Real-time reliability measure-driven multihypothesis tracking using 2D and 3D features

Marcos Zuniga, Francois Bremond, Monique Thonnat

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Real-time reliability measure-driven multi-hypothesis tracking using 2D and 3D features

Marcos D Zúñiga[1*], François Brémond[2] and Monique Thonnat[2]

Abstract

We propose a new multi-target tracking approach, which is able to reliably track multiple objects even with poor segmentation results due to noisy environments. The approach takes advantage of a new dual object model combining 2D and 3D features through reliability measures. In order to obtain these 3D features, a new classifier associates an object class label to each moving region (e.g. person, vehicle), a parallelepiped model and visual reliability measures of its attributes. These reliability measures allow to properly weight the contribution of noisy, erroneous or false data in order to better maintain the integrity of the object dynamics model. Then, a new multi-target tracking algorithm uses these object descriptions to generate tracking hypotheses about the objects moving in the scene. This tracking approach is able to manage many-to-many visual target correspondences. For achieving this characteristic, the algorithm takes advantage of 3D models for merging dissociated visual evidence (moving regions) potentially corresponding to the same real object, according to previously obtained information. The tracking approach has been validated using video surveillance benchmarks publicly accessible. The obtained performance is real time and the results are competitive compared with other tracking algorithms, with minimal (or null) reconfiguration effort between different videos.

Keywords: multi-hypothesis tracking, reliability measures, object models

1 Introduction

Multi-target tracking is one of the most challenging problems in the domain of computer vision. It can be utilised in interesting applications with high impact in the society. For instance, in computer-assisted video surveillance applications, it can be utilised for filtering and sorting the scenes which can be interesting for a human operator. For example, SAMU-RAI European project [1] is focused on developing and integrating surveillance systems for monitoring activities of critical public infrastructure. Another interesting application domain is health-care monitoring. For example, GERHOME project for elderly care at home [2,3]) utilises heat, sound and door sensors, together with video cameras for monitoring elderly persons. Tracking is critical for the correct achievement of any further high-level analysis in video. In simple terms, tracking consists in assigning consistent labels to the tracked objects in different frames of a video [4], but it is also desirable for real-world applications that the extracted features in the process are reliable and meaningful for the description of the object invariants and the current object state and that these features are obtained in real time. Tracking presents several challenging issues as complex object motion, nonrigid or articulated nature of objects, partial and full object occlusions, complex object shapes, and the issues related to problems related to the multi-target tracking (MTT) problem. These tracking issues are major challenges in the vision community [5].

Following these directions, we propose a new method for real-time multi-target tracking (MTT) in video. This approach is based on multi-hypothesis tracking (MHT) approaches [6,7], extending their scope to multiple visual evidence-target associations, for representing an object observed as a set of parts in the image (e.g. due to poor motion segmentation or a complex scene). In order to properly represent uncertainty on data, an accurate dynamic model is proposed. This model utilises reliability measures, for modelling different aspects of the uncertainty. Proper representation of uncertainty,

* Correspondence: marcos.zuniga@usm.cl
[1]Electronics Department, Universidad Técnica Federico Santa María, Av. España 1680, Casilla 110-V, Valparaíso, Chile
Full list of author information is available at the end of the article

together with proper control over hypothesis generation, allows to reduce substantially the number of generated hypotheses, achieving stable tracks in real time for a moderate number of simultaneous moving objects. The proposed approach efficiently estimates the most likely tracking hypotheses in order to manage the complexity of the problem in real time, being able to merge dissociated visual evidence (moving regions or blobs), potentially corresponding to the same real object, according to previously obtained information. The approach combines 2D information of moving regions, together with 3D information from generic 3D object models, to generate a set of mobile object configuration hypotheses. These hypotheses are validated or rejected in time according to the information inferred in later frames combined with the information obtained from the currently analysed frame, and the reliability of this information.

The 3D information associated to the visual evidence in the scene is obtained based on generic parallelepiped models of the expected objects in the scene. At the same time, these models allow to perform object classification on the visual evidence. Visual reliability measures (confidence or degree of trust on a measurement) are associated to parallelepiped features (e.g. width, height) in order to account for the quality of analysed data. These reliability measures are combined with temporal reliability measures to make a proper selection of meaningful and pertinent information in order to select the most likely and reliable tracking hypotheses. Other beneficial characteristic of these measures is their capability to weight the contribution of noisy, erroneous or false data to better maintain the integrity of the object dynamics model. This article is focused on discussing in detail the proposed tracking approach, which has been previously introduced in [8] as a phase of an event learning approach. Therefore, the main contributions of the proposed tracking approach are:

- a new algorithm for tracking multiple objects in noisy environments,
- a new dynamics model driven by reliability measures for proper selection of valuable information extracted from noisy data and for representing erroneous and absent data,
- the improved capability of MHT to manage multiple visual evidence-target associations, and
- the combination of 2D image data with 3D information extracted using a generic classification model. This combination allows the approach to improve the description of objects present in the scene and to improve the computational performance by better filtering generated hypotheses.

This article is organised as follows. Section 2 presents related work. In Section 3, we present a detailed description of the proposed tracking approach. Next, Section 4 analyses the obtained results. Finally, Section 5 concludes and presents future work.

2 Related work

One of the first approaches focusing on MTT problem is the Multiple Hypothesis Tracking (MHT) algorithm [6], which maintains several correspondence hypotheses for each object at each frame. An iteration of MHT begins with a set of current track hypotheses. Each hypothesis is a collection of disjoint tracks. For each hypothesis, a prediction is made for each object state in the next frame. The predictions are then compared with the measurements on the current frame by evaluating a distance measure. MHT makes associations in a deterministic sense and exhaustively enumerates all possible associations. The final track of the object is the most likely hypothesis over the time period. The MHT algorithm is computationally exponential both in memory and time. Over more than 30 years, MHT approaches have evolved mostly on controlling this exponential growth of hypotheses [7,9-12]. For controlling this combinatorial explosion of hypotheses, all the unlikely hypotheses have to be eliminated at each frame. Several methods have been proposed to perform this task (for details refer to [9,13]). These methods can be classified in: screening [9], grouping methods for selectively generating hypotheses, and pruning, grouping methods for elimination of hypotheses after their generation.

MHT methods have been extensively used in radar (e.g. [14,15]) and sonar tracking systems (e.g. [16]). Figure 1 depicts an example of MHT application to radar systems [14]. In [17] a good summary of MHT applications is presented. However, most of these systems have been validated with simple situations (e.g. non-noisy data).

MHT is an approach oriented to single point target representation, so a target can be associated to just one measurement, not giving any insight on how can a set of measurements correspond to the same target, whether these measurements correspond to parts of the same target. Moreover, situations where a target separates into more than one track are not treated, then not considering the case where a tracked object corresponds to a group of visually overlapping set of objects [4]. When objects to track are represented as regions or multiple points, other issues must be addressed to properly perform tracking. For instance, in [18], authors propose a method for tracking multiple non-rigid objects. They define a target as an individually tracked moving region or as a group of moving regions globally tracked. To perform tracking, their approach performs a matching process, comparing the predicted location of targets

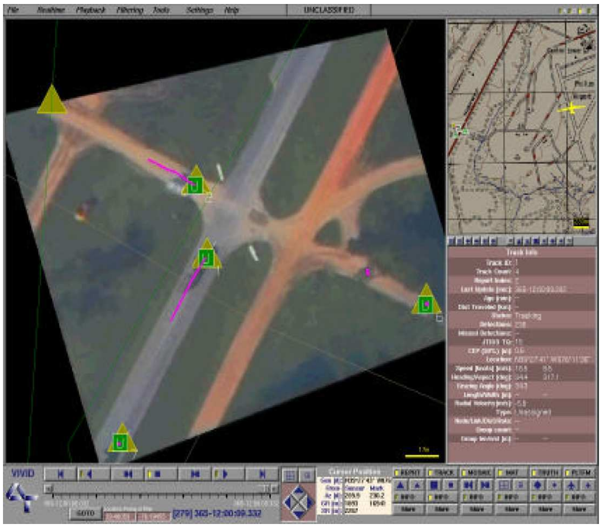


Figure 1 Example of a Multi-Hypothesis Tracking (MHT) application to radar systems [14]. This figure shows the tracking display and operator interface for real-time visualisation of the scene information. The yellow triangles indicate video measurement reports, the green squares indicate tracked objects and the purple lines indicate track trails.

with the location of newly detected moving regions through the use of an ambiguity distance matrix between targets and newly detected moving regions. In the case of an ambiguous correspondence, they define a compound target to freeze the associations between targets and moving regions until more accurate information is available. In this study, the used features (3D width and height) associated to moving regions often did not allow the proper discrimination of different configuration hypotheses. Then, in some situations, as badly segmented objects, the approach is not able to properly control the combinatorial explosion of hypotheses. Moreover, no information about the 3D shape of tracked objects was used, preventing the approach from taking advantage of this information to better control the number of hypotheses. Another example can be found in [19]. Authors use a set of ellipsoids to approximate the 3D shape of a human. They use a Bayesian multi-hypothesis framework to track humans in crowded scenes, considering colour-based features to improve their tracking results. Their approach presents good results in tracking several humans in a crowded scene, even in presence of partial occlusion. The processing time performance of their approach is reported as slower than frame rate. Moreover, their tracking approach is focused on tracking adult humans with slight variation in posture (just walking or standing). The improvement of associations in multi-target tracking, even for simple representations, is still considered a challenging subject, as in [20] where authors combine two boosting algorithms with object tracklets (track fragments), to improve the tracked objects association. As the authors focus on the association problem, the feature points are considered as already obtained, and no consideration is taken about noisy features.

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. For further details on classical dynamics models used in MHT, refer to [6,7,9-11,21]. The common features in the dynamics model of these algorithms are the utilisation of Kalman filtering [22] for estimation and prediction of object attributes.

An alternative to MHT methods is the class of Monte Carlo methods. These methods have widely spread into the literature as bootstrap filter [23], CONDENSATION (CONditional DENSity PropagATION) algorithm [24], Sequential Monte Carlo method (SMC) [25] and particle filter [26-28]. 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 and they do not scale properly for multiple objects present in the scene. 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). Then, it is appropriate to design object dynamics considering several measures modelling the different sources of uncertainty. In the literature, when dealing with the (single) object tracking problem, frequently authors tend to ignore the object initialisation problem assuming that the initial information can be set manually or that appearance of tracking target can be a priori learnt. Even new methods in object tracking, as MIL (Multiple Instance Learning) tracking by detection, make this assumption [29]. The problem of automatic object initialisation cannot be ignored for real-world applications, as it can pose challenging issues when the object appearance is not known, significantly changes with the object position relative to the camera and/or object orientation, or the analysed scene presents other difficulties to be dealt with (e.g. shadows, reflections, illumination changes, sensor noise). When interested in this kind of problem, it is necessary to consider the mechanisms to detect the arrival of new objects in the scene. This can be achieved in several ways. The most popular methods are based in background subtraction and object detection. Background subtraction methods extract motion from previously acquired information (e.g. background image or model) [30] and build object models from the foreground image. These models have to deal with noisy image frames, illumination changes, reflections, shadows and bad contrast, among other issues, but their computer performance is high. Object detection methods obtain an object model from training samples and then search occurrences of this model in new image frames [31]. This kind of approaches depend on the availability of training samples, are also sensitive to noise, are, in general, dependant on the object view point and orientation, and the processing time is still an issue, but they do not require a fixed camera to properly work.

The object representation is also 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 [32], ellipses [33]) 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 [34]) 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 [35], parallelepipeds [36]) 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 [37]. 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. The estimation of 3D features for different object classes posses a good challenge for a mono camera application, due to the fact that the projective transform poses an ill-posed problem (several possible solutions). Some works in this direction can be already found in the literature, as in [38], where the authors propose a simple planar 3D model, based on the 2D projection. To discriminate between vehicles and persons, they train a Support Vector Machine (SVM). The model is limited to this planar shape which is a really coarse representation, especially for vehicles and other postures of pedestrians. Also, they rely on a good segmentation as no treatment is done in case of several object parts, the approach is focused on single-object tracking, and the results in processing time and quality performance do not improve the state-of-the-art. The association of several moving regions to a same real object is still an open problem. But, for real-world applications it is necessary to address this problem in order to cope with situations related to disjointed object parts or occluding objects. Then, screening and pruning methods must be also adapted to these situations, in order to achieve performances adequate for real-world applications. Moreover, the dynamics models of multi-target tracking approaches do not handle properly noisy data. Therefore, the object features could be weighted according to their reliability to generate a new dynamics model which takes advantage able to cope with noisy, erroneous or missing data. Reliability measures have been used in the literature for focusing on the relevant information [39-41], allowing more robust processing. Nevertheless, these measures have been only used for specific tasks of the video understanding process. A generic mechanism is needed to compute in a consistent way the reliability measures of the whole video understanding process. In general, tracking algorithm implementations publicly available are hard to be found. A popular available implementation is a blob tracker, which is part of the OpenCV libraries [a], and is presented in [42]. The approach consists in a frame-to-frame blob tracker, with two components. A connected-component tracker when no dynamic occlusion occurs, and a tracker based on mean-shift [43] algorithms and particle filtering [44] when a collision occurs. They use a Kalman Filter for the dynamics model. The implementation is utilised for validation of the proposed approach.

3 Reliability-driven multi-target tracking

3.1 Overview of the approach

We propose a new multi-target tracking approach for handling several issues mentioned in Section 2. A scheme of the approach is shown in Figure 2. The tracking approach uses as input moving regions enclosed by a bounding box (blobs from now on) obtained from a previous image segmentation phase. More specifically, we apply a background subtraction method for

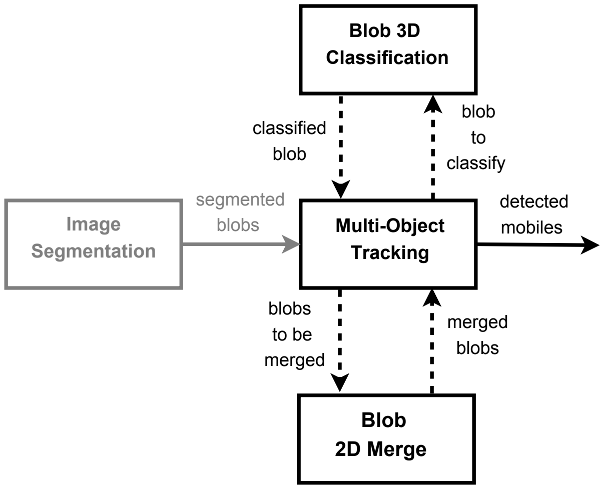


Figure 2 Proposed scheme for our new tracking approach.

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 study, we have considered a basic segmentation algorithm 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.

Using the set of blobs as input, the proposed tracking approach generates the hypotheses of tracked objects in the scene. The algorithm uses the blobs obtained in the current frame together with generic 3D models, to create or update hypotheses about 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 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 combines 2D information guided by 3D object models and the coherence of the previously tracked objects in the scene.

A blob 3D classification task is also utilised to obtain 3D information about the tracked objects, which allows to validate or reject hypotheses according to a priori information about the expected objects in the scene. The 3D classification method utilised in this study is discussed in the next section. Then, in section 3.3.1, the representation of the mobile hypotheses and the calculation of their attributes are presented. Finally, section 3.3.2 describes the proposed tracking algorithm, which encompasses all these elements.

3.2 Classification using 3D generic models

The tracking approach interacts with a 3D classification method which uses a generic parallelepiped 3D model of the expected objects in the scene. According to the best possible associations for previously tracked objects or testing a initial configuration for a new object, the tracking method sends a merged set of blobs to the 3D classification algorithm, in order to obtain the most likely 3D description of this blobs configuration, considering 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 3. For simplicity, lateral parallelepiped planes are considered perpendicular to top and bottom parallelepiped planes.

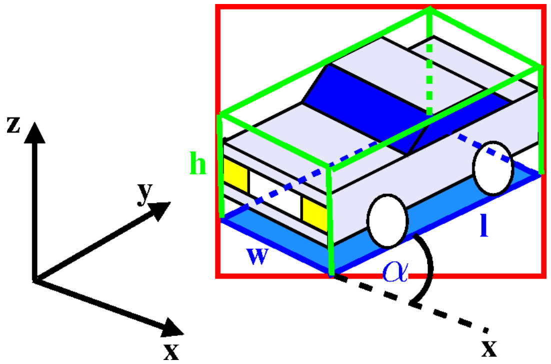


Figure 3 Example of a parallelepiped representation of an object. The figure depicts a vehicle enclosed by a 2D bounding box (coloured in red) and also by the parallelepiped representation. The base of the parallelepiped is coloured in blue and the lines projected in height are coloured in green. Note 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.

The proposed parallelepiped model representation allows to quickly determine the object class 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. 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, for faster computation. Formally, an attribute model $\tilde{\mathbf{q}}$, for an attribute q can be defined as:

$$\tilde{\mathbf{q}} = (Pr_{\mathrm{q}}(\mu_{\mathrm{q}}, \sigma_{\mathrm{q}}), q_{\min}, q_{\max}), \tag{1}$$

where Pr_q is a probability distribution described by its mean μ_q and its standard deviation σ_q, where $q \sim Pr_q$ $(\mu_{q,}\ \sigma_q)$. $q_{\min}$ and $q_{\max}$ represent the minimal and maximal values for the attribute q, respectively. Then, a pre-defined 3D parallelepiped model $Q_{\mathbf{C}}$ (a pre-defined model) for an object class $\mathbf{C}$ can be defined as:

$$Q_C = (\tilde{\mathbf{w}}, \tilde{\mathbf{l}}, \tilde{\mathbf{h}}), \tag{2}$$

where $\tilde{\mathbf{w}}$, $\tilde{\mathbf{l}}$ and $\tilde{\mathbf{h}}$ represent the attribute models for the 3D attributes width, length and height, respectively. The attributes w, l and h have been modelled as Gaussian probability distributions. 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 $S_{\mathbf{O}}$ (found while processing an image sequence) for an object O is described by:

$$S_{\mathbf{O}} = (\alpha,\ (w, R_w),\ (l, R_l),\ (h, R_h)), \tag{3}$$

where α represents the parallelepiped orientation angle, 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 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 3.2.5. This parallelepiped model has been first introduced in [45], and more deeply discussed in [8]. 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. The idea of this classification approach is to find a parallelepiped bounded by the limits of the 2D blob b. For completely determining the parallelepiped instance $S_{\mathbf{O}}$, 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 (x, y) of 3D coordinates representing the base coordinates of the vertexes. Therefore, a total of 12 variables have to be determined.

Considering that the 3D parallelepiped is bounded by the 2D bounding box found on a previous segmentation phase, we can use a pin-hole camera model transform to find four linear equations between the intersection of 3D vertex points and 2D bounds. Other six equations can be derived from the fact that the parallelepiped base points form a rectangle. As there are 12 variables and 10 equations, 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. Then, the wisest decision is to consider variable 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 10 equations previously discussed. For further details on the formulation of these equations, refer to [8].

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

Therefore, considering perspective matrix M and 2D blob $b = (X_{\text{left}}, Y_{\text{bottom}}, X_{\text{right}}, Y_{\text{top}})$, 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.

3.2.1 Classification method for parallelepiped model

The problem of finding a parallelepiped model instance $S_{\mathbf{O}}$ for an object $\mathbf{O}$, bounded by a blob b has been solved, as previously described. 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, a classification algorithm is proposed, which searches the optimal fit for each pre-defined parallelepiped class model, scanning different values of h and α. After finding optima for each class based on the probability measure PM (defined in Equation (6)), the method infers the class of the analysed blob also using the reliability measure PM. This operation is performed for each blob on the current video frame.

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

Given a perspective matrix **M**, object classification is performed for each blob *b* from the current frame as shown in Figure 4.

The presented algorithm corresponds to the basic optimisation procedure for obtaining the most likely parallelepiped given a blob as input. Several other issues have been considered in this classification approach, in order to cope with static occlusion, ambiguous solutions and objects changing postures. Next sections are dedicated to these issues.

3.2.2 Solving static occlusion

The problem of static occlusion occurs when a mobile object is occluded by the border of the image, or by a static object (e.g. couch, tree, desk, chair, wall and so on). In the proposed approach, static objects are manually modelled as a polygon base with a projected 3D height. On the other hand, the possibility of occlusion with the border of the image just depends on the proximity of a moving object to the border of the image. Then, the possibility of occurrence of this type of static occlusion can be determined based on 2D image information. To determine the possibility of occlusion by a static object present in scene is a more complicated task, as it becomes compulsory to interact with the 3D world.

In order to treat static occlusion situations, both possibilities of occlusion are determined in a stage prior to calculation of the 3D parallelepiped model. In case of occlusion, projection of objects can be bigger. Then, the limit of possible blob growth for the image referential directions left, bottom, right and top are determined, according to the position and shape of the possibly occluding elements (polygons) and the maximal dimensions of the expected objects in the scene (given different blob sizes). For example, if a blob has been detected very near the left limit of the image frame, then the blob could be bigger to the left, so its limit to the left is really bounded by the expected objects in the scene. For determining the possibility of occlusion by a static object, several tests are performed:

1. The 2D proximity to the static object 2D bounding box is evaluated,
2. if 2D proximity test is passed (object is near), the blob proximity to the 2D projection of the static object in the image plane is evaluated and
3. if the 2D projection test is also passed, the faces of the 3D polygonal shape are analysed, identifying the nearest faces to the blob. If some of these faces are hidden from the camera view, it is considered that the static object is possibly occluding the object enclosed by the blob. This process is performed in a similar way as [46].

```
**For each** class $C$ of pre-defined models
    **For all** valid pairs $(h, \alpha)$
        $S_O \leftarrow F(\alpha, h, M, b)$;
        **if** $PM(S_O, C)$ improves best current fit $S_O^{(C)}$ for $C$,
        **then** update optimal $S_O^{(C)}$ for $C$;
**Class**$(b) = argmax_C(PM(S_O^{(C)}, C))$;
```

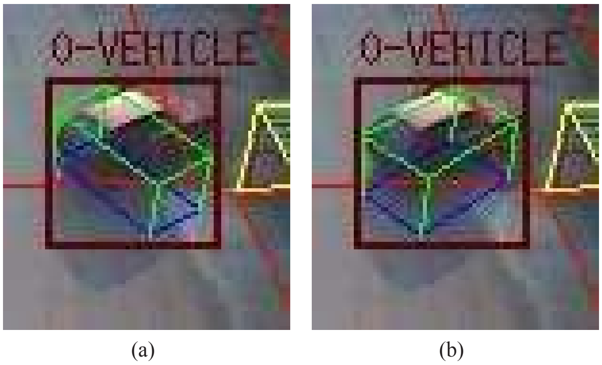
Figure 4 Classification algorithm for optimising the parallelepiped model instance associated to a blob.

When a possible occlusion exists, the maximal possible growth for the possibly occluded blob bounds is determined. First, in order to establish an initial limit for the possible blob bounds, the largest maximum dimensions of expected objects are considered at the blob position, and those who exceed the dimensions of the analysed blob are enlarged. If all possible largest expected objects do not impose a larger bound to the blob, the hypothesis of possible occlusion is discarded. Next, the obtained limits of growth for blob bounds are adjusted for static context objects, by analysing the hidden faces of the object polygon which possibly occlude the blob, and extending the blob, until its 3D ground projection collides the first hidden polygon face.

Finally, for each object class, the calculation of occluded parallelepipeds is performed by taking several starting points for extended blob bounds which represent the most likely configurations for a given expected object class. Configurations which pass the allowed limit of growth are immediately discarded and the remaining blob bound configurations are optimised locally with respect to the probability measure PM, defined in Equation (6), using the same algorithm presented in Figure 4. Notice that the definition of a general limit of growth for all possible occlusions for a blob allows to achieve an independence between the kind of static occlusion and the resolution of the static occlusion problem, obtaining the parallelepipeds describing the static object and border occlusion situations in the same way.

3.2.3 Solving ambiguity of solutions

As the determination of a parallelepiped to be associated to a blob has been considered as an optimisation problem of geometric features, several solutions can sometimes be likely, leading to undesirable solutions far from the visual reality. A typical example is the one presented in Figure 5, where two solutions are very likely geometrically given the model, but the most likely from the expected model has the wrong orientation.



(a) (b)

Figure 5 Geometrically ambiguous solutions for the problem of associating a parallelepiped to a blob. **(a)** An ambiguity between vehicle model instances, where the one with incorrect orientation has been chosen. **(b)** Correct solution to the problem.

A good way for discriminating between ambiguous situations is to return to moving pixel level. A simple solution is to store the most likely found parallelepiped configurations and to select the instance which better fits with the moving pixels found in the blob, instead of just choosing the most likely configuration. 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), false positives (FP), true negatives (TN) and false negatives (FN) are counted, considering a TP as a moving pixel which is inside the 2D image projection of the parallelepiped, a FP as a moving pixel outside the parallelepiped projection, a TN as a background pixel outside the parallelepiped projection and a FN as a background pixel inside the parallelepiped projection. Then, the chosen parallelepiped will be the one with higher TP + TN value.

Another type of ambiguity is related to the fact that a blob can be represented by different classes. Even if normally the probability measure PM (Equation (6)) will be able to discriminate which is the most likely object type, it exists also the possibility that visual evidence arising from overlapping objects give good PM values for bigger class models. This situation is normal as visual evidence can correspond to more than one mobile object hypothesis at the same time. The classification approach gives as output the most likely configuration, but it also stores the best result for each object class. This way, the decision on which object hypotheses are the real ones can be postponed to the object tracking task, where temporal coherence information can be utilised in order to chose the correct model for the detected object.

3.2.4 Coping with changing postures

Even if a parallelepiped is not the best suited representation for an object changing postures, it can be used for this purpose by modelling the postures of interest of an object. The way of representing these objects is to first define a general parallelepiped model enclosing every posture of interest for the object class, which can be utilised for discarding the object class for blobs too small or too big to contain it. Then, specific models for each posture of interest can be modelled, in the same way as the other modelled object classes. Then, these posture representations can be treated as any other object model. Each of these posture models are classified and the most likely posture information is associated to the object class. At the same time, the information for every analysed posture is stored in order to have the possibility of evaluating the coherence in time of an object changing postures by the tracking phase.

With all these previous considerations, the classification task has shown a good processing time performance. Several tests have been performed in a computer Intel Pentium IV, Xeon 3.0 GHz. These tests have been shown a performance of nearly 70 blobs/s, for four predefined object models, a precision for α of $\pi/40$ radians and a precision for h of 4 cm. These results are good considering that, in practice, classification is guided by tracking, achieving performances over 160 blobs/s.

3.2.5 Dimensional reliability measures

A reliability measure R_q for a dimension $q \in \{w, l, h\}$ is intended to quantify the visual evidence for the estimated dimension, by visually analysing how much of the dimension can be seen from the camera point of view. The chosen function is $R_q(S_\mathbf{O}) \to 0[1]$, where visual reliability of the attribute is 0 if the attribute is not visible and 1 if is completely visible. These measures represent visual reliability as the maximal magnitude of projection of a 3D dimension onto the image plane, in proportion with the magnitude of each 2D blob limiting segment. Thus, the maximal value 1 is achieved if the image projection of a 3D dimension has the same magnitude compared with one of the 2D blob segments. The function is defined in Equation (7).

$$R_a = \min\left(\frac{dY_\mathrm{a} \cdot Y_\mathrm{occ}}{H} + \frac{dX_\mathrm{a} \cdot X_\mathrm{occ}}{W}, 1\right), \tag{7}$$

where a stands for the concerned 3D dimension (l, w or h). dX_a and dY_a represent the length in pixels of the projection of the dimension a 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. The occlusion flags are used to eliminate the contribution to the value of the function of the projections in each 2D image reference axis in case of occlusion, as dimension is not visually reliable due to occlusion. An exception occurs in the case of a top view of an object, where reliability for h dimension is $R_h = 0$, because the dimension is occluded by the object itself.

These reliability measures are later used in the object tracking phase of the approach to weight the contribution of new attribute information.

3.3 Reliability multi-hypothesis tracking algorithm

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, the dynamics model incorporating reliability measures and the fact that this representation differs from the point representation (rather than region) frequently utilised in the MHT methods. The utilisation of region-based representations implies that several visual evidences could be associated to a mobile object (object parts). This consideration implies the conception of new methods for creation and update of object hypotheses.

3.3.1 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 6. 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 (8).

$$P_H = \sum_{i \in \Omega(H)} p_i \cdot T_i, \tag{8}$$

where $\Omega(H)$ corresponds to the set of mobiles represented in hypothesis H, p_i to the likelihood measure for a mobile i (obtained from the dynamics model (Section 3.4) in Equation (19)), 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 (9).

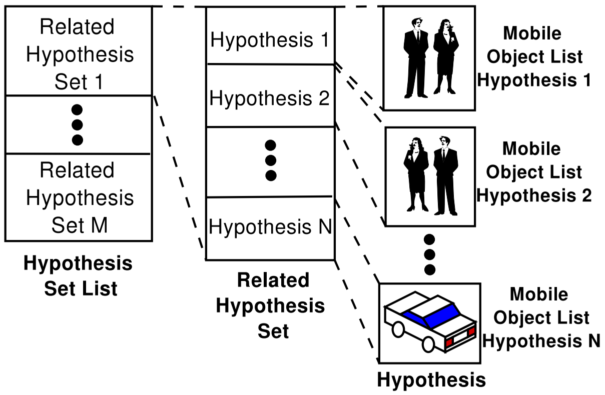


Figure 6 Representation scheme utilised by our new tracking approach. The representation consists in a list of hypotheses sets. Each hypotheses set consists in a set of hypotheses temporally or visually related. Each hypothesis corresponds to a set of mobile objects representing a possible objects configuration in the scene.

$$T_i = \frac{F_i}{\sum_{j \in \Omega(H)} F_j}, \tag{9}$$

where F_i is the number of frames since an object i has been seen for the first time. Then, this temporal measure lies between 0 and 1 too, as it is normalised by the sum of the number of frames of all the objects in hypothesis H.

3.3.2 Reliability tracking algorithm

The complete object tracking process is depicted in Figure 7. First, a hypothesis preparation phase is performed:

- It starts with a pre-merge task, which performs preliminary merge operations over blobs presenting highly unlikely initial features, reducing the number of blobs to be processed by the tracking procedure. This pre-merge process consist in first ordering blobs by proximity to the camera, and then merging blobs in this order, until minimal expected object model sizes are achieved. See Section 3.2, for further details on the expected object models.
- Then, the blob-to-mobile potential correspondences are calculated according to the proximity to the currently estimated mobile attributes to the blobs serving as visual evidence for the current

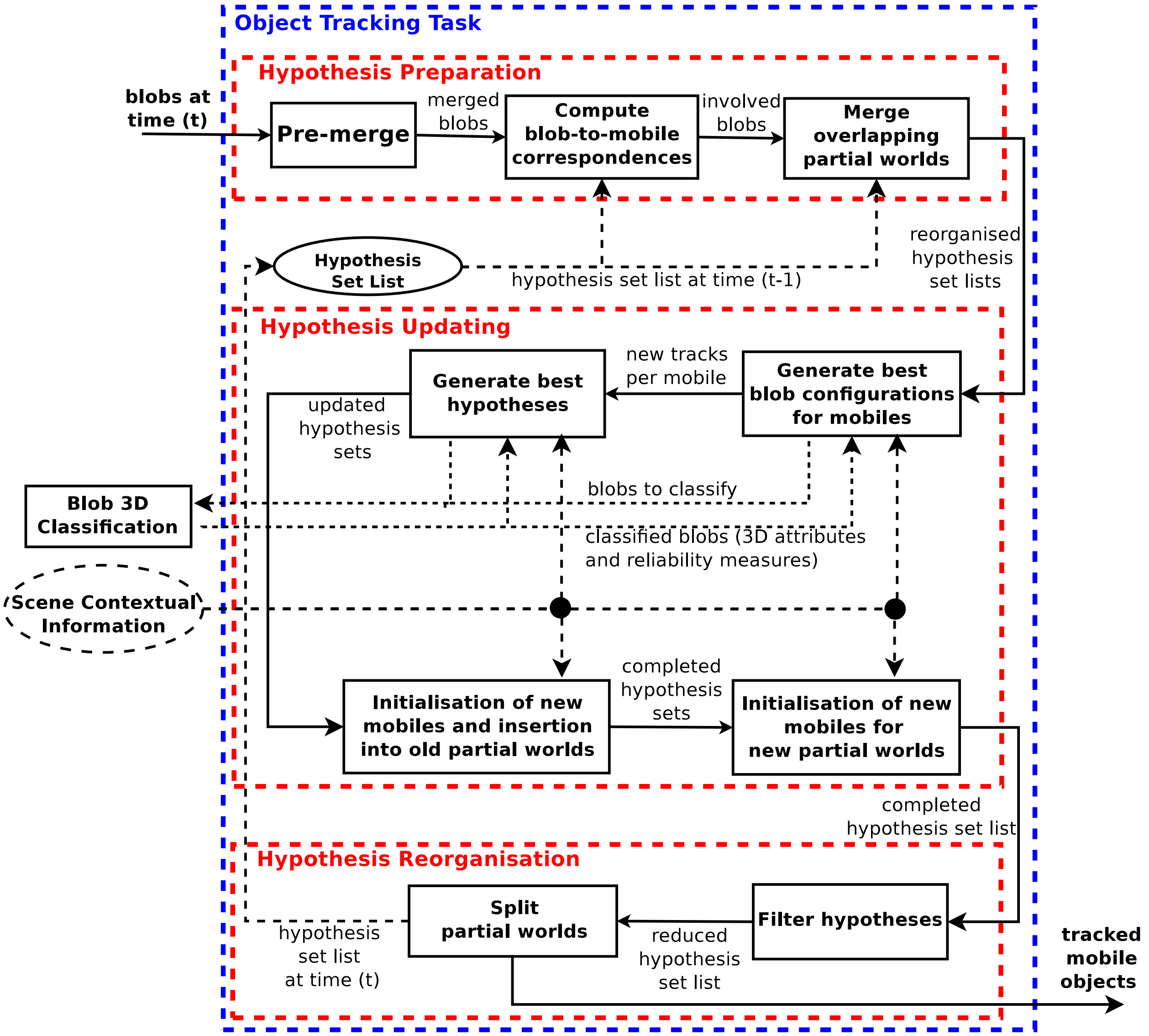

Object Tracking Task
Hypothesis Preparation
blobs at time (t)
Pre-merge
merged blobs
Compute blob-to-mobile correspondences
involved blobs
Merge overlapping partial worlds
reorganised hypothesis set lists
Hypothesis Set List
hypothesis set list at time (t-1)
Hypothesis Updating
Generate best hypotheses
new tracks per mobile
Generate best blob configurations for mobiles
updated hypothesis sets
Blob 3D Classification
blobs to classify
classified blobs (3D attributes and reliability measures)
Scene Contextual Information
completed hypothesis sets
Initialisation of new mobiles and insertion into old partial worlds
Initialisation of new mobiles for new partial worlds
completed hypothesis set list
Hypothesis Reorganisation
Split partial worlds
Filter hypotheses
hypothesis set list at time (t)
reduced hypothesis set list
tracked mobile objects

Figure 7 The proposed object tracking approach. The blue dashed line represents the limit of the tracking process. The red dashed lines represent the different phases of the tracking process.

frame. This set of blob potential correspondences associated to a mobile object is defined as the involved blob set which consists of the blobs that can be part of the visual evidence for the mobile in the current analysed frame. The involved blob sets allow to easily implement classical screening techniques, as described in Section 2.

- Finally, partial worlds (hypothesis sets) are merged if the objects at each hypothesis set are sharing a common set of involved blobs (visual evidence). This way, new object configurations are produced based on this shared visual evidence, which form a new hypothesis set.

Then, a hypothesis updating phase is performed:

- It starts with the generation of the new possible tracks for each mobile object present in each hypothesis. This process has been conceived to consider the immediate creation of the most likely tracks for each mobile object, instead of calculating all the possible tracks and then keeping the best solutions. It generates the initial solution which is nearest to the estimated mobile attributes, according to the available visual evidence, and then generates the other mobile track possibilities starting from this initial solution. This way, the generation is focused on optimising the processing time performance. In this process, different visual evidence sets are merged according to expected mobile size and position, and initially merged based on the models of the expected objects in the scene.

- Then, the obtained sets of most likely tracks are combined in order to obtain the most likely hypotheses representing the current alternatives for a partial world. The hypothesis generation process is oriented on looking for the most likely valid combinations, according to the observed visual evidence.
- After, new mobiles are initialised with the visual evidence not used by a given hypothesis, but utilised by other hypotheses sharing the same partial world. This way, all the hypotheses are complete in the sense that they provide a coherent description of the partial world they represent.
- In a similar way, visual evidence not related to any of the currently existing partial worlds is utilised to form new partial worlds according to the proximity of this new visual evidence.

A last phase of hypothesis reorganisation is then performed:

- First, mobiles definitely lost, and unlikely or redundant hypotheses are filtered (pruning process).
- Finally, a partial world can be separated, when the mobile objects in it are not currently related. This process reduces the number of generated hypotheses, as less mobile object configurations must be evaluated.

The most likely hypotheses are utilised to generate the list of most likely mobile objects which corresponds to the output of the tracking process.

3.3.3 3D classification and RMHT interactions

The best mobile tracks and hypothesis generation tasks interact with the 3D classification approach described in Section 3.2 in order to associate the 3D information for the most likely expected object classes associated to the mobiles. As reliability of mobile object attributes increases in time (becomes stable), the parallelepiped classification process can also be guided to boost the search of most likely parallelepiped configurations. This can be done by using the expected values of reliable 3D mobile attributes to give a starting point in the search of parallelepiped attributes, and optimising in a local neighbourhood of α and h parallelepiped attributes.

When a mobile object has validated its existence during several frames, even a better performance can be obtained by the 3D classification process, as the parallelepiped can be estimated just for one object class, assuming that the correct guess has been validated. In the other extreme, when information is still unreliable to perform 3D classification, only 2D mobile attributes are updated, as a way to avoid unnecessary computation of bad quality tentative mobiles.

3.4 Dynamics model

The dynamics model is the process of computing and updating the attributes of a mobile object, considering previous information and current observations. 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. In the case of the proposed object model combining 2D blob and 3D parallelepiped features, the attributes considered for the calculation of the mobile statistics belong to the set $A = \{X, Y, W, H, x_\mathrm{p}, y_\mathrm{p}, w, l, h, \alpha\}$. (X,Y) is the centroid position of the blob, W and H are the 2D blob width and height, respectively. (xp, yp) is the centroid position of the 3D parallelepiped base. w, l and h correspond to the 3D width, length and height of the parallelepiped. 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 available in the blob buffer. When the possibility of erroneous and lost data is considered, it is necessary to consider a blob buffer which can serve as backup information, as instant speed requires at least two available data instances.

3.4.1 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. Then, the design object dynamics must consider several measures for modelling these different sources. 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 differs according to the attribute.
- For the 3D attributes w, l and h, they 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.

- When no 3D model is found given a blob, the reliability measures for 3D attributes are set to 0. This allows to restrict the incorporation of attribute information to the dynamics model, if some attribute value is lost on the current frame.
- To account for the coherence of values obtained for attribute a throughout time, the coherence reliability measure $RC_{\mathrm{a}}(t_{\mathrm{c}})$, updated to current time t_{c}, is defined:

$$RC_{\mathrm{a}}(t_{\mathrm{c}}) = 1.0 - min\left(1.0, \frac{\sigma_{\mathrm{a}}(t_{\mathrm{c}})}{a_{\max} - a_{\min}}\right), \qquad (10)$$

where values $a_{\max}$ and $a_{\min}$ in (10) correspond to predefined minimal and maximal values for a, respectively. The standard deviation $\sigma_{\mathrm{a}}(t_{\mathrm{c}})$ of the attribute a at time t_{c} (incremental form) is defined as:

$$\sigma_{\mathrm{a}}(t_{\mathrm{c}}) = \sqrt{\widehat{RV}(a) \cdot \left(\sigma_{\mathrm{a}}(t_{\mathrm{p}})^2 + \frac{RV_{a_{\mathrm{c}}} \cdot (a_{\mathrm{c}} - \bar{a} - (t_{\mathrm{p}}))^2}{RVacc_{\mathrm{a}}(t_{\mathrm{c}})}\right)}, \qquad (11)$$

where a_{c} is the value of attribute a extracted from visual evidence at frame c, and $\bar{a}(t_{\mathrm{p}})$(as later defined in Equation (16)) is the mean value of a, considering information until previous frame p.

$$RVacc_{\mathrm{a}}(t_{\mathrm{c}}) = RV_{\mathrm{a_c}} + e^{-\lambda \cdot (t_{\mathrm{c}} - t_{\mathrm{p}})} \cdot RVacc_{\mathrm{a}}(t_{\mathrm{p}}), \qquad (12)$$

is the accumulated visual reliability, adding current reliability $RV_{\mathrm{a_c}}$ to previously accumulated values $RVacc_{\mathrm{a}}$ (t_{p}) weighted by a cooling function, and

$$\widehat{RV}(a) = \frac{e^{-\lambda \cdot (t_{\mathrm{c}} - t_{\mathrm{p}})} \cdot RVacc_{\mathrm{a}}(t_{\mathrm{p}})}{RVacc_{\mathrm{a}}(t_{\mathrm{c}})} \qquad (13)$$

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

The value $e^{-\lambda \cdot (t_{\mathrm{c}} - t_{\mathrm{p}})}$, present in Equations (11) and (12), 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 $\overline{RV}_{\mathrm{a}}(t_{\mathrm{k}})$ represents the mean of visual reliability measures RV_{a} until frame k, and is defined using the accumulated visual reliability (Equation (12)) as

$$\overline{RV}_{\mathrm{a}}(t_{\mathrm{c}}) = \frac{RVacc_{\mathrm{a}}(t_{\mathrm{c}})}{sumCooling(t_{\mathrm{c}})}, \qquad (14)$$

with

$$sumCooling(t_{\mathrm{c}}) = sumCooling(t_{\mathrm{p}}) + e^{-\lambda \cdot (t_{\mathrm{c}} - t_{\mathrm{p}})}, \qquad (15)$$

where $sumCooling(t_{\mathrm{c}})$ 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 $V_{\mathrm{a_k}}$ 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 to k. Then, $RV_{V_{\mathrm{a_k}}}$ corresponds to the visual reliability of the current instant velocity and is calculated as the mean between the visual reliabilities $RV_{\mathrm{a_k}}$and $RV_{\mathrm{a_j}}$.

3.4.2 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:

$$\bar{a}(t_{\mathrm{c}}) = \frac{a_{\mathrm{c}} \cdot RV_{\mathrm{a_c}} + e^{-\lambda \cdot (t_{\mathrm{c}} - t_{\mathrm{p}})} \cdot a_{\exp}(t_{\mathrm{p}}) \cdot RVacc_{\mathrm{a}}(t_{\mathrm{p}})}{RVacc_{\mathrm{a}}(t_{\mathrm{c}})}, \qquad (16)$$

where the expected value $a_{\exp}$ corresponds to the expected value for attribute a at current time t_c, based on previous information. This formulation is intentionally related to respective prediction and filtering estimates of Kalman filters [22]. This computation radically differs from the literature by incorporating reliability measures and a cooling function to control pertinence of attribute data. a_c is the value and $RV_{\mathrm{a_c}}$ is the visual reliability of the attribute a, extracted from the visual evidence observed at frame c. $RVacc_{\mathrm{a}}(t_{\mathrm{k}})$ is the accumulated visual reliability until a frame k, as described in

Equation (12). $e^{-\lambda\cdot(t_c-t_p)}$ is the cooling function. This way, $\bar{a}(t_c)$ value is updated by adding the value of the attribute for the current visual evidence, weighted by the visual reliability for this attribute value, while previously obtained estimation is weighted by the forgetting factor and by the accumulated visual reliability.

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

$$a_{\exp}(t_c) = \bar{a}(t_p) + V_a(t_p)\cdot(t_c - t_p). \tag{17}$$

$V_a(t_p)$ corresponds to the estimated velocity of a (Equation (18)) at previous frame p.

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

$$V_a(t_c) = \frac{V_{a_c}\cdot RV_{V_{a_c}} + e^{-\lambda\cdot(t_c-t_p)}\cdot V_a(t_p)\cdot RVacc_{V_a}(t_p)}{RVacc_{V_a}(t_c)}, \tag{18}$$

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 3.4.1. 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 3.4.1.

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 probability measures for different groups of attributes ($\{w, l, h\}$ 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, as presented in Equation (19).

$$p_m = \frac{\sum_{k\in K} R_k C_k}{\sum_{k\in K} R_k} \tag{19}$$

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

$$C_{D_{3D}} = \frac{\sum_{d\in\{w,l,h\}} (RC_d + P_d)\,\overline{RV}_d}{2\sum_{d\in\{w,l,h\}} RD_d} \tag{20}$$

$$C_{V_{3D}} = \frac{MP_V + P_V + RC_V}{3.0}, \tag{21}$$

$$C_{D_{2D}} = R_{\text{valid}_{2D}}\cdot\frac{RC_W + RC_H}{2}, \tag{22}$$

$$C_{V_{2D}} = R_{\text{valid}_{2D}}\cdot\frac{RC_{V_X} + RC_{V_Y}}{2.0}, \tag{23}$$

where $R_{\text{valid}_{2D}}$ 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 (19):

$$-R_{D_{2D}} = R_{\text{va1id}_{2D}}\frac{\overline{RV}_W(t_c)+\overline{RV}_H(t_c)}{2}$$

with $\overline{RV}_W(t_c)$ and $\overline{RV}_H(t_c)$ mean visual reliabilities of W and H, respectively.

$$-R_{V_{2D}} = R_{\text{va1id}_{2D}}\frac{\overline{RV}_X(t_c)+\overline{RV}_Y(t_c)}{2}$$

with $\overline{RV}_X(t_c)$ and $\overline{RV}_Y(t_c)$ mean visual reliabilities of X and Y, respectively.

$$-R_{D_{3D}} = R_{\text{va1id}_{3D}}\frac{\overline{RV}_w(t_c)+\overline{RV}_1(t_c)+\overline{RV}_h(t_c)}{3}$$

with $\overline{RV}_w(t_c)$, $\overline{RV}_1(t_c)$, and $\overline{RV}_h(t_c)$ the mean visual reliabilities for 3D dimensions w, l and h, respectively. $R_{\text{valid}_{3D}}$ corresponds to the number of classified blobs in the blob buffer, over the current blob buffer size.

$$-\ R_{V_{3D}} = R_{\text{va1id}_{3D}}\frac{\overline{RV}_x(t_c)+\overline{RV}_y(t_c)}{2}$$

with $\overline{RV}_x(t_c)$, and $\overline{RV}_y(t_c)$ the mean visual reliabilities for 3D position coordinates x, and y, respectively.

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). The measures $R_{V_{3D}}$, $R_{V_{3D}}$, $R_{D_{2D}}$ and $R_{V_{2D}}$ are the accumulation of visibility measures in time (with decreasing factor). P_w, P_l and P_h in Equation (20) 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 (21) 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, and defined in the same way as other attribute models described in Section 3.2. 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.

4 Evaluation and results

In order to validate the approach, two tests have been performed. The objective of the first test is to evaluate the performance of the proposed tracking approach in terms of quality of solutions. against the participants of the ETISEO project [47] for video analysis performance evaluation benchmarking. The obtained results have been compared with algorithms developed by 15 anonymous participants in the ETISEO project, considering four benchmark videos publicly available, which are part of the evaluation framework. The objective of the second test is to evaluate the performance of the proposed tracking approach in terms of quality of solutions and time performance, suppressing different features of the proposed approach, against a known tracker implementation. For this purpose, the performance of the proposed approach has been compared with the OpenCV frame-to-frame tracker [42] implementation, presented in Section 2. The same videos of the ETISEO project have been used for this test, considering versions with only 2D features and suppressing the reliability effect, in order to understand the contribution of each feature of the approach. Also a single-object video has been tested to give a closer insight of the effects of multi-target to object associations (poorly segmented objects). 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 Test I: Quality test against ETISEO participants

For evaluating the object tracking quality of the approach, the Tracking Time metric (T_{Tracked} from now on), utilised in ETISEO project, has been considered. 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 match between a reference datum *RD* and a physical object *C* is done with the bounding box distance D1 and with the constraint that object ID is constant over the time. The distance value D1 is defined in the context of ETISEO project as the dice coefficient, as twice the overlapping area between RD and C, divided by the sum of both the area of *RD* and *C* (Equation (24)).

$$D1 = \frac{2 \cdot area(RD \cap C)}{area(RD) + area(C)} \tag{24}$$

This matching process can give as result more than one candidate object *C* to be associated to a reference object *RD*. The chosen *C* candidate corresponds to the one with the greatest intersection time interval with the reference object *RD*. Then, the tracking time metric corresponds to the mean time during which a reference object is well tracked, as defined in Equation (25).

$$T_{\text{Tracked}} = \frac{1}{NB_{\text{RefData}}} \sum_{RefData} \frac{card(RD \cap C)}{card(RD)}, \tag{25}$$

where the function *card*() corresponds to the cardinality in terms of frames. From the available videos of the ETISEO project, the videos for evaluating the T_{Tracked} metric 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.

In terms of the Tracking Time metric, the results are summarised in Figure 8. The results are very competitive with respect to the other tracking approaches. For this experiment, 15 of the 22 participants of the real evaluation cycle have presented results for the Tracking Time metric [b]. Over these 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. In terms of reconfiguration between videos, the effort was minimal.

For understanding these results, it is worthy to analyse the videos separately. In further figures, a green

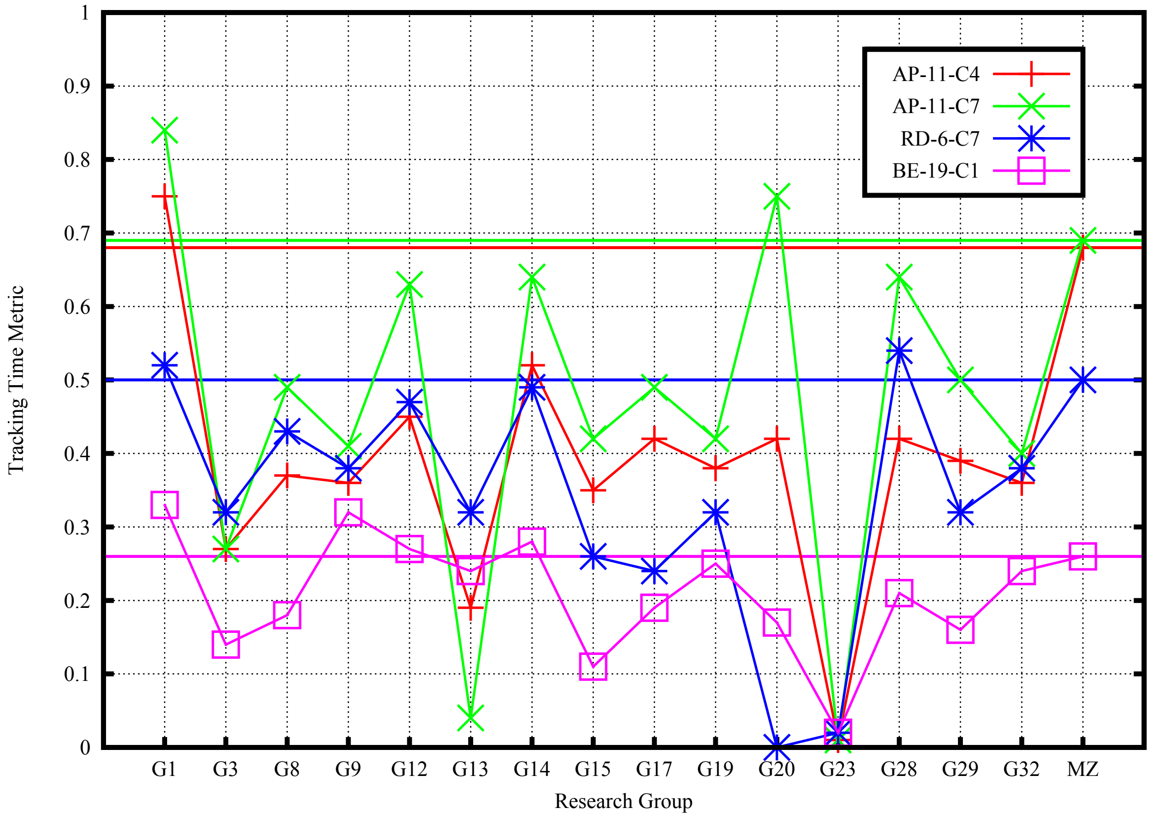


Figure 8 Summary of results for the Tracking Time metric T_{Tracked} for the four analysed videos. The labels starting with a **G**, at the horizontal axis, represent the identifiers for anonymous research groups participating on the evaluation, except for the **MZ** label, which represents the proposed 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.

bounding box enclosing an object means that the currently associated blob has been classified, while a red one means that the blob has not been classified. 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.

- AP-11-C4: For the first apron video, a Time Tracking metric value of 0.68 has been obtained. According to the appearance of the obtained results, it seemed that the metric value would be higher, as apparently no track has been lost over the analysis of the video. The metric value could have been affected by parts of the video where tracked objects became totally occluded until the end of the sequence. In this case, the tracking approach discarded these paths after certain number of frames. Results of the tracking process for this video are shown in Figure 9.

- AP-11-C7: For the second apron video, a Time Tracking metric value of 0.69 has been obtained. Similarly to the first video sequence, a higher metric value was expected, as apparently no track had been lost over the analysis of the video. The metric value could have been affected by the same reasons of video AP-11-C4. Results of the tracking process for this video are shown in Figure 10.

- RD-6-C7: For the road video, a Time Tracking metric value of 0.50 has been obtained. This video was hard compared with the apron videos. The main difficulties of this video were the total static occlusion situations at the bottom of the scene. At this position, the objects were often lost, because they were poorly segmented, and when the static occlusion situation occurred, no enough reliable information was available to keep their track, until they reappeared in the scene. Nevertheless, several objects were appropriately tracked and even the lost objects by static occlusion were correctly tracked after the

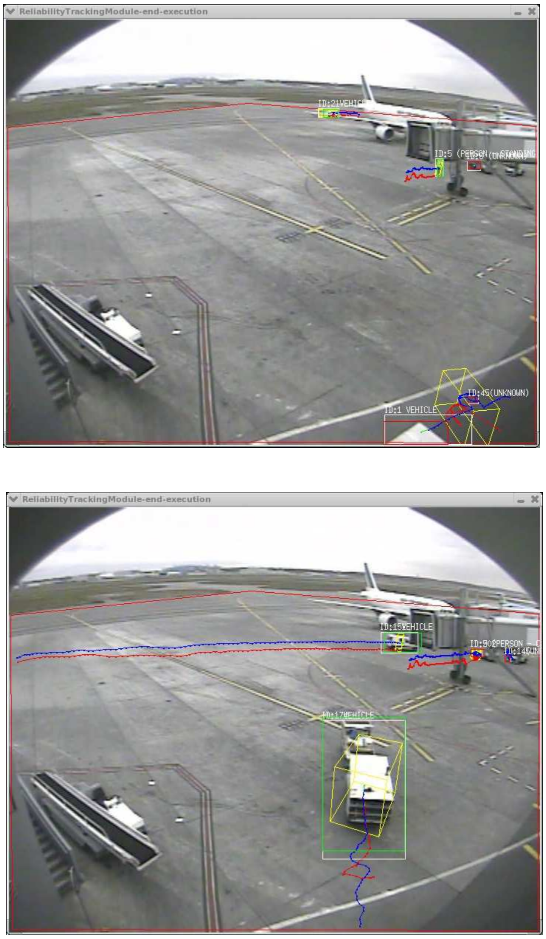

ReliabilityTrackingModule-end-execution
ID:21VEHICLE
ID:45(UNKNOWN)
ID:1 VEHICLE
ReliabilityTrackingModule-end-execution
ID:15VEHICLE
ID:17VEHICLE

Figure 9 Tracking results for the apron video AP-11-C4.

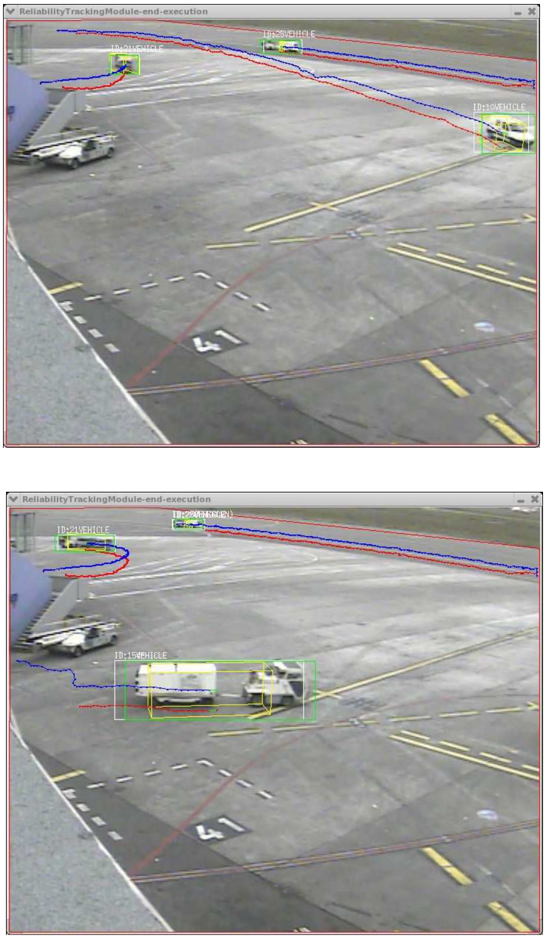

ReliabilityTrackingModule-end-execution
ID:26VEHICLE
ID:21VEHICLE
ID:10VEHICLE
ReliabilityTrackingModule-end-execution
ID:21VEHICLE
ID:15VEHICLE

Figure 10 Tracking results for the apron video AP-11-C7.

problem, showing a correct overall behaviour of the tracking approach. This video presented a real challenge for real-time processing as often nearly ten objects were tracked simultaneously. Results of the tracking process for this video are shown in Figure 11. A video with the tracking results is also publicly available [c].

- BE-19-C1: For the building entrance video, a Time Tracking metric value of 0.26 has been obtained. This video was the hardest of the four analysed videos, as presented dynamic occlusion situations and poor segmentation of the persons moving in the scene. Results of the tracking process for this video are shown in Figure 12.

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 $\overline{T_\text{p}}$, mean frame rate $\overline{F_\text{p}}$, standard deviation of the processing time per frame $\sigma_{\text{T}_\text{p}}$ and maximal processing time utilised in a frame $T_{\text{p}(\max)}$. The results show a high processing time performance, even for the road video RD-6-C7 ($\overline{F_\text{p}} = 42.7(\text{frames}/\text{s})$), which concentrated several objects simultaneously moving in the scene. The fastest processing times for videos AP-11-C7 ($\overline{F_\text{p}} = 85.5(\text{frames}/\text{s})$) and BE-19-C1 ($\overline{F_\text{p}} = 86.1(\text{frames}/\text{s})$) 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 ($\overline{F_\text{p}} = 76.4(\text{frames}/\text{s})$) 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

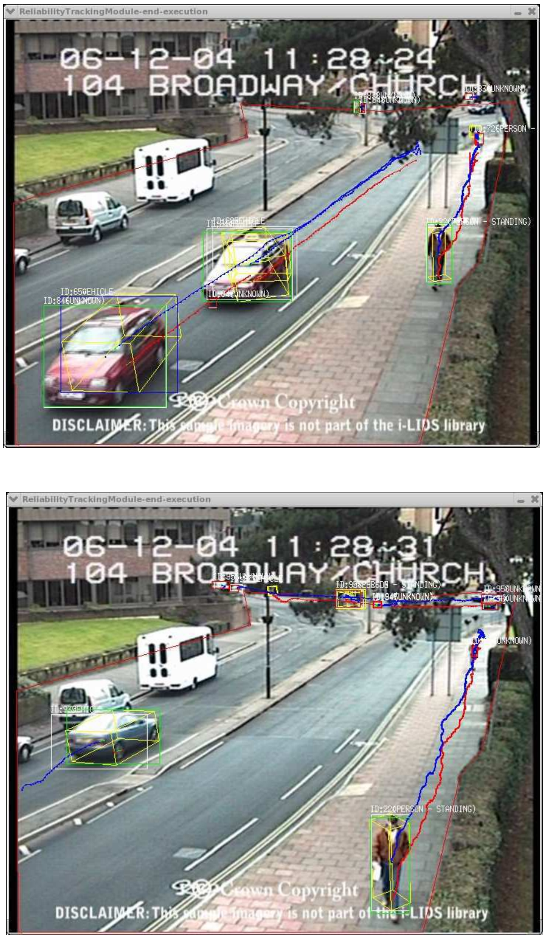

ReliabilityTrackingModule-end-execution
06-12-04 11:28:24
104 BROADWAY/CHURCH
ID:65VEHICLE
ID:84UNKNOWN)
Crown Copyright
DISCLAIMER: This sample imagery is not part of the i-LIDS library
ReliabilityTrackingModule-end-execution
06-12-04 11:28:31
104 BROADWAY/CHURCH
ID:2(PERSON - STANDING)
Crown Copyright
DISCLAIMER: This sample imagery is not part of the i-LIDS library

Figure 11 Tracking results for the road video RD-6-C7.

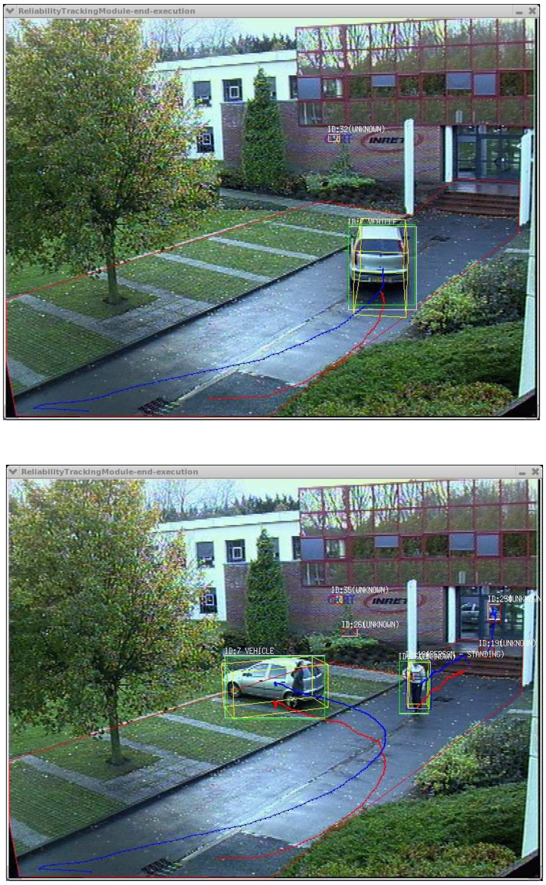


Figure 12 Tracking results for the building entrance video BE-19-C1.

the $\overline{T_p}$ and σ_{T_p} metrics show that this maximal value can correspond to isolated cases.

4.2 Test II: Testing different features of the approach

For this test, different algorithms are compared in order to understand the contribution of the different features:

- Tracker2D-R: A version of the proposed approach, suppressing 3D features.
- Tracker2D-NR: A version of the proposed approach, suppressing 3D features and reliability measures effect (every reliability measure set to 1).
- OpenCV-Tracker: The implementation of the OpenCV frame-to-frame tracker [42].

First, tests have been performed using the four ETISEO videos utilised in Section 4.1, evaluating the T_{Tracked} metric and the execution time performance. Tables 2 and 3 summarise T_{Tracked} metric and execution time performance, respectively. The videos of the results for Tracker2D-R, Tracker2D-NR and Tracker-OpenCV algorithms are available online at http://profesores.elo.utfsm.cl/~mzuniga/video.

According to the T_{Tracked} metric, the results show that the quality of tracking is greatly improved considering 3D features (see Table 2), and slightly improved considering reliability measures with only 2D features. It is worthy to highlight that the 3D features compulsory need the utilisation of reliability measures for representing not found 3D representations, occlusion and lost frames, among other issues. Even if utilising or not the reliability measures for only 2D features does not make a high difference in terms of quality, more complicated

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

Video	Length	$\overline{F_p}$(frames/s)	$\overline{T_p}$(s)	σ_{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 2 Quality evaluation using $T_{Tracked}$ metric, for different versions of the proposed approach and the OpenCV-Tracker.

Tracker	AP11-C4	AP11-C7	BE19-C1	RD6-C7
Tracker3D	0.68	0.69	0.26	0.50
Tracker2D-R	0.49	0.69	0.17	0.47
Tracker2D-NR	0.49	0.67	0.16	0.47
OpenCV-Tracker	0.41	0.65	0.12	0.48

scenes result in a better time performance when using reliability measures, as can be noticed in the time performance results for sequences RD6-C7 and BE19-C1, at Table 3. The Tracker3D highly outperforms OpenCV-Tracker in quality of solutions, obtaining these results with a higher time performance. Both Tracker2D versions outperform the quality performance of OpenCV-Tracker, while having a better time performance in almost an order of magnitude. Finally, in order to illustrate the difference in performance considering the multi-target to object association capability of the proposed approach, we have tested the Tracker2D-R version of the approach versus the OpenCV-Tracker on a single-object sequence of a rodent [d]. Figure 13 shows an illustration of the obtained sequence. The complete sequence is available online at http://profesores.elo.utfsm.cl/~mzuniga/video/VAT-HAMSTER.mp4. The sequences clearly show the smoothness achieved by the proposed approach in terms of object attributes estimation, compared with OpenCV-Tracker. This can be justified with the robustness of the dynamics model given by the cooling function, reliability measures and proper generation of multi-target to object hypotheses.

4.3 Discussion of results

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 and outperformed OpenCV-Tracker both in time and quality performance. It is important to consider that no system parameter reconfiguration has been 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 when 3D features are utilised, while the time performance using only 2D features is considerably higher, showing also a good quality/time compromise.

The road and building entrance videos have shown that there are still unsolved issues. Both 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.

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. More concretely, reliability measures associated to the object attributes have been proposed in order to measure the quality and coherence of this information. The proposed tracking method presents similar ideas in the structure for creating, generating and eliminating mobile object hypotheses compared to the MHT methods. The main differences from these methods are induced by the object representation utilised for tracking and the fact that this representation differs from the point representation normally utilised in the MHT methods. The utilisation of a representation different from a point representation implies the consideration of the possibility that several visual evidences could be associated to a mobile object. This consideration implies the conception

Table 3 Time performance evaluation for different versions of the proposed approach and the OpenCV-Tracker.

Tracker	AP11-C4		AP11-C7		BE19-C1		RD6-C7		Total Mean	
	(frame/s)	μ(s)	(frame/s)	μ(s)	(frame/s)	μ (s)	(frame/s)	μ (s)	(frame/s)	μ(s)
Tracker2D-R	434.8	0.0023	454.5	0.0022	208.3	0.0048	370.4	0.0027	358.2	0.0028
Tracker2D-NR	434.8	0.0023	454.5	0.0022	192.3	0.0052	333.3	0.003	342.3	0.0029
Tracker3D	76.4	0.013	85.5	0.012	86.1	0.012	42.7	0.023	70.4	0.0142
OpenCV-Tracker	57.1	0.0175	57.8	0.0173	41.5	0.0241	40.3	0.0248	47.8	0.0209

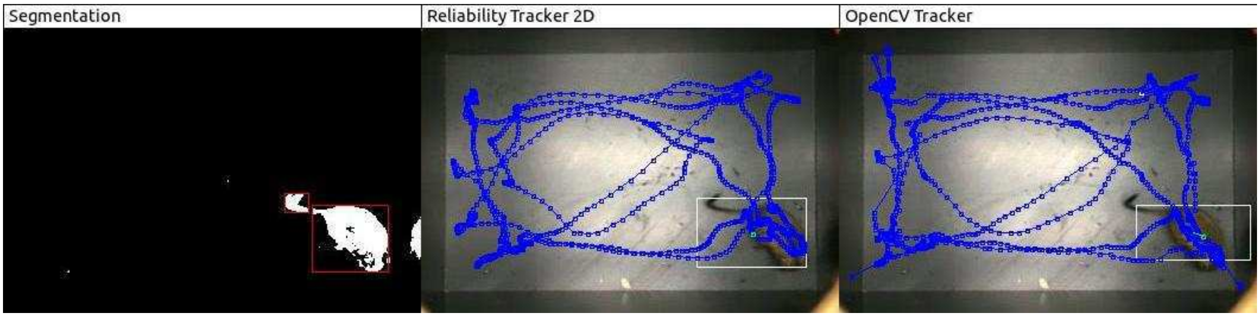


Figure 13 A frame of the HAMSTER sequence for illustrating the usefulness of the multi-target to object association capability. Left image shows a poor segmentation result. Note the green arrow at the end of the trajectory, which estimates the next position according to the dynamics model. Central image shows how the proposed approach handles the situation, while at the right image, the OpenCV-Tracker fails on properly managing the situation.

of new methods for creation and update of object hypotheses.

The tracking approach proposes a new dynamics model for object tracking which 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 are 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 are 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 approach has also shown its capability on solving static occlusion, sub-segmentation and object segmented by parts problems. Several features of the proposed tracking approach point to the objective of obtaining a processing time performance which could be considered as adequate for real-world applications: (a) explicit cooperation with the object classification process, by guiding the classification process using the previously learnt mobile object attributes, (b) the parallelepiped is estimated just for one object class if a mobile object class has proven to be reliable, (c) when mobiles are still unreliable, only 2D mobile attributes are updated as a way to avoid unnecessary computation of bad quality tentative mobiles, (d) the involved blob sets allow an easy implementation of gating and clustering techniques, (e) a hypothesis updating process oriented to optimise the estimation of the updated mobile tracks and hypothesis sets, in order to immediately obtain the most likely hypotheses, avoiding the generation of unlikely hypotheses (that must be eliminated later, anyway), (f) filtering redundant, not useful, or unlikely hypotheses and (g) the split process for hypothesis sets generating separated hypothesis sets, which can be treated as separated and simpler tracking sub-problems.

The results on object tracking have shown to be really competitive compared with other tracking approaches in benchmark videos. However, there is still work to do in refining the capability of the approach on coping with occlusion situations. 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.

Endnotes

[a]Documentation available at http://opencv.willowgarage.com/wiki/VideoSurveillance. [b]details about the results of the ETISEO participants are publicly accessible at http://www-sop.inria.fr/orion/ETISEO/download.htm, more specifically, obtained from the final results of ETISEO project after partners feedback at http://www-sop.inria.fr/orion/ETISEO/iso_album/

final_results_w_feedback.zip. [c]Obtained RD-6-C7 results can be observed in the video at

http://profesores.elo.utfsm.cl/~mzuniga/road.avi. [d]We would like to thank PhD. Adrian Palacios Research Lab, at the *Centro Interdisciplinario de Neurociencia* of Valparaiso, Universidad de Valparaiso, Chile, for providing us with the rodent video sequence.

Author details
[1]Electronics Department, Universidad Técnica Federico Santa María, Av. España 1680, Casilla 110-V, Valparaíso, Chile [2]Project-Team PULSAR - INRIA, 2004 route des Lucioles, Sophia Antipolis, France

Competing interests
The authors declare that they have no competing interests.

Received: 30 May 2011 Accepted: 30 December 2011
Published: 30 December 2011

References

1. W Zheng, S Gong, T Xiang, Quantifying contextual information for object detection. in *Proceedings of the IEEE International Conference on Computer Vision* (ICCV09), Kyoto, Japan 932–939 (2009)
2. GERHOME, Research Project. http://gerhome.cstb.fr (2005)
3. N Zouba, F Bremond, M Thonnat, VT Vu, Multi-sensors analysis for everyday elderly activity monitoring. in *Proceedings of the 4th International Conference SETIT'07: Sciences of Electronic, Technologies of Information and Telecommunications* Tunis, Tunisia 1–9 (2007)
4. A Yilmaz, O Javed, M Shah, Object tracking: A survey. ACM Comput Surveill. **38**(4), 45
5. IC Society (ed.), *IEEE International Series of Workshops on Performance Evaluation of Tracking and Surveillance (PETS)* (IEEE Computer Society, (2007) http://visualsurveillance.org
6. DB Reid, An algorithm for tracking multiple targets. IEEE Trans Autom Control. **24**(6), 843–854 (1979). doi:10.1109/TAC.1979.1102177
7. Y Bar-Shalom, S Blackman, RJ Fitzgerald, The dimensionless score function for measurement to track association. IEEE Trans Aerosp Electron Syst. **41**(1), 392–400 (2007)
8. M Zúñiga, F Brémond, M Thonnat, Uncertainty control for reliable video understanding on complex environments, in Video Surveillance, INTECH, 2011. Chap **21**, ed. by W Lin, 383–408 (2011)
9. T Kurien, Issues in the design of practical multitarget tracking algorithms, in *Multitarget-Multisensor Tracking: Advanced Applications chap 3*, vol. 1, ed. by Bar-Shalom Y (Artech House, Norwood, 1990), pp. 43–83
10. IJ Cox, JJ Leonard, Modeling a dynamic environment using a bayesian multiple hypothesis approach. Artif Intell. **66**(2), 311–344 (1994). doi:10.1016/0004-3702(94)90029-9
11. I Cox, S Hingorani, An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Trans Pattern Anal Mach Intell. **18**(2), 138–150 (1996). doi:10.1109/34.481539
12. S Blackman, R Dempster, R Reed, Demonstration of multiple hypothesis tracking (mht) practical real-time implementation feasibility, in *Signal and Data Processing of Small Targets*, vol. 4473, ed. by Drummond E (SPIE Proceedings, 2001), pp. 470–475
13. KR Pattipati, RL Popp, T Kirubarajan, Survey of assignment techniques for multitarget tracking, in *Multitarget-Multisensor Tracking: Advanced Applications chap 2*, vol. 3, ed. by Bar-Shalom Y, WD Blair (Artech House, Norwood, MA, 2000), pp. 77–159
14. PO Arambel, J Silver, J Krant, M Antone, T Strat, Multiple-hypothesis tracking of multiple ground targets from aerial video with dynamic sensor control. in *Signal Processing, Sensor Fusion, and Target Recognition XIII. Proceedings of the SPIE, vol 5429 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, ed. by Kadar I23–32 (2004)
15. B Rakdham, M Tummala, PE Pace, JB Michael, ZP Pace, Boost phase ballistic missile defense using multiple hypothesis tracking, in *Proceedings of the IEEE International Conference on System of Systems Engineering* (SoSE'07), (San Antonio, TX, 2007), pp. 1–6
16. BA Moran, JJ Leonard, C Chryssostomidis, Curved shape reconstruction using multiple hypothesis tracking. IEEE J Ocean Eng. **22**(4), 625–638 (1997). doi:10.1109/48.650829
17. S Blackman, Multiple hypothesis tracking for multiple target tracking. IEEE Trans Aerosp Electron Syst. **19**(1), 5–18 (2004)
18. F Brémond, M Thonnat, Tracking multiple non-rigid objects in video sequences. IEEE Trans Circuits Syst Video Technol J. **8**(5) (2004)
19. T Zhao, R Nevatia, Tracking multiple humans in crowded environment, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04)*, vol. 2. IEEE Computer Society, Washington, DC, pp. 406–413 (2004)
20. Y Li, C Huang, R Nevatia, Learning to associate: Hybridboosted multi-target tracker for crowded scene. in *CVPR* 2953–2960 (2009)
21. RL Streit, TE Luginbuhl, Maximum likelihood method for probabilistic multi-hypothesis tracking. in *Proceedings of the International Society for Optical Engineering (SPIE)*. **2235**, 394–405 (1994)
22. R Kalman, A new approach to linear filtering and prediction problems. J Basic Eng. **82**(1), 35–45 (1960). doi:10.1115/1.3662552
23. NJ Gordon, DJ Salmond, AFM Smith, Novel approach to nonlinear/non-gaussian bayesian state estimation. Radar Signal Process IEE Proc F. **140**(2), 107–113 (1993). doi:10.1049/ip-f-2.1993.0015
24. M Isard, A Blake, Condensation–conditional density propagation for visual tracking. Int J Comput Vis. **29**(1), 5–28 (1998). doi:10.1023/A:1008078328650
25. Doucet A, de Freitas N, Gordon N (eds.), *Sequential Monte Carlo Methods in Practice* (Springer-Verlag, 2001)
26. C Hue, J-PL Cadre, P Perez, Sequential monte carlo methods for multiple target tracking and data fusion. IEEE Trans. Signal Process. **50**(2), 309–325 (2002)
27. C Hue, J-PL Cadre, P Perez, Tracking multiple objects with particle filtering. IEEE Trans Aerosp Electron Syst. **38**(3), 791–812 (2002). doi:10.1109/TAES.2002.1039400
28. Y Jin, F Mokhtarian, Variational particle filter for multi-object tracking, in *International Conference on Computer Vision (ICCV'07)*, Rio de Janeiro, Brasil, pp. 1–8 (2007)
29. B Babenko, M-H Yang, S Belongie, Visual tracking with online multiple instance learning. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR09)* 983–990 (2009)
30. A McIvor, Background subtraction techniques, in *Proceedings of the Conference on Image and Vision Computing (IVCNZ 2000)*, Hamilton, New Zealand, pp. 147–153 (2000)
31. E Seemann, B Leibe, B Schiele, Multi-aspect detection of articulated objects, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE Computer Society, Washington DC, pp. 1582–1588 (2006)
32. R Cucchiara, A Prati, R Vezzani, Posture classification in a multi-camera indoor environment. in *Proceedings of IEEE International Conference on Image Processing (ICIP)* Genova, Italy. **1**, 725–728 (2005)
33. D Comaniciu, V Ramesh, P Andmeer, Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell. **25**, 564–575 (2003). doi:10.1109/TPAMI.2003.1195991
34. B Boulay, F Bremond, M Thonnat, Applying 3D human model in a posture recognition system, pattern recognition letter. Special Issue vis Crime Detect Prev. **27**(15), 1788–1796 (2006)
35. G Scotti, A Cuocolo, C Coelho, L Marchesotti, A novel pedestrian classification algorithm for a high definition dual camera 360 degrees surveillance system. in *Proceedings of the International Conference on Image Processing (ICIP 2005)* Genova, Italy. **3**, 880–883 (2005)
36. A Yoneyama, C Yeh, C-C Kuo, Robust vehicle and traffic information extraction for highway surveillance. EURASIP J Appl Signal Process. **2005**(1), 2305–2321 (2005). doi:10.1155/ASP.2005.2305
37. T Quack, V Ferrari, B Leibe, L Van Gool, Efficient mining of frequent and distinctive feature configurations. in *International Conference on Computer Vision* (ICCV 2007) (Rio de Janeiro, Brasil) 1–8 (2007)
38. P-H Lee, T-H Chiu, Y-L Lin, Y-P Hung, Real-time pedestrian and vehicle detection in video using 3d cues. in *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, ICME'09 (IEEE Press, Piscataway 614–617 (2009)
39. L Torresani, A Hertzmann, C Bregler, Learning non-rigid 3d shape from 2d motion, in *Advances in Neural Information Processing Systems*, vol. 16, ed. by Thrun S, Saul L, Schölkopf B (MIT Press, Cambridge, 2004)

40. S Treetasanatavorn, U Rauschenbach, J Heuer, A Kaup, Bayesian method for motion segmentation and tracking in compressed videos, in *DAGM-Symposium, vol. 3663 of Lecture Notes in Computer Science (LNCS) on Pattern Recognition and Image Processing*, ed. by Kropatsch WG, Sablatnig R, Hanbury A (Springer, 2005), pp. 277–284
41. E Erzin, Y Yemez, AM Tekalp, A Ercil, H Erdogan, H Abut, Multimodal person recognition for human-vehicle interaction. IEEE MultiMedia. **13**(2), 18–31 (2006). doi:10.1109/MMUL.2006.37
42. T Chen, H Haussecker, A Bovyrin, R Belenov, K Rodyushkin, A Kuranov, V Eruhimov, Computer vision workload analysis: Case study of video surveillance systems. Intel Techn J
43. D Comaniciu, V Ramesh, P Andmeer, Real-time tracking of non-rigid objects using mean-shift. in *Proceedings of the IEEE International Conference on Pattern Recognition*. **2**, 142–149 (2000)
44. K Nummiaro, E Koller-meier, L Van Gool, A color-based particle filter. 53–60 (2002)
45. M Zúñiga, F Brémond, M Thonnat, Fast and reliable object classification in video based on a 3D generic model. in *Proceedings of the International Conference on Visual Information Engineering (VIE2006)* Bangalore, India 433–440 (2006)
46. B Georis, M Mazière, F Brémond, M Thonnat, A video interpretation platform applied to bank agency monitoring, in *Proceedings of the International Conference on Intelligent Distributed Surveillance Systems (IDSS04)*, London, Great Britain, pp. 46–50 (2004)
47. A-T Nghiem, F Brémond, M Thonnat, V Valentin, Etiseo: performance evaluation for video surveillance systems. in *Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007)*, London, United Kingdom 476–481 (2007)

doi:10.1186/1687-6180-2011-142
Cite this article as: Zúñiga *et al.*: Real-time reliability measure-driven multi-hypothesis tracking using 2D and 3D features. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:142.