21

Uncertainty Control for Reliable Video Understanding on Complex Environments

Marcos Zúñiga[1], François Brémond[2] and Monique Thonnat[3]
[1]*Electronics Department, Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso*
[2,3]*Project-Team PULSAR, INRIA, 2004 route des Lucioles, Sophia Antipolis*
[1]*Chile*
[2,3]*France*

1. Introduction

The most popular applications for video understanding are those related to video-surveillance (e.g. alarms, abnormal behaviours, expected events, access control). Video understanding has several other applications of high impact to the society as medical supervision, traffic control, violent acts detection, crowd behaviour analysis, among many others. This interest can be clearly observed through the significant number of research projects approved in this domain: GERHOME [1], CARETAKER [2], ETISEO [3], BEWARE [4], SAMURAI [5], among many others.
We propose a new generic video understanding approach able to extract and learn valuable information from noisy video scenes for real-time applications. This approach is able to estimate the reliability of the information associated to the objects tracked in the scene, in order to properly control the uncertainty of data due to noisy videos and many other difficulties present in video applications. **This approach comprises motion segmentation, object classification, tracking and event learning phases.**
A fundamental objective of this new approach is to treat the video understanding problem in a generic way. This implies implementing a platform able to classify and track diverse objects (e.g. persons, cars, air-planes, animals), and to dynamically adapt to different scene and video configurations. This generality will allow to adapt the approach to different applications with minimal effort. Achieving a completely general video understanding approach is an extremely ambitious goal, due to the complexity of the problem and the infinite possibilities of situations occurring in real-time. That is why it must be considered as a long term goal, considering many building blocks in the process. **This work is focused on building the first fundamental blocks allowing a proper management of uncertainty of data in every phase of the video understanding process.**

[1] GERHOME Project 2005, `http://gerhome.cstb.fr`
[2] CARETAKER Project 2006, `http://cordis.europa.eu/ist/kct/caretaker_synopsis.htm`
[3] ETISEO Project 2006, `http://www-sop.inria.fr/orion/ETISEO/`
[4] BEWARE Project 2008, `http://www.eecs.qmul.ac.uk/~sgg/BEWARE/`
[5] SAMURAI Project 2008, `http://www.samurai-eu.org/`

To date, several video understanding platforms have been proposed in the literature (Hu et al., 2004; Lavee et al., 2009). These platforms are normally designed for specific contexts or for treating specific issues. Normally, they are tested over well-known videos or in extremely controlled environments in order to be validated. Moreover, reality is not controlled and it is hardly well-known. **The main novelty of this research is to treat the video understanding problem in a general way**, by modelling different types of uncertainty introduced when analysing a video sequence. Modelling uncertainty allows to understand when something will go wrong in the analysis and then to prepare the system to take the necessary actions for preventing this situation.

The main contributions of the proposed approach are: (i) a new algorithm for tracking multiple objects in noisy environments, (ii) the utilisation of reliability measures for modelling uncertainty in data and for proper selection of valuable information extracted from noisy data, (iii) the improved capability of tracking to manage multiple visual evidence-target associations, (iv) the combination of 2D image data with 3D information in a dynamics model governed by reliability measures for proper control of uncertainty in data, and (v) a new approach for event recognition through incremental event learning, driven by reliability measures for selecting the most stable and relevant data.

This chapter is organised as follows. First, Section 2 describes the state-of-the-art focused on justifying the decisions taken for each phase of the approach. Next, Section 3 describes the proposed approach and the involved phases. Then, Section 4 presents results for different benchmark videos and applications.

2. Related work

As properly stated in (Hu et al., 2004), general structure in video understanding is comprised by four main phases: motion segmentation, object classification, tracking, and behaviour analysis (event recognition and learning). In general, two phases can be identified as critical for the correct achievement of any further event analysis in video: image segmentation and object tracking. Image segmentation (McIvor, 2000) consists in extracting motion from a currently analysed image frame, based on information extracted from previously acquired information (e.g. background image or model). Multi-target tracking (MTT) problem (Yilmaz et al., 2006) consists in estimating the trajectory of multiple objects as they move in a video scene. In other words, tracking consists in assigning consistent labels to the tracked objects in different frames of a video.

One of the first approaches focusing on MTT problem is the Multiple Hypothesis Tracking (MHT) algorithm (Reid, 1979), which maintains several correspondence hypotheses for each object at each frame. Over more than 30 years, MHT approaches have evolved mostly on controlling the exponential growth of hypotheses (Bar-Shalom et al., 2007; Blackman et al., 2001). For controlling this combinatorial explosion of hypotheses all the unlikely hypotheses have to be eliminated at each frame (for details refer to (Pattipati et al., 2000)). MHT methods have been extensively used in radar (Rakdham et al., 2007) and sonar tracking systems (Moran et al., 1997). In (Blackman, 2004) a good summary of MHT applications is presented. However, most of these systems have been validated with simple situations (e.g. non-noisy data).

The dynamics models for tracked object attributes and for hypothesis probability calculation utilised by the MHT approaches are sufficient for point representation, but are not suitable for this work because of their simplicity. The common feature in the dynamics model of these algorithms is the utilisation of Kalman filtering (Kalman, 1960) for estimation and prediction of object attributes.

An alternative to MHT methods is the class of Monte Carlo methods. The most popular of these algorithms are CONDENSATION (CONditional DENSity PropagATION) (Isard & Blake, 1998) and particle filtering (Hue et al., 2002). They represent the state vector by a set of weighted hypotheses, or particles. Monte Carlo methods have the disadvantage that the required number of samples grows exponentially with the size of the state space. In these techniques, uncertainty is modelled as a single probability measure, whereas uncertainty can arise from many different sources (e.g. object model, geometry of scene, segmentation quality, temporal coherence, appearance, occlusion).

When objects to track are represented as regions or multiple points other issues must be addressed to properly perform tracking. Some approaches have been found pointing in this direction (e.g. in (Brémond & Thonnat, 1998), the authors propose a method for tracking multiple non-rigid objects; in (Zhao & Nevatia, 2004), the authors use a set of ellipsoids to approximate the 3D shape of a human).

For a complete video understanding approach, the problem of obtaining reliable information from video concerns the proper treatment of the information in every phase of the video understanding process. For solving this problem, each phase has to measure the quality of the concerning information, in order to be able of evaluating the overall reliability of a framework. Reliability measures have been used in the literature for focusing on the relevant information, allowing more robust processing (e.g. (Heisele, 2000; Nordlund & Eklundh, 1999; Treetasanatavorn et al., July 2005)). Nevertheless, these measures have been only used for specific tasks of the video understanding process.

The object representation is a critical choice in tracking, as it determines the features which will be available to determine the correspondences between objects and acquired visual evidence. Simple 2D shape models (e.g. rectangles (Cucchiara et al., 2005), ellipses (Comaniciu et al., 2003)) can be quickly calculated, but they lack in precision and their features are unreliable, as they are dependant on the object orientation and position relative to camera. In the other extreme, specific object models (e.g. articulated models (Boulay et al., 2006)) are very precise, but expensive to be calculated and lack of flexibility to represent objects in general. In the middle, 3D shape models (e.g. cylinders (Scotti et al., 2005), parallelepipeds (Yoneyama et al., 2005)) present a more balanced solution, as they can still be quickly calculated and they can represent various objects, with a reasonable feature precision and stability. As an alternative, appearance models utilise visual features as colour, texture template, or local descriptors to characterise an object (Quack et al., 2007). They can be very useful for separating objects in presence of dynamic occlusion, but they are ineffective in presence of noisy videos, low contrast, or objects too far in the scene, as the utilised features become less discriminative.

In the context of video event learning, most of these approaches are supervised using general techniques as Hidden Markov Models (HMM) and Dynamic Bayesian Network (DBN) (Ghahramani, 1998), requesting annotated videos representative of the events to be learnt. Few approaches can learn events in an unsupervised way using clustering techniques. For example, in (Xiang & Gong, 2008) the authors propose a method for unusual event detection, which first clusters a set of seven blob features using a Gaussian Mixture Model, and then represents behaviours as an HMM, using the cluster set as the states of the HMM.

Some other techniques can learn on-line the event model by taking advantage of specific event distributions. For example, in (Piciarelli et al., 2005), the authors propose a method for incremental trajectory clustering by mapping the trajectories into the ground plane decomposed in a zone partition. Their approach performs learning only on spatial information, it cannot take into account time information, and do not handle noisy data.

Briefing, among the main issues present in video analysis applications are their lack of generality and adaptability to new scenarios. This lack of generality can be observed in several aspects: (a) applications focused on few object attributes and not suited to process new ones, (b) processes not capable of interpreting uncertainty in input, processed data, and algorithms, (c) tracking approaches not properly prepared to treat several observations (visual evidences) associated to the same target (or object) (e.g. detected object parts), (d) learning approaches incorporating data that can be really noisy or even false, (e) applications focused in scenarios with very restricted environmental (e.g. illumination), structural (e.g. cluttered scene) and geometric conditions (e.g. camera view angle).
Next section details a new video understanding approach, facing several of the main issues previously discussed.

3. Video analysis approach with reliability measures for uncertainty control

All the issues involved with the different stages of the video understanding process introduce different types of uncertainty. For instance, different zones of an image frame can be affected by different issues (e.g. illumination changes, reflections, shadows), or object attributes at different distances with respect to the camera present different estimation errors, and so on. In order to properly control this uncertainty, reliability measures can be utilised. Different types of uncertainty can be modelled by different reliability measures, and these measures can be scaled and combined to represent the uncertainty of different processes (e.g. motion segmentation, object tracking, event learning).
This new video understanding approach is composed of four tasks, as depicted in Figure 1.

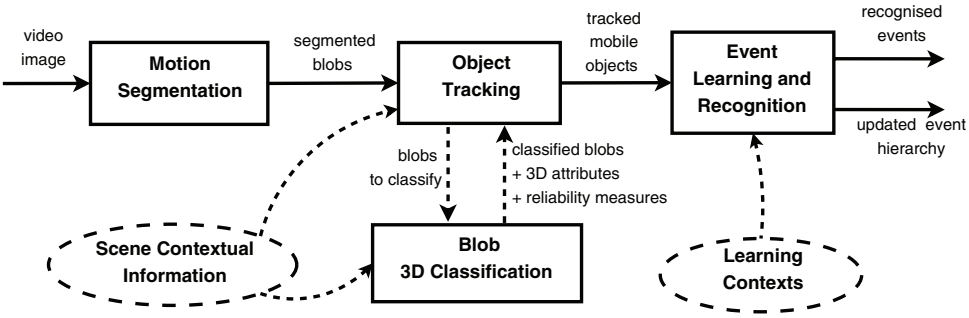


Fig. 1. Proposed video understanding approach.

First, at each video frame, a segmentation task detects the moving regions, represented by bounding boxes enclosing them. We first apply an image segmentation method to obtain a set of moving regions enclosed by a bounding box (*blobs* from now on). More specifically, we apply a background subtraction method for segmentation, but any other segmentation method giving as output a set of blobs can be used. The proper selection of a segmentation algorithm is crucial for obtaining quality overall system results. For the context of this work, we have considered a basic thresholding algorithm (McIvor, 2000) for segmentation in order to validate the robustness of the tracking approach on noisy input data. Anyway, keeping the segmentation phase simple allows the system to perform in real-time.
Second, and using the output blobs from segmentation as input, a new tracking approach is performed to generate the hypotheses of tracked objects in the scene. The tracking phase

uses the blobs information of the current frame to create or update hypotheses of the mobiles present in the scene. These hypotheses are validated or rejected according to estimates of the temporal coherence of visual evidence. The hypotheses can also be merged or split according to the separability of observed blobs, allowing to divide the tracking problem into groups of hypotheses, each group representing a tracking sub-problem. The tracking process uses a 2D merge task to combine neighbouring blobs, in order to generate hypotheses of new objects entering the scene, and to group visual evidence associated to a mobile being tracked. This blob merge task simply combines 2D information. A new 3D classification approach is also utilised in order to obtain 3D information about the tracked objects, which provides new means of validating or rejecting hypotheses according to a priori information about the expected objects in the scene.

This new 3D classifier associates an object class label (e.g. person, vehicle) to a moving region. This class label represents the object model which better fits with the 2D information extracted from the moving region. The objects are modelled as a 3D parallelepiped described by its width, height, length, position, orientation, and visual reliability measures of these attributes. The proposed parallelepiped model representation allows to quickly determine the type of object associated to a moving region and to obtain a good approximation of the real 3D dimensions and position of an object in the scene. This representation tries to cope with the majority of the limitations imposed by 2D models, but being general enough to be capable of modelling a large variety of objects and still preserving high efficiency for real world applications. Due to its 3D nature, this representation is independent from the camera view and object orientation. Its simplicity allows users to easily define new expected mobile objects. For modelling uncertainty associated to visibility of parallelepiped 3D dimensions, reliability measures have been proposed, also accounting for occlusion situations.

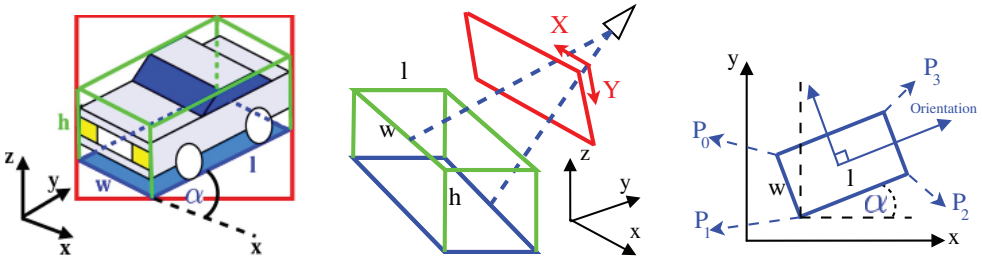
Finally, we propose a new general event learning approach called **MILES** (**M**ethod for **I**ncremental **L**earning of **E**vents and **S**tates). This method aggregates on-line the **attributes** and **reliability information** of tracked objects (e.g. people) to **learn** a hierarchy of concepts corresponding to **events**. Reliability measures are used to focus the learning process on the most valuable information. Simultaneously, MILES **recognises** new occurrences of events previously learnt. The only hypothesis of MILES is the availability of tracked object attributes, which are the needed input for the approach, which is fulfilled by the new proposed tracking approach. MILES is an incremental approach, which allows on-line learning, as no extensive reprocessing is needed upon the arrival of new information. The incremental aspect is important as the available examples of the training phase can be insufficient for describing all the possible scenarios in a video scene. This approach proposes an automatic bridge between the low-level image data and higher level conceptual information, where the learnt events can serve as building blocks for higher level behavioural analysis. The main novelties of the approach are the capability of learning events in general and on-line, the utilisation of a explicit quality measure for the built event hierarchy, and the consideration of measures to focus learning in reliable data.

The 3D classification method utilised in this work is discussed in the next section 3.1. Then, in section 3.2 the proposed tracking algorithm is described. Next, in section 3.3, MILES algorithm for event learning is described.

3.1 Reliable classification using 3D generic models

The proposed tracking approach interacts with a 3D classification method which uses a generic parallelepiped 3D model of the expected objects in the scene. The parallelepiped

model is described by its 3D dimensions (width w, length l, and height h), and orientation α with respect to the ground plane of the 3D referential of the scene, as depicted in Figure 2(a). The utilised representation tries to cope with several limitations imposed by 2D representations, but keeping its capability of being a general model able to describe different objects, and a performance adequate for real world applications.



(a) (b) (c)

Fig. 2. 3D parallelepiped model for detected objects. (a) Vehicle enclosed by a 2D bounding box (coloured in red) and by the parallelepiped representation (blue base and green projections). (b) 3D view of the scene. (c) Top view of the scene.

A large variety of objects can be modelled (or, at least, enclosed) by a parallelepiped. The proposed model is defined as a parallelepiped perpendicular to the ground plane of the analysed scene. Starting from the basis that a moving object will be detected as a 2D blob b with 2D limits $(X_{left}, Y_{bottom}, X_{right}, Y_{top})$, 3D dimensions can be estimated based on the information given by pre-defined 3D parallelepiped models of the expected objects in the scene. These pre-defined parallelepipeds, which represent an object class, are modelled with three dimensions w, l, and h described by a Gaussian distribution (representing the probability of different 3D dimension sizes for a given object), together with a minimal and maximal value for each dimension.
Formally, a pre-defined 3D parallelepiped model $Q_{\mathbf{C}}$ for an object class $\mathbf{C}$ can be defined as:

$$Q_{\mathbf{C}} = \{(\mathcal{N}(\mu_q, \sigma_q), q_{min}, q_{max}) | q \in \{w, l, h\}\}, \tag{1}$$

The objective of the classification approach is to obtain the class C for an object $\mathbf{O}$ detected in the scene, which better fits with an expected object class model $Q_{\mathbf{C}}$.
A 3D parallelepiped instance $S_{\mathbf{O}}$ for an object O (see Figure 2) is described by:

$$S_{\mathbf{O}} = (\alpha, (w, R_w), (l, R_l), (h, R_h)), \tag{2}$$

Note that the orientation α corresponds to the angle between the length dimension l of the parallelepiped and the x axis of the 3D referential of the scene. where α represents the parallelepiped orientation angle (Figure 2(c)), defined as the angle between the direction of length 3D dimension and x axis of the world referential of the scene. The orientation of an object is usually defined as its main motion direction. Therefore, the real orientation of the object can only be computed after the tracking task. Dimensions w, l and h represent the 3D values for width, length and height of the parallelepiped, respectively. l is defined as the 3D dimension which direction is parallel to the orientation of the object. w is the 3D dimension which direction is perpendicular to the orientation. h is the 3D dimension parallel to the z axis

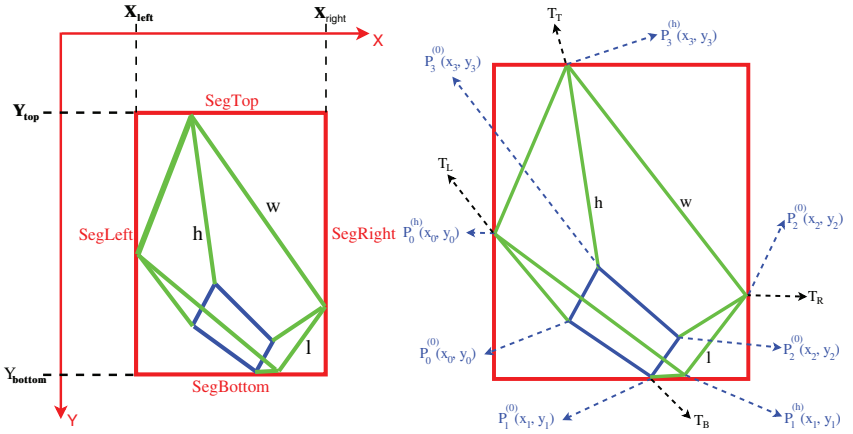


Fig. 3. Camera view of 3D parallelepiped model for detected objects. (a) Image 2D referential variables. (b) World 3D referential variables.

of the world referential of the scene. R_w, R_l and R_h are 3D visual reliability measures for each dimension. These measures represent the confidence on the visibility of each dimension of the parallelepiped and are described in Section 3.1.2.

The dimensions of the 3D model are calculated based on the 3D position of the vertexes of the parallelepiped in the world referential of the scene. Eight points $P_i^z(x_i, y_i) = (x_i, y_i, z)$ are defined, with $i \in \{0,1,2,3\}$ and $z \in \{0,h\}$, as the 3D points that define the parallelepiped vertexes, with $P_i^{(0)}$ corresponding to the i-th base point and $P_i^{(h)}$ corresponding to the i-th vertex on height h, as shown in Figure 3(b). Also, P_i are defined (and respectively E_i), with $i \in \{0,1,2,3\}$, as the 3D points (x_i, y_i) on the ground plane xy representing each vertical edge E_i of the parallelepiped, as depicted in Figure 2(b). The parallelepiped position (x_p, y_p) is defined as the central point of the rectangular base of the parallelepiped, and can be inferred from points P_i.

The idea of this classification approach is to find a parallelepiped bounded by the limits of the 2D blob b corresponding to a group of moving pixels. For completely determining the parallelepiped instance S_O, it is necessary to determine the values for the orientation α in 3D scene ground, the 3D parallelepiped dimensions w, l, and h and the four pairs of 3D coordinates from $P_i = (x_i, y_i)$, with $i \in \{0,1,2,3\}$, defining the base of the parallelepiped. Therefore, a total of 12 variables have to be determined.

To find these values, a system of equations has to be solved. A first group of four equations arise from the constraints imposed by the vertexes of the parallelepiped which are bounded by the 2D limits of the blob. Other six equations can be derived from the fact that the parallelepiped base points P_i, with $i \in \{0,1,2,3\}$, form a rectangle. Then, considering the parallelepiped orientation α, these equations are written in terms of the parallelepiped base points $P_i = (x_i, y_i)$, as shown in Equation (3).

$$\begin{array}{l} x_2 - x_1 = l \times cos(\alpha) \;\; ; \;\; y_2 - y_1 = l \times sin(\alpha) \;\; ; \\ x_3 - x_2 = -w \times sin(\alpha) \; ; \; y_3 - y_2 = w \times cos(\alpha) \;\; ; \\ x_0 - x_3 = -l \times cos(\alpha) \; ; \; y_0 - y_3 = -l \times sin(\alpha) \end{array} \tag{3}$$

These six[6] equations define the rectangular base of the parallelepiped, considering an orientation α and base dimensions w and l. As there are 12 variables and 10 equations (considering the first four from blob bounds), there are two degrees of freedom for this problem. In fact, posed this way, the problem defines a complex non-linear system, as sinusoidal functions are involved, and the indexes $j \in \{L, B, R, T\}$ for the set of bounded vertexes T are determined by the orientation α. Then, the wisest decision is to consider α as a known parameter. This way, the system becomes linear. But, there is still one degree of freedom. The best next choice must be a variable with known expected values, in order to be able to fix its value with a coherent quantity. Variables w, l and h comply with this requirement, as a pre-defined Gaussian model for each of these variables is available. The parallelepiped height h has been arbitrarily chosen for this purpose.
Therefore, the resolution of the system results in a set of linear relations in terms of h of the form presented in Equation (4). Just three expressions for w, l, and x_3 were derived from the resolution of the system, as the other variables can be determined from the four relations arising from the vertexes of the parallelepiped which are bounded by the 2D limits of the blob and the relations presented in Equation (3).

$$\begin{aligned} w &= M_w(\alpha; M, b) \times h + N_w(\alpha; M, b) \\ l &= M_l(\alpha; M, b) \times h + N_l(\alpha; M, b) \\ x_3 &= M_{x_3}(\alpha; M, b) \times h + N_{x_3}(\alpha; M, b) \end{aligned} \tag{4}$$

Therefore, considering perspective matrix M and 2D blob $b = (X_{left}, Y_{bottom}, X_{right}, Y_{top})$, a parallelepiped instance $S_{\mathbf{O}}$ for a detected object $\mathbf{O}$ can be completely defined as a function f:

$$S_{\mathbf{O}} = f(\alpha, h, M, b) \tag{5}$$

Equation (5) states that a parallelepiped model O can be determined with a function depending on parallelepiped height h, and orientation α, 2D blob b limits, and the calibration matrix M. The visual reliability measures remain to be determined and are described below.
The obtained solution states that the parallelepiped orientation α and height h must be known in order to calculate the parallelepiped. Taking these factors into consideration h and α are found for the optimal fit for each pre-defined parallelepiped class model, based on the probability measure PM defined in Equation (6)).

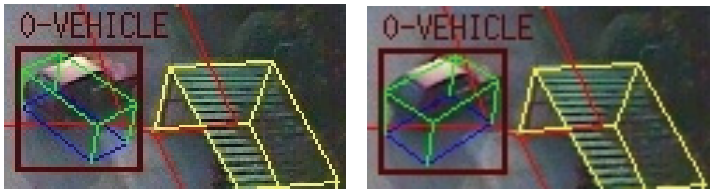
$$PM(S_{\mathbf{O}}, C) = \prod_{q \in \{w,l,h\}} Pr_{q_C}(q_{\mathbf{O}} | \mu_{q_C}, \sigma_{q_C}) \tag{6}$$

After finding the optimal model for each class based on PM, the class of the model with the highest PM value is considered as the class associated to the analysed 2D blob. This operation is performed for each blob on the current video frame.

3.1.1 Solving ambiguity of solutions

As the determination of a parallelepiped has been considered as an optimisation problem of only geometric features, this can lead to solutions far from the visual reality. A typical example is the one presented in Figure 4, where two solutions are very likely geometrically given the model, but the most likely from the expected model has the wrong orientation.

[6] In fact there are eight equations of this type. The two missing equations correspond to the relations between the variable pairs $(x_0; x_1)$ and $(y_0; y_1)$, but these equations are not independent. Hence, they have been suppressed.



(a) (b)

Fig. 4. Geometrically ambiguous solutions for the problem of associating a parallelepiped to a blob. Figure (a), shows an ambiguity between vehicle model instances, where the one with incorrect orientation has been chosen. In Figure (b), the correct solution to the problem.

A good way for discriminating between ambiguous situations is to return to pixel level. A simple solution is to store the most likely found parallelepipeds and to select the instance which better fits with the pixels inside the blob. This way, a moving pixel analysis is associated to the most likely parallelepiped instances by sampling the pixels enclosed by the blob and analysing if they fit the parallelepiped model instance. The sampling process is performed at a low pixel rate, adjusting this pixel rate to a pre-defined interval of sampled pixels number. True positives (TP), and true negatives (TN) are counted. A TP is considered as a moving pixel which is inside the 2D image projection of the parallelepiped, and TN as a background pixel outside the parallelepiped projection. Then, the chosen parallelepiped will be the one with higher $TP + TN$ value.

3.1.2 Dimensional reliability measures

A reliability measure R_q has been defined for each dimension $q \in \{w, l, h\}$ in the parallelepiped. This measure quantifies the visual evidence for the estimated dimension, by analysing how much of the dimension can be seen from the camera view. The measure gives a minimal value 0 when attribute is not visible, and a maximal value 1 when the attribute is totally visible. It is also influenced by static occlusion (image borders, static objects). The chosen function for modelling this reliability is $R_q \rightarrow [0, 1]$ (Equation (7)).

$$R_q = \min\left(\frac{dY_q \cdot Y_{occ}}{H} + \frac{dX_q \cdot X_{occ}}{W}, 1\right), \quad with \;\; q \in \{l, w, h\} \tag{7}$$

dX_q and dY_q represent the length in pixels of the projection of the dimension q on the X and Y reference axes of the image plane, respectively. H and W are the 2D height and width of the currently analysed 2D blob. Y_{occ} and X_{occ} are occlusion flags, which value is 0 if occlusion exists with respect to the Y or X reference axes of the image plane, respectively. These measures represent visual reliability as the sum of contributions of each 3D dimension projection onto the image axes, in proportion with the magnitude of each 2D blob limiting segment. Thus, the maximal value 1 is achieved if the the sum of the partial contributions for each 2D axis is higher than 1. The occlusion flags are used to eliminate the contribution to the reliability for a 2D axis projection in case of occlusion possibility in this axis direction.

3.2 Reliability multi hypothesis tracking

In this section, the new tracking algorithm, Reliability Multi-Hypothesis Tracking (RMHT), is described in detail. In general terms, this method presents similar ideas in the structure for creating, generating, and eliminating mobile object hypotheses compared to the MHT

methods presented in Section 2. The main differences from these methods are induced by the object representation utilised for tracking (section 3.1), and the dynamics model enriched by uncertainty control (section 3.2.1). The utilisation of region-based representations implies that several visual evidences could be associated to a mobile object (object parts). This consideration implies adapting the methods for creation and updating of object hypotheses. For further details on these adaptations, refer to (Zuniga, 2008).

3.2.1 Dynamics model

The dynamics model is the process for computing and updating the attributes of the mobile objects. Each mobile object in a hypothesis is represented as a set of statistics inferred from visual evidences of their presence in the scene. These visual evidences are stored in a short-term history buffer of blobs representing these evidences, called **blob buffer**. The attributes considered for the calculation of the mobile statistics belong to the set $A = \{X, Y, W, H, x_p, y_p, w, l, h, \alpha\}$. (X, Y) is the centroid position of the blob, W and H are the 2D blob width and height in image plane coordinates, respectively. (xp, yp) is the centroid position of the calculated 3D parallelepiped base. w, l, and h correspond to the 3D width, length, and height of the calculated parallelepiped in 3D scene coordinates. At the same time, an attribute V_a for each attribute $a \in A$ is calculated, representing the instant speed based on values estimated from visual evidence at different frames.

3.2.1.1 Modelling Uncertainty with reliability measures

Uncertainty on data can arise from many different sources. For instance, these sources can be the object model, the geometry of the scene, segmentation quality, temporal coherence, appearance, occlusion, among others. Following this idea, the proposed dynamics model integrates several reliability measures, representing different uncertainty sources.

Let RV_{a_k} be the **visual reliability** of the attribute a, extracted from the visual evidence observed at frame k. The visual reliability of an attribute RV_{a_k} changes according to the attribute. In the case of 3D dimensional attributes w, l, and h, these measures are obtained with the Equation (7). For 3D attributes x_p, y_p, and α, their visual reliability is calculated as the mean between the visual reliability of w and l, because the calculation of these three attributes is related to the base of the parallelepiped 3D representation. For 2D attributes W, H, X and Y a visual reliability measure inversely proportional to the distance to the camera is calculated, accounting for the fact that the segmentation error increases when objects are farther from the camera.

To account for the coherence of values obtained for attribute a throughout time, the **coherence reliability** measure $RC_a(t_c)$, updated to current time t_c, is defined:

$$RC_a(t_c) = 1.0 - min\left(1.0, \frac{\sigma_a(t_c)}{a_{max} - a_{min}}\right), \tag{8}$$

where values a_{max} and a_{min} in (8) correspond to pre-defined minimal and maximal values for a, respectively. The standard deviation $\sigma_a(t_c)$ of the attribute a at time t_c (incremental form) is defined as:

$$\sigma_a(t_c) = \sqrt{\hat{RV}(a) \cdot \left(\sigma_a(t_p)^2 + \frac{RV_{a_c} \cdot (a_c - \bar{a}(t_p))^2}{RVacc_a(t_c)}\right)}, \tag{9}$$

where a_c is the value of attribute a extracted from visual evidence at frame c, and $\bar{a}(t_p)$ (as later defined in Equation (14)) is the mean value of a, considering information until previous

frame p.

$$RVacc_a(t_c) = RV_{a_c} + e^{-\lambda \cdot (t_c - t_p)} \cdot RVacc_a(t_p), \tag{10}$$

is the **accumulated visual reliability**, adding current reliability RV_{a_c} to previously accumulated values $RVacc_a(t_p)$ weighted by a cooling function, and

$$\hat{RV}(a) = \frac{e^{-\lambda \cdot (t_c - t_p)} \cdot RVacc_a(t_p)}{RVacc_a(t_c)} \tag{11}$$

is defined as the ratio between current and previous accumulated visual reliability, weighted by a cooling function.

The value $e^{-\lambda \cdot (t_c - t_p)}$, present in Equations (10) and (11), and later in Equation (16), corresponds to the cooling function of the previously observed attribute values. It can be interpreted as a *forgetting factor* for reinforcing the information obtained from newer visual evidence. The parameter $\lambda \geq 0$ is used to control the strength of the forgetting factor. A value of $\lambda = 0$ represents a perfect memory, as forgetting factor value is always 1, regardless the time difference between frames, and it is used for attributes w, l, and h when the mobile is classified with a rigid model (i.e. a model of an object with only one posture (e.g. a car)).

Then, the **mean visual reliability measure** $\overline{RV}_a(t_k)$ represents the mean of visual reliability measures RV_a until frame k, and is defined using the accumulated visual reliability (Equation (10)) as

$$\overline{RV}_a(t_c) = \frac{RVacc_a(t_c)}{sumCooling(t_c)}, \tag{12}$$

with

$$sumCooling(t_c) = sumCooling(t_p) + e^{-\lambda \cdot (t_c - t_p)}, \tag{13}$$

where $sumCooling(t_c)$ is the accumulated sum of cooling function values.

In the same way, reliability measures can be calculated for the speed V_a of attribute a. Let V_{a_k} correspond to current instant velocity, extracted from the values of attribute a observed at video frames k and j, where j corresponds to the nearest valid previous frame index in time to k. Then, $RV_{V_{a_k}}$ corresponds to the visual reliability of the current instant velocity and is calculated as the mean between the visual reliabilities RV_{a_k} and RD_{a_j}.

3.2.1.2 Mathematical formulation of dynamics

The statistics associated to an attribute $a \in A$, similarly to the presented reliability measures, are calculated incrementally in order to have a better processing time performance, conforming a **new dynamics model** for tracked object attributes. This dynamics model proposes a new way of utilising reliability measures to weight the contribution of the new information provided by the visual evidence at the current image frame. The model also incorporates a cooling function utilised as a forgetting factor for reinforcing the information obtained from newer visual evidence.

Considering t_c as the time-stamp of the current frame c and t_p the time-stamp of the previous frame p, the obtained statistics for each mobile are now described. The **mean value** $\bar{a}$ for attribute a is defined as:

$$\bar{a}(t_c) = \frac{a_{exp}(t_c) \cdot R_{a_{exp}}(t_c) + a_{est}(t_c) \cdot R_{a_{est}}(t_c)}{R_{a_{exp}}(t_c) + R_{a_{est}}(t_c)}, \tag{14}$$

where the expected value a_{exp} corresponds to the expected value for attribute a at current time t_c, based on previous information, and a_{est} represents the value of a estimated from the

observed visual evidence associated to the mobile until current time t_c. These two values are intentionally related to respective **prediction** and **filtering** estimates of Kalman filters (Kalman, 1960). Their computation radically differs from these estimates by incorporating reliability measures and cooling functions to control pertinence of attribute data. $R_{a_{exp}}(t_c)$ and $R_{a_{est}}(t_c)$ correspond to reliability measures weighting the contributions of each of these elements.

The **expected value** a_{exp} of a corresponds to the value of a predictively obtained from the dynamics model. Given the **mean value** $\bar{a}(t_p)$ for a at the previous frame time t_p, and the estimated speed $V_a(t_p)$ of a at previous frame p, it is defined as

$$a_{exp}(t_c) = \bar{a}(t_p) + V_a(t_p) \cdot (t_c - t_p). \tag{15}$$

$V_a(t_c)$ corresponds to the estimated velocity of a (equation (17)) at current frame c.

The reliability measure $R_{a_{exp}}$ represents the reliability of the estimated value a_est of attribute a. It is determined as the mean of the **global reliabilities** R_a and R_{V_a} of a and V_a, respectively, at the previous time t_p. This way, the uncertainty of elements used for the calculation of a_{exp} as $\bar{a}(t_p)$ and $V_a(t_p)$, is utilised for modelling the uncertainty of a_{exp}. A **global reliability** measure $R_x(t_k)$ for an attribute x can be calculated as the mean between $R_{a_{exp}}$ and $R_{a_{est}}$ at t_k.

The **estimated value** a_{est} represents the value of a extracted from the observed visual evidence associated to the mobile, and is defined in Equation (16). This way, $a_{est}(t_c)$ value is updated by adding the value of the attribute for the current visual evidence, weighted by the visual reliability value for this attribute value, while previously obtained estimation is weighted by the forgetting factor.

$$a_{est}(t_c) = \frac{a_c \cdot RV_{a_c} + e^{-\lambda \cdot (t_c - t_p)} \cdot a_{est}(t_p) \cdot RVacc_a(t_p)}{RVacc_a(t_c)}, \tag{16}$$

where a_k is the value and RV_{a_k} is the visual reliability of the attribute a, extracted from the visual evidence observed at frame k. $RVacc_a(t_k)$ is the accumulated visual reliability until frame k, as described in Equation 10). $e^{-\lambda \cdot (t_c - t_p)}$ is the cooling function.

The reliability measure $R_{a_{est}}$ represents the reliability of the estimated value a_est of attribute a. It is calculated as the mean between the visual reliability $RV_a(t_c)$ (Equation (12)) and coherence reliability $RC_a(t_c)$ (Equation (8)) values at current frame c, weighted by the reliability measure R_{valid}. The R_{valid} reliability measure corresponds to the number of *valid* blobs in the blob buffer of the mobile over the size of the buffer. For a 2D attribute, a *valid* blob corresponds to a blob not corresponding to a lost object (no visual evidence correspondence), while for a 3D attribute, a *valid* blob corresponds to a blob which has been classified and has then valid 3D information. Not classified blobs correspond to blobs where the 3D classification method was not able to find a coherent 3D solution with respect to the current mobile attributes 3D information.

The statistics considered for velocity V_a follow the same idea of the previously defined equations for attribute a, with the difference that no expected value for the velocity of a is calculated, obtaining the value of the statistics of V_a directly from the visual evidence data. The velocity V_a of a is then defined as

$$V_a(t_c) = \frac{V_{a_c} \cdot RV_{V_{a_c}} + e^{-\lambda \cdot (t_c - t_p)} \cdot V_a(t_p) \cdot RVacc_{V_a}(t_p)}{RVacc_{V_a}(t_c)}, \tag{17}$$

where V_{a_k} corresponds to current instant velocity, extracted from the a attribute values observed at video frames k and j, where j corresponds to the nearest previous valid frame index previous to k. $RV_{V_{a_k}}$ corresponds to the visual reliability of the current instant velocity as defined in previous Section 3.2.1.1. Then, visual and coherence reliability measures for attribute V_a can be calculated in the same way as for any other attribute, as described in Section 3.2.1.1.

Finally, the **likelihood measure** p_m **for a mobile** m can be defined in many ways by combining the present attribute statistics. The chosen likelihood measure for p_m is a weighted mean of the probability measures for different group of attributes (group $\{w, l, h\}$ as D_{3D}, $\{x, y\}$ as V_{3D}, $\{W, L\}$ as D_{2D}, and $\{X, Y\}$ as V_{2D}), weighted by a joint reliability measure for each group, throughout the video sequence, as presented in Equation (18).

$$p_m = \frac{\sum_{k \in K} R_k C_k}{\sum_{k \in K} R_k} \tag{18}$$

with $K = \{D_{3D}, V_{3D}, D_{2D}, V_{2D}\}$ and

$$C_{D_{3D}} = \frac{\sum_{d \in \{w,l,h\}} (RC_d + P_d)\overline{RV}_d}{2 \sum_{d \in \{w,l,h\}} RD_d} \tag{19}$$

$$C_{V_{3D}} = \frac{MP_V + P_V + RC_V}{3.0}, \tag{20}$$

$$C_{D_{2D}} = R_{valid_{2D}} . \frac{RC_W + RC_H}{2}, \tag{21}$$

$$C_{V_{2D}} = R_{valid_{2D}} . \frac{RC_{V_X} + RC_{V_Y}}{2.0}, \tag{22}$$

where $R_{valid_{2D}}$ is the R_{valid} measure for 2D information, corresponding to the number of not *lost* blobs in the blob buffer, over the current blob buffer size. From equation (18), $R_{D_{2D}}$ is the mean between mean visual reliabilities $\overline{RV}_W(t_c)$ and $\overline{RV}_H(t_c)$, multiplied by $R_{valid_{2D}}$ measure. $R_{V_{2D}}$ is the mean between $\overline{RV}_X(t_c)$ and $\overline{RV}_Y(t_c)$, also multiplied by $R_{valid_{2D}}$ measure. $R_{D_{3D}}$ is the mean between $\overline{RV}_w(t_c)$, $\overline{RV}_l(t_c)$, and $\overline{RV}_h(t_c)$ for 3D dimensions w, l, and h, respectively, and multiplied by $R_{valid_{3D}}$ measure. $R_{valid_{3D}}$ is the R_{valid} measure for 3D information, corresponding to the number of not *classi□ed* blobs in the blob buffer, over the current blob buffer size. $R_{V_{3D}}$ is the mean between $\overline{RV}_x(t_c)$ and $\overline{RV}_y(t_c)$ for 3D coordinates x and y, also multiplied by $R_{valid_{3D}}$ measure. Measures $C_{D_{2D}}$, $C_{D_{3D}}$, $C_{V_{2D}}$, and $C_{V_{3D}}$ are considered as measures of temporal coherence (i.e. discrepancy between estimated and measured values) of the dimensional attributes (D_{2D} and D_{3D}) and the position velocities (V_{2D} and V_{3D}). The measures $R_{D_{3D}}$, $R_{V_{3D}}$, $R_{D_{2D}}$, and $R_{V_{2D}}$ are the accumulation of visibility measures in time (with decreasing factor).

P_w, P_l, and P_h in Equation (19) correspond to the mean probability of the dimensional attributes according to the a priori models of objects expected in the scene, considering the cooling function as in Equation (16). Note that parameter t_c has been removed for simplicity. MP_V, P_V, and RC_V values present in Equation (20) are inferred from attribute speeds V_x and V_y. MP_V represents the probability of the current velocity magnitude $V = \sqrt{V_x^2 + V_y^2}$

with respect to a pre-defined velocity model for the classified object, added to the expected object model, defined in the same way as described in Section 3.1. P_V corresponds to the mean probability for the position probabilities P_{V_x} and P_{V_y}, calculated with the values of P_w and P_l, as the 3D position is inferred from the base dimensions of the parallelepiped. RC_V corresponds to the mean between RC_{V_x} and RC_{V_y}.
This way, the value p_m for a mobile object m will mostly consider the probability values for attribute groups with higher reliability, using the values that can be trusted the most. At the same time, different aspects of uncertainty have been considered in order to better represent and identify several issues present in video analysis.

3.2.2 Hypothesis representation

In the context of tracking, a hypothesis corresponds to a set of mobile objects representing a possible configuration, given previously estimated object attributes (e.g. width, length, velocity) and new incoming visual evidence (blobs at current frame).
The representation of the tracking information corresponds to a *hypothesis set list* as seen in figure 5. Each *related hypothesis set* in the *hypothesis set list* represents a set of hypotheses exclusive between them, representing different alternatives for mobiles configurations temporally or visually related. Each hypothesis set can be treated as a different tracking sub-problem, as one of the ways of controlling the combinatorial explosion of mobile hypotheses. Each hypothesis has associated a likelihood measure, as seen in equation (23).

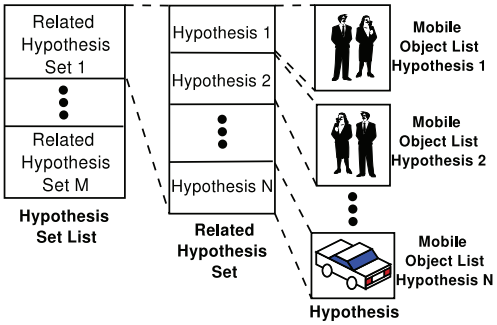


Fig. 5. Representation scheme utilised by our new tracking approach. The representation consists in a list of hypothesis sets. Each hypothesis set consists of hypotheses temporally or visually related. Each hypothesis corresponds to a set of mobile objects representing a possible objects configuration in the scene.

$$P_H = \sum_{i \in \Omega(H)} p_i \cdot T_i, \tag{23}$$

where $\Omega(H)$ corresponds to the set of mobiles represented in hypothesis H, p_i to the likelihood measure for a mobile i (as previously obtained from the dynamics model in Equation (18)), and T_i to a temporal reliability measure for a mobile i relative to hypothesis H, based on the life-time of the object in the scene.
Then, the likelihood measure P_H for an hypothesis H corresponds to the summation of the likelihood measures for each mobile object, weighted by a temporal reliability measure for each mobile, accounting for the life-time of each mobile. This reliability measure allows to

give higher likelihood to hypotheses containing objects validated for more time in the scene, and is defined in equation (24).

$$T_i = \frac{F_i}{\sum_{j \in \Omega(H)} F_j}. \tag{24}$$

This reliability measure intends to grant the survival of hypotheses containing objects of proved existence.

3.3 MILES: A new approach for incremental event learning and recognition

MILES is based on *incremental concept formation models* (Gennari et al., 1990). Conceptual clustering consists in describing classes by first generating their conceptual descriptions and then classifying the entities according to these descriptions. *Incremental concept formation models* is a conceptual clustering approach which incrementally creates a new concept without extensive reprocessing of the previously encountered instances. The knowledge is represented by a hierarchy of concepts partially ordered by generality. A *category utility* function is used to evaluate the quality of the obtained concept hierarchies (McKusick & Thompson, 1990).

MILES is an extension of incremental concept formation models for learning video events. The approach uses as input a set of attributes from the tracked objects in the scene. Hence, the only hypothesis of MILES is the availability of tracked object attributes (e.g. position, posture, class, speed). MILES constructs a **hierarchy of state and event concepts h**, based on the **state and event instances** extracted from the tracked object attributes.

A **state concept** is the model of a spatio-temporal property valid at a given instant or stable on a time interval. A **state concept** $S^{(c)}$, in a hierarchy **h**, is modelled as a **set of attribute models** $\{n_i\}$, with $i \in \{1, .., T\}$, where n_i is modelled as a random variable N_i which follows a Gaussian distribution $N_i \sim \mathcal{N}(\mu_{n_i}; \sigma_{n_i})$. T is the number of attributes of interest. The state concept $S^{(c)}$ is also described by its **number of occurrences** $N(S^{(c)})$, its **probability of occurrence** $\mathcal{P}(S^{(c)}) = N(S^{(c)})/N(S^{(p)})$ ($S^{(p)}$ is the root state concept of **h**), and the **number of event occurrences** $N_E(S^{(c)})$ (number of times that state $S^{(c)}$ passed to another state, generating an event).

A **state instance** is an instantiation of a state concept, associated to a tracked object **o**. The state instance $S^{(o)}$ is represented as the set attribute-value-measure triplets $\mathbf{T_o} = \{(v_i; V_i; R_i)\}$, with $i \in \{1, \ldots, T\}$, where R_i is the reliability measure associated to the obtained value V_i for the attribute v_i. The measure $R_i \in [0, 1]$ is 1 if associated data is totally reliable, and 0 if totally unreliable.

An **event concept** $E^{(c)}$ is defined as the change from a starting state concept $S_a^{(c)}$ to the arriving state concept $S_b^{(c)}$ in a hierarchy **h**. An **event concept** $E^{(c)}$ is described by its **number of occurrences** $N(E^{(c)})$, and its **probability of occurrence** $\mathcal{P}(E^{(c)}) = N(E^{(c)})/N_E(S_a^{(c)})$ (with $S_a^{(c)}$ its starting state concept).

The state concepts are hierarchically organised by generality, with the children of each state representing specifications of their parent. A unidirectional link between two state concepts corresponds to an event concept. An example of a hierarchy of states and events is presented in Figure 6. In the example, the state S_1 is a more general state concept than states $S_{1.1}$ and $S_{1.2}$, and so on. Each pair of state concepts $(S_{1.1}\ ;\ S_{1.2})$ and $(S_{3.2}\ ;\ S_{3.3})$, is linked by two events concepts, representing the occurrence of events in both directions.

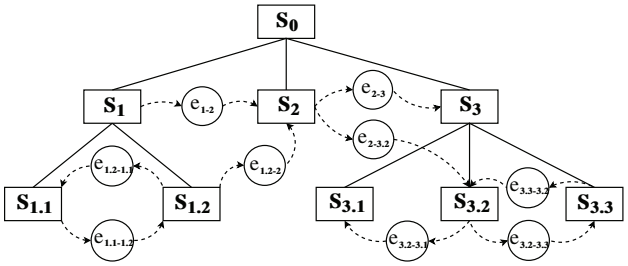


Fig. 6. Example of a hierarchical event structure resulting from the proposed event learning approach. Rectangles represent states, while circles represent events.

3.3.1 MILES learning process

The input of MILES corresponds to a list of tracked object attributes. MILES needs that the objects are tracked in order to detect the occurrence of *events*. There is no constraint on the number of attributes, as MILES has been conceived for learning state and event concepts in general. For each attribute, MILES needs a normalisation value to be defined prior to its computation. This value corresponds to the concept of *acuity*.

The **acuity** (Gennari et al., 1990) is a system parameter that specifies the minimal value for numerical attributes standard deviation σ in a state concept. In psycho-physics, the *acuity* corresponds to the notion of a *just noticeable difference*, the lower limit on the human perception ability. This concept is used for the same purpose in MILES, but the main difference with its utilisation in previous work (Gennari et al., 1990) is that the *acuity* was used as a single parameter, while in MILES each numerical attribute n_i has associated an acuity value A_{n_i}. This improvement allows to represent different normalisation scales and units associated to different attributes (e.g. kilo, meter, centimetre) and to represent the interest of users for different applications (more or less coarse precision). The acuity parameter needs to be set-up manually to enable the user to regulate the granularity of the earned states.

Initially, before the first execution of MILES, the hierarchy **h** is initialised as an empty tree. If MILES has been previously executed, the incremental nature of MILES learning process allows that the resulting hierarchy **h** can be utilised as the initial hierarchy of a new execution. At each video frame, MILES utilises the list of all tracked objects **O** for updating the hierarchy **h**. For each object **o** in **O**, MILES first gets the set of triplets $\mathbf{T_o}$, which serves as input for the state concept updating process of **h**. This updating process is described in Section 3.3.2. The updating process returns a list $\mathbf{L_o}$ of the current state concepts recognised for the object **o** at each level of **h**.

Then, the event concepts $E^{(c)}$ of the hierarchy **h** are updated comparing the new state concept list L_o with the list of state concepts recognised for the object **o** at the previous frame.

Finally, MILES gives as output for each video frame, the updated hierarchy **h** and the list of the currently recognised state and event concepts for each object **o** in **O**.

3.3.2 States updating algorithm

The hierarchy updating algorithm incorporates the new information at each level of the tree, starting from the root state.

The algorithm starts by accessing the analysed state **C** from the current hierarchy **h**. If the tree is empty, the initialisation of the hierarchy is performed by creating a state with the triplets $\mathbf{T_o}$, for the first processed object.

Then, for the case that **C** corresponds to a terminal state (the state has no children), a *cutoff* test is performed. The **cutoff** is a criteria utilised for stopping the creation (i.e. specialisation) of children states. It can be defined as:

$$\text{cutoff} = \begin{cases} \text{true} & \text{if} \quad \{ \quad \mu_{n_i} - V_{n_i} \leq A_{n_i} \\ & \qquad\qquad | \quad \forall\ i \in \{1,..,T\} \quad \} \\ \text{false} & \text{else} \end{cases}, \tag{25}$$

where V_{n_i} is the value of the i-th triplet of $\mathbf{T_o}$. This equation means that the learning process will stop at the concept state $S_k^{(c)}$ if no meaningful difference exists between each attribute value of $\mathbf{T_o}$ and the mean value μ_{n_i} of the attribute n_i for the state concept $S_k^{(c)}$ (based on the attribute *acuity* A_{n_i}).
If the *cutoff* test is passed, two children are generated for **C**, one initialised with $\mathbf{T_o}$ and the other as a copy of **C**. Then, passing or not passing the *cutoff* test, $\mathbf{T_o}$ is incorporated to the state **C** (state incorporation is described in Section 3.3.3). In this terminal state case, the updating process then stops.
If **C** has children, first $\mathbf{T_o}$ is immediately incorporated to **C**. Next, different new hierarchy configurations have to be evaluated among all the children of **C**. In order to determine in which state concept the triplets list $\mathbf{T_o}$ is next incorporated (i.e. the state concept is recognised), a quality measure for state concepts called **category utility** is utilised, which measures how well the instances are represented by a given category (i.e. state concept).
The category utility CU for a class partition of K state concepts (corresponding to a possible configuration of the children for the currently analysed state **C**) is defined as:

$$CU = \frac{\sum_{k=1}^{K} \frac{\mathcal{P}(S_k^{(c)}) \sum_{i=1}^{T} \left(\frac{A_{n_i}}{\sigma_{n_i}^{(k)}} - \frac{A_{n_i}}{\sigma_{n_i}^{(p)}} \right)}{2 \cdot T \cdot \sqrt{\pi}}}{K}, \tag{26}$$

where $\sigma_{n_i}^{(k)}$ (respectively for $\sigma_{n_i}^{(p)}$) is the standard deviation for the attribute n_i of $\mathbf{T_o}$, with $i \in \{1, 2, .., T\}$, in the state concept $S_k^{(c)}$ (respectively for the root state $S_p^{(c)}$).
It is worthy to note that the category utility CU serves as the major criteria to decide how to balance the states given the learning data. CU is an efficient criteria because it compares the relative frequency of the candidate states together with the relative Gaussian distribution of their attributes, weighted by their significant precision (predefined acuity).
Then, the different alternatives for the incorporation of $\mathbf{T_o}$ are:
(a) The incorporation of $\mathbf{T_o}$ to a existing state **P** gives the best CU score. In this case, the hierarchy updating algorithm is recursively called, considering **P** as root.
(b) The generation of a new state concept **Q** from instance $\mathbf{T_o}$ gives the best CU score **x**. In this case, the new state **Q** is inserted as child of **C**, and the updating process stops.
(c) Consider the state **M** as the resulting state from merging the best state **P** and the second best state **R**. Also, consider **y** as the CU score of replacing states **P** and **R** with **M**. If the best CU score is **y**, the hierarchy is modified by the **merge operator**. Then, the hierarchy updating algorithm is recursively called, using the sub-tree from state **M** as the tree to be analysed. The **merge operator** consists in merging two state concepts S_p and S_q into one state S_M, while S_p and S_q become the children of S_M, and the parent of S_p and S_q becomes the parent of S_M,

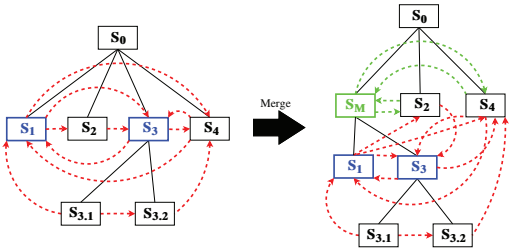


Fig. 7. Result of a merging operation. Blue boxes represent the states to be merged. The green box represents the resulting merged state. Red dashed lines represent the existing events, while the green dashed lines are the new events from the merging process.

as depicted in Figure 7. The merge operator also generates new events for state S_M which generalise the transitions incoming and leaving states S_p and S_q.
(d) Consider **z** as the CU score of replacing state **P** with its children. If the best CU score is **z**, the hierarchy is modified by the **split operator**. Then, the hierarchy updating algorithm is recursively called, using the sub-tree from the current state **C** again. The **split operator** consists in replacing a state S with its children, as depicted in Figure 8. This process implies to suppress the state concept S together with all the events in which the state is involved. Then, the children of the state S must be included as children of the parent state of S.

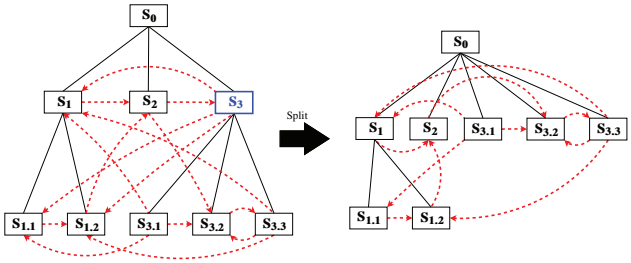


Fig. 8. Split operator in MILES approach. The blue box represents the state to be split. Red dashed lines represent events.

At the end of the hierarchy updating algorithm, each current state **C** for the different levels of the hierarchy is stored in the list **L** of current state concepts for object **o**.

3.3.3 Incorporation of new object attribute values

The incorporation process consists in updating a state concept with the triplets $\mathbf{T_o}$ for an object **o**. The proposed updating functions are incremental in order to improve the processing time performance of the approach. The incremental updating function for the mean value μ_n of an attribute n is presented in Equation (27).

$$\mu_n(t) = \frac{V_n \cdot R_n + \mu_n(t-1) \cdot Sum_n(t-1)}{Sum_n(t)}, \tag{27}$$

with

$$Sum_n(t) = R_n + Sum_n(t-1), \tag{28}$$

where V_n is the attribute value and R_n is the reliability. Sum_n is the accumulation of reliability values R_n.
The incremental updating function for the standard deviation σ_n for attribute n is presented in Equation (29).

$$\begin{aligned} &\sigma_n(t) = \sqrt{\frac{Sum_n(t-1)}{Sum_n(t)} \cdot \left(\sigma_n(t-1)^2 + \frac{R_n \cdot \Delta_n}{Sum_n(t)}\right)}. \\ &with \\ &\Delta_n = (V_n - \mu_n(t-1))^2 \end{aligned} \tag{29}$$

For a new state concept, the initial values taken for Equations (27), (28), and (29) with $t = 0$ correspond to $\mu_n(0) = V_n$, $Sum_n(0) = R_n$, and $\sigma_n(0) = A_n$, where A_n is the *acuity* for the attribute n.
In case that, after updating the standard deviation Equation (29), the value of $\sigma_n(i)$ is lower than the *acuity* A_n, $\sigma_n(i)$ is reassigned to A_n. This way, the acuity value establishes a lower bound for the standard deviation of an attribute.

4. Evaluation and results

4.1 Evaluating tracking

For evaluating the tracking approach, four benchmark videos publicly accessible have been evaluated. These videos are part of the evaluation framework proposed in ETISEO project (Nghiem et al., 2007). The obtained results have been compared with other algorithms which have participated in the ETISEO project. These four chosen videos are:

- **AP-11-C4:** Airport video of an apron (AP) with one person and four vehicles moving in the scene over 804 frames.
- **AP-11-C7:** Airport video of an apron (AP) with five vehicles moving in the scene over 804 frames.
- **RD-6-C7:** Video of a road (RD) with approximately 10 persons and 15 vehicles moving in the scene over 1200 frames.
- **BE-19-C1:** Video of a building entrance (BE) with three persons and one vehicle over 1025 frames.

The tests were performed with a computer with processor Intel Xeon CPU 3.00 GHz, with 2 Giga Bytes of memory. For obtaining the 3D model information, two parallelepiped models have been pre-defined for person and vehicle classes. The precision on 3D parallelepiped height values to search the classification solutions has been fixed in $0.08[m]$, while the precision on orientation angle has been fixed in $\pi/40[rad]$.

4.1.1 Results

The **Tracking Time** metric utilised in ETISEO project for evaluating object tracking has been used ($T_{Tracked}$ from now on). This metric measures the ratio of time that an object present in the reference data has been observed and tracked with a consistent ID over tracking period. The results using this metric are summarised in Figure 9.
The results are very competitive with respect to the other tracking approaches. Over 15 tracking results, the proposed approach has the second best result on the apron videos, and the third best result for the road video. The worst result for the proposed tracking approach has been obtained for the building entrance video, with a fifth position. For understanding these

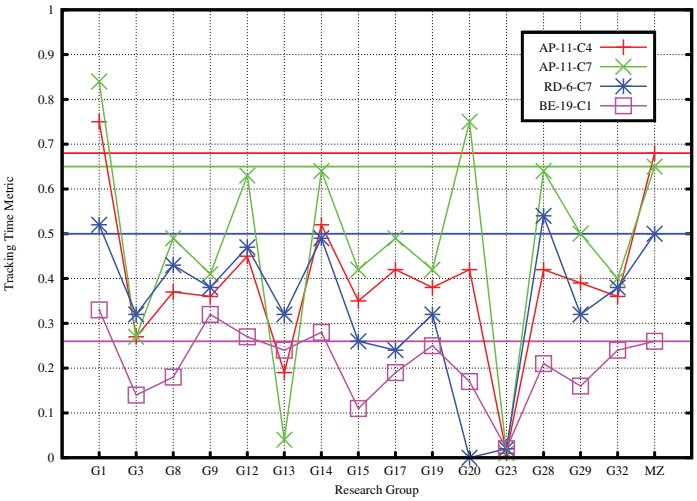


Fig. 9. Summary of results for the Tracking Time metric $T_{Tracked}$ for the four analysed videos. The labels at the horizontal axis represent the identifiers for anonymous research groups participating to the evaluation, except for the **MZ** label, which represents the proposed tracking approach. Horizontal lines at the level of the obtained results for the proposed approach have been added to help in the comparison of results with other research groups.

results it is worthy to analyse the videos separately. In further figures, the green bounding box enclosing an object represents the currently associated blob. The white bounding box enclosing a mobile corresponds to its 2D representation, while yellow lines correspond to its 3D parallelepiped representation. Red lines following the mobiles correspond to the 3D central points of the parallelepiped base found during the tracking process for the object. In the same way, blue lines following the mobiles correspond to the 2D representation centroids found. Images of these results are shown in Figure 10.

The processing time performance of the proposed tracking approach has been also analysed in this experiment. Unfortunately, ETISEO project has not incorporated the processing time performance as one of its evaluation metrics, thus it is not possible to compare the obtained results with the other tracking approaches. Table 1 summarises the obtained results for time metrics: mean processing time per frame $\overline{T_p}$, mean frame rate $\overline{F_p}$, standard deviation of the processing time per frame σ_{T_p}, and maximal processing time utilised in a frame $T_p^{(max)}$. The

Video	**Length**	$\overline{F_p}[frames/s]$	$\overline{T_p}[s]$	$\sigma_{T_p}[s]$	$T_p^{(max)}[s]$
AP-11-C4	804	76.4	0.013	0.013	0.17
AP-11-C7	804	85.5	0.012	0.027	0.29
RD-6-C7	1200	42.7	0.023	0.045	0.56
BE-19-C1	1025	86.1	0.012	0.014	0.15
Mean		**70.4**	**0.014**		

Table 1. Evaluation of results obtained for both analysed video clips in terms of processing time performance.

(a) AP-11-C7 (b) AP-11-C4

(c) RD-6-C7 (d) BE-19-C1

Fig. 10. Results for tracking experiment.

results show a high processing time performance, even for the road video **RD-6-C7** ($\overline{F_p} = 42.7[frames/sec]$), which concentrated several objects simultaneously moving in the scene. The fastest processing times for videos **AP-11-C7** ($\overline{F_p} = 85.5[frames/sec]$) and **BE-19-C1** ($\overline{F_p} = 86.1[frames/sec]$) are explained from the fact that there was a part of the video where no object was present in the scene, and because of the reduced number of objects. The high performance for the video **AP-11-C4** ($\overline{F_p} = 76.4[frames/sec]$) is because of the reduced number of objects. The maximal processing time for a frame $T_p^{(max)}$ is never greater than one second, and the $\overline{T_p}$ and σ_{T_p} metrics show that this maximal value can correspond to isolated cases.

The comparative analysis of the tracking approach has shown that the proposed algorithm can achieve a high performance in terms of quality of solutions for video scenes of moderated complexity. The results obtained by the algorithm are encouraging as they were always over the 69% of the total of research groups. It is important to consider that no system parameters reconfiguration has been made made between different tested videos, as one of the advantages on utilising a generic object model.

In terms of processing time performance, with a mean frame rate of $70.4[frames/s]$ and a frame rate of $42.7[frames/s]$ for the hardest video in terms of processing, it can be concluded that the proposed object tracking approach can have a real-time performance for video scenes of moderated complexity.

The road and building entrance videos show the need of new efforts on the resolution of harder static and dynamic occlusion problems. The interaction between the proposed parallelepiped model with appearance models can be an interesting first approach to analyse in the future for these cases. Nevertheless, appearance models are not useful in case of noisy data, bad contrast, or objects too far in the scene, but the general object model utilised in the proposed approach, together with a proper management of possible hypotheses, allows to better respond to these situations.

4.2 Evaluation of MILES

The capability of MILES for automatically learning and recognising real world situations has been evaluated, using two videos for elderly care at home. The video scene corresponds to an apartment with a table, a sofa, and a kitchen, as shown in Figure 11. The videos correspond to an elderly man (Figure 11(a)) and an elderly woman (Figure 11(b)), both performing tasks of everyday life as cooking, resting, and having lunch. The lengths of the sequences are 40000 frames (approximately 67 minutes) and 28000 frames (approximately 46 minutes).

The input information is obtained from a tracking method which computes reliability measures to object attributes, which is not included due to space constraints. The attributes

(a) (b)

Fig. 11. Video sequences for elderly care at home application. Figures (a) and (b) respectively show the observed elderly man and woman.

of interest for the evaluation are 3D position (x, y), an attribute for standing or crouching posture, and interaction attributes $SymD_{table}$, $SymD_{sofa}$, and $SymD_{kitchen}$ between the person and three objects present in the scene (table, sofa, and kitchen table). For simplicity, The interaction attributes are represented with three flags: $FAR : distance \geq 100[cm]$, $NEAR : 50[cm] < distance < 100[cm]$, and $VERY\_NEAR : distance \leq 50[cm]$. The contextual objects in the video scene (sofa, table, and kitchen) have been modelled in 3D.

All the attributes are automatically computed by a tracking method, which is able to compute the reliability measures of the attributes. These reliability measures account the quality and coherence of the acquired data.

The learning process applied over the 68000 frames have resulted in a hierarchy of 670 state concepts and 28884 event concepts. From the 670 states, 338 state concepts correspond to terminal states (50.4%). From the 28884 events, 1554 event concepts correspond to events occurring between terminal states (5.4%). This number of state and event concepts can be reduced considering a state stability parameter, defining the minimal duration for considering a state as stable.

This evaluation consists in comparing the recognised events with the ground-truth of a sequence. Different 750 frames from the elderly woman video are used for comparison,

corresponding to a duration of 1.33 minutes. The recognition process has obtained as result the events summarised in Figure 12.

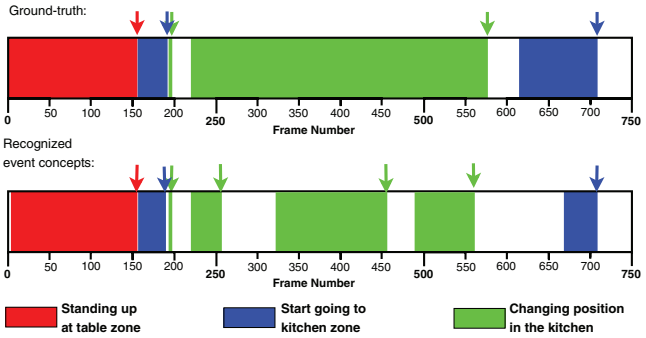


Fig. 12. Sequence of recognised events and ground-truth for the elderly woman video. The coloured arrows represent the events, while coloured zones represent the duration of a state before the occurrence of an event.

The evaluation has obtained 5 true positives (TP) and 2 false positives (FP) on event recognition. This results in a precision (TP/(TP+FP)) of 71%. MILES has been able to recognise all the events from the ground-truth, but also has recognised two nonexistent events, and has made a mean error on the starting state duration of 4 seconds. These errors are mostly due to bad segmentation near the kitchen zone, which had strong illumination changes, and to the similarity between the colours of the elderly woman legs and the floor. The results are encouraging considering the fact that the description of the sequence generated by a human has found a very close representation in the hierarchy.
The results show that the system is able to learn and recognise meaningful events occurring in the scene. The computer time performance of MILES is 1300$[frames/second]$ for a video with one tracked object and six attributes, showing the real-time capability of the learning approach. However, the learnt events are frequent and stable, but are not always meaningful for the user. Despite the calculation of the category utility, which formally measures the information density, an automatic process for measuring the usefulness of the learnt events for the user is still needed.

5. Conclusion

Addressing real world applications implies that a video analysis approach must be able to properly handle the information extracted from noisy videos. This requirement has been considered by proposing a generic mechanism to measure in a consistent way the reliability of the information in the whole video analysis process.
The proposed tracking method presents similar ideas in the structure of MHT methods. The main difference from these methods lies in the dynamics model, where features from different models (2D and 3D) are combined according to their reliability. This new dynamics model keeps redundant tracking of 2D and 3D object information, in order to increase robustness. This dynamics model integrates a reliability measure for each tracked object feature, which accounts for quality and coherence of utilised information. The calculation of this features considers a forgetting function (or cooling function) to reinforce the latest acquired information.

The reliability measures have been utilised to control the uncertainty in the obtained information, learning more robust object attributes and knowing which is the quality of the obtained information. These reliability measures have been also utilised in the event learning task of the video understanding framework to determine the most valuable information to be learnt.

The proposed tracking method has shown that is capable of achieving a high processing time performance for sequences of moderated complexity. But nothing can still be said for more complex situations. The results on object tracking have shown to be really competitive compared with other tracking approaches in benchmark videos, with a minimal reconfiguration effort. However, there is still work to do in refining the capability of the approach on coping with occlusion situations.

MILES algorithm allows to learn a model of the states and events occurring in the scene, when no a priori model is available. It has been conceived for learning state and event concepts in a general way. Depending on the availability of tracked object features, the possible combinations are large. MILES has shown its capability for recognising events, processing noisy image-level data with a minimal configuration effort. The proposed method computes the probability of transition between two states, similarly as HMM. The contribution MILES is to learn the global structure of the states and the events and to structure them in a hierarchy.

This work can be extended in several ways. Even if the proposed object representation serves for describing a large variety of objects, the result from the classification algorithm is a coarse description of the object. More detailed and class-specific object models could be utilised when needed, as articulated models, object contour, or appearance models. The proposed tracking approach is able to cope with dynamic occlusion situations where the occluding objects keep the coherence in the observed behaviour previous to the occlusion situation. Future work can point to the utilisation of appearance models utilised pertinently in these situations in order to identify which part of the visual evidence belongs to each object. The tracking approach could also be used in a feedback process with the motion segmentation phase in order to focus on zones where movement can occur, based on reliable mobile objects.

For the event learning approach, more evaluation is still needed for other type of scenes, for other attribute sets, and for different number and type of tracked objects. The anomaly detection capability of the approach on a large application must also be evaluated. Future work will be also focused in the incorporation of attributes related to interactions between tracked objects (e.g. meeting someone). The automatic association between the learnt events and semantic concepts and user defined events will be also studied.

6. References

Bar-Shalom, Y., Blackman, S. & Fitzgerald, R. J. (2007). The dimensionless score function for measurement to track association, *IEEE Transactions on Aerospace and Electronic Systems* 41(1): 392–400.

Blackman, S. (2004). Multiple hypothesis tracking for multiple target tracking, *IEEE Transactions on Aerospace and Electronic Systems* 19(1): 5–18.

Blackman, S., Dempster., R. & Reed, R. (2001). Demonstration of multiple hypothesis tracking (mht) practical real-time implementation feasibility, *in* . E. Drummond (ed.), *Signal and Data Processing of Small Targets*, Vol. 4473, SPIE Proceedings, pp. 470–475.

Boulay, B., Bremond, F. & Thonnat, M. (2006). Applying 3d human model in a posture recognition system, *Pattern Recognition Letter, Special Issue on vision for Crime Detection and Prevention* 27(15): 1788–1796.

Brémond, F. & Thonnat, M. (1998). Tracking multiple non-rigid objects in video sequences, *IEEE Transaction on Circuits and Systems for Video Technology Journal* 8(5).

Comaniciu, D., Ramesh, V. & Andmeer, P. (2003). Kernel-based object tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25: 564–575.

Cucchiara, R., Prati, A. & Vezzani, R. (2005). Posture classification in a multi-camera indoor environment, *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Vol. 1, Genova, Italy, pp. 725–728.

Gennari, J., Langley, P. & Fisher, D. (1990). Models of incremental concept formation, *in* J. Carbonell (ed.), *Machine Learning: Paradigms and Methods*, MIT Press, Cambridge, MA, pp. 11 – 61.

Ghahramani, Z. (1998). Learning dynamic bayesian networks, *Adaptive Processing of Sequences and Data Structures, International Summer School on Neural Networks*, Springer-Verlag, London, UK, pp. 168–197.

Heisele, B. (2000). Motion-based object detection and tracking in color image sequences, *Proceedings of the Fourth Asian Conference on Computer Vision (ACCV2000)*, Taipei, Taiwan, pp. 1028–1033.

Hu, W., Tan, T., Wang, L. & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors, *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 34(3): 334–352.

Hue, C., Cadre, J.-P. L. & Perez, P. (2002). Sequential monte carlo methods for multiple target tracking and data fusion, *IEEE Transactions on Signal Processing* 50(2): 309–325.

Isard, M. & Blake, A. (1998). Condensation - conditional density propagation for visual tracking, *International Journal of Computer Vision* 29(1): 5–28.

Kalman, R. (1960). A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* 82(1): 35–45.

Lavee, G., Rivlin, E. & Rudzsky, M. (2009). Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video, *SMC-C* 39(5): 489–504.

McIvor, A. (2000). Background subtraction techniques, *Proceedings of the Conference on Image and Vision Computing (IVCNZ 2000)*, Hamilton, New Zealand, pp. 147–153.

McKusick, K. & Thompson, K. (1990). Cobweb/3: A portable implementation, *Technical report*, Technical Report Number FIA-90-6-18-2, NASA Ames Research Center, Moffett Field, CA.

Moran, B. A., Leonard, J. J. & Chryssostomidis, C. (1997). Curved shape reconstruction using multiple hypothesis tracking, *IEEE Journal of Oceanic Engineering* 22(4): 625–638.

Nghiem, A.-T., Brémond, F., Thonnat, M. & Valentin, V. (2007). Etiseo, performance evaluation for video surveillance systems, *Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007)*, London (United Kingdom), pp. 476–481.

Nordlund, P. & Eklundh, J.-O. (1999). Real-time maintenance of figure-ground segmentation, *Proceedings of the First International Conference on Computer Vision Systems (ICVS'99)*, Vol. 1542 of *Lecture Notes in Computer Science*, Las Palmas, Gran Canaria, Spain, pp. 115–134.

Pattipati, K. R., Popp, R. L. & Kirubarajan, T. (2000). Survey of assignment techniques for multitarget tracking, *in* Y. Bar-Shalom & W. D. Blair (eds), *Multitarget-Multisensor Tracking: Advanced Applications, chapter 2*, Vol. 3, Artech House, Norwood, MA, pp. 77–159.

Piciarelli, C., Foresti, G. & Snidaro, L. (2005). Trajectory clustering and its applications for video surveillance, *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2005)*, IEEE Computer Society Press, Los Alamitos, CA, pp. 40–45.

Quack, T., Ferrari, V., Leibe, B. & Van Gool, L. (2007). Efficient mining of frequent and distinctive feature configurations, *International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brasil, pp. 1–8.

Rakdham, B., Tummala, M., Pace, P. E., Michael, J. B. & Pace, Z. P. (2007). Boost phase ballistic missile defense using multiple hypothesis tracking, *Proceedings of the IEEE International Conference on System of Systems Engineering (SoSE'07)*, San Antonio, TX, pp. 1–6.

Reid, D. B. (1979). An algorithm for tracking multiple targets, *IEEE Transactions on Automatic Control* 24(6): 843–854.

Scotti, G., Cuocolo, A., Coelho, C. & Marchesotti, L. (2005). A novel pedestrian classification algorithm for a high definition dual camera 360 degrees surveillance system, *Proceedings of the International Conference on Image Processing (ICIP 2005)*, Vol. 3, Genova, Italy, pp. 880–883.

Treetasanatavorn, S., Rauschenbach, U., Heuer, J. & Kaup, A. (July 2005). Model based segmentation of motion fields in compressed video sequences using partition projection and relaxation, *Proceedings of SPIE Visual Communications and Image Processing (VCIP)*, Vol. 5960, Beijing, China, pp. 111–120.

Xiang, T. & Gong, S. (2008). Video behavior profiling for anomaly detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(5): 893–908.

Yilmaz, A., Javed, O. & Shah, M. (2006). Object tracking: A survey, *ACM Computer Surveillance* 38(4). Article 13, 45 pages.

Yoneyama, A., Yeh, C. & Kuo, C.-C. (2005). Robust vehicle and traffic information extraction for highway surveillance, *EURASIP Journal on Applied Signal Processing* 2005(1): 2305–2321.

Zhao, T. & Nevatia, R. (2004). Tracking multiple humans in crowded environment, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04)*, Vol. 2, IEEE Computer Society, Washington, DC, USA, pp. 406–413.

Zuniga, M. (2008). *Incremental Learning of Events in Video using Reliable Information*, PhD thesis, Université de Nice Sophia Antipolis, École Doctorale STIC.