may be modelled slightly different from one scenario to another and many models are to be defined. Fast modelling is enabled by the use of an extended Scenario Recognition Engine in our approach. We have shown how crowd event models are defined. Particularly composite events accommodating evidence accumulation allow to increase detection reliability. Tests have been conducted on real surveillance video sequences containing crowd scenes and the algorithm proves to be robust and gives reliable crowd motion vectors. The crowd event detection on real sequences gives reliable results of a few common crowd behaviors. In the future, we plan to work in the following directions. One is to work out a way to determine as automatically as possible the parameters involved in crowd motion computation. Also we will review the noise vector removal because sometimes small vector length conveys relevant information about the crowd behavior too (eg. slowing down, stop). The second is to apply the approach to more thorough and finer scenarios (e.g., throwing a stone from a crowd) as well as related issues such as false alarms when the number of defined events becomes important.

Fig. 3. Multiple-frame KLT tracking.

Fig. 4. Erratic movement and falling on a bar terrace.

References

1. P. Viola, M.J. Jones, D. Snow, Detecting Pedestrians Using Patterns of Motion and Appearance, *International Journal of Computer Vision*, 63(2), 2005, 153-161.
2. Khan, S.M., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: 9th European Conference on Computer Vision, LNCS, vol. 3954, pp. 133146. Springer, Heidelberg, 2006.
3. A.N. Marana, S.A. Velastin, L.F. Costa, R.A.Lotufo, Automatic Estimation of Crowd Density Using Texture, *Safety Science*, 28(3), 1998, 165-175.
4. A.C. Davies, J.H. Yin and S.A. Velastin, Crowd Monitoring Using Image Processing, *IEE Electronic and Communications Engineering Journal*, 7(1), 1995, 37-47.
5. X. HUANG, T. SIM, Stereo-based human head detection from crowd scenes, *Proc. International Conference on Image Processing*, Singapore, 2004, 1353-1356.

6. M. Isard and J. MacCormick, BraMBLe: A Bayesian Multiple-Blob Tracker, *Proc. International Conference on Computer Vision*, Vancouver, Canada, 2001, vol.2 34-41.
7. N. Paragios, V. Ramesh, A MRF-based approach for real time subway monitoring, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Madison, USA, 2003, 1034-1040.
8. V. Colin de Verdie, J.L. Crowley, Visual Recognition using Local Appearance, *Proc. European Conference on Computer Vision*, Freiburg, Germany, 1998, vol.1 640-654.
9. B.D. Lucas, T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision, *International Joint Conference on Artificial Intelligence*, Vancouver, Canada, 1981, 674-679.
10. J. Shi, C. Tomasi, Good Features to Track, *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 1994, 593-600.
11. W.E.L. Grimson, C. Stauffer, R. Romano, L. Lee, Using Adaptive Tracking to Classify and Monitor Activities in a Site, *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, USA, 1998, 22-27.
12. Ruihua Ma, Liyuan Li, Weimin Huang, Qi Tian. On Pixel Count Based Crowd Density Estimation. *IEEE Conference on Cybernetics and Intelligent Systems*. December 1-3, 2004, Singapore.
13. E. L. Andrade, S. Blunsden, R. B. Fisher. Modelling Crowd Scenes for Event Detection, in *Proc. Int. Conf. on Pat. Recog. (ICPR'06)*, Vol 1, pp 175 - 178, Hong Kong, Aug. 2006.
14. T. Vu, F. Brémond and M. Thonnat. Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition. *The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, 9-15 August 2003.
15. T. Vu. Scenario Description Language for Video Interpretation. Internal document. INRIA Sophia Antipolis, Revision 2007.
16. Alberto Avanzi, Francois Bremond, Christophe Tornieri and Monique Thonnat, Design and Assessment of an Intelligent Activity Monitoring Platform, in *EURASIP Journal on Applied Signal Processing, special issue in "Advances in Intelligent Vision Systems: Methods and Applications"*, 2005:14, pp.2359-2374.
17. Beibei Zhan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin and Li-Qun Xu. Crowd analysis: a survey. To appear in Journal of Machine Vision and Applications. Published online: 10 April 2008.