UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THESE

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Spécialité : INFORMATIQUE

présentée et soutenue par

Marcos ZUNIGA

Incremental Learning of Events in Video using Reliable Information

Thése dirigée par Monique THONNAT

et co-dirigée par François BRÉMOND

Équipe d'accueil: PULSAR - INRIA Sophia Antipolis

soutenue le 28 Novembre 2008

Jury:

M.	Gian Luca	FORESTI	Pr. University of Udine, Italy	Rapporteur
M.	David	HOGG	Pr. University of Leeds, England	Rapporteur
M.	Cédric	AWANZINO	Bertin Technologies, France	Examinateur
M.	François	BRÉMOND	CR, INRIA Sophia Antipolis, France	Co-Directeur
Mme.	Monique	THONNAT	DR, INRIA Sophia Antipolis, France	Directrice
M.	Pierre	COMON	Pr. UNSA, France	Président

To my beloved family, Carolina and Cristóbal...

Acknowledgements

I would like to thank Pr. David Hogg and Pr. Gian Luca Foresti for accepting to review this manuscript and for their very pertinent advises and remarks.

Thanks to M. Cédric Awanzino for accepting been part of the committee and for his very interesting feedback about the thesis manuscript. I would like to thank Pr. Pierre Comon for accepting being the president of the committee.

Thanks to the Science and Technology Research Council of Chile (CONICYT) which has supported in part this PhD Thesis in the framework of INRIA (Sophia-Antipolis) and CONICYT cooperation agreement.

Thanks to my adviser Monique Thonnat for accepting me in the team and for teaching me all the rigurosity needed for expressing my ideas in a scientific level. Thanks to my co-adviser François Brémond for guiding me in all the technical and redaction aspects of my thesis, which have given me the knowledge about a subject which was totally new for me.

Thanks to Catherine Martin who makes easier the life of PULSAR team for her efficiency and good will on solving any imaginable administrative problem.

Special thanks to all my PULSAR team colleagues, specially for Becha, Nadia, Valery, Bernard, Lan, Guido, Etienne, Luis, Antonio, and Guillaume for all the good moments spent in the team. More specially for Becha, Valery, Bernard, Nadia and Guido for our friendship.

Thanks to all the Chilean Mafia at INRIA for the good moments together. Special thanks to Juan Carlos Maureira and Mara Quintana for receiving me and my family in their home on our last days in France.

Thanks to my parents Carlos y Verónica for their support and love. Thanks also to all the family and friends from Chile who sent me their support.

The most special and big thanks are for my wife Carolina who always supported and comforted me in the hard moments, who always have filled me with her love, and who gave me the most important achievement of our life: our son Cristóbal. This thesis is

thanks to them and is for them.

Finally, i would like to thank and to present my excuses to all the persons i have forgotten to mention in this section.

Résumé

L'objectif de cette thèse est de proposer une approche générale de compréhension de vidéo pour l'apprentissage et la reconnaisance d'événements, dans des applications du monde réel. L'approche est composée de quatre tâches:
En premier lieu, pour chaque frame de la vidéo, une tâche de segmentation consiste à détecter les régions mobiles, lesquelles sont représentées par des boîtes englobantes qui les délimitent.
En second lieu, une nouvelle méthode de classification 3D associe à chaque région mobile un label de la classe d'objet (par exemple, personne, voiture) et un parallélépipède 3D décrit par sa largeur, sa hauteur, sa longueur, sa position, son orientation, et des mesures de fiabilité associées à ces attributs.
En troisième lieu, une nouvelle approche de suivi d'objets multiples utilise ces descriptions d'objet pour générer des hypothèses de suivi par rapport aux objets évoluant dans la scène. En dernier lieu, une nouvelle approche d'apprentissage incrémental d'événements agrège en ligne les attributs et l'information de fiabilité des objets suivis afin d'apprendre des concepts qui décrivent les événements se déroulant dans la scène. Des mesures de fiabilité sont utilisées pour focaliser le processus d'apprentissage sur l'information la plus pertinente. Simultanément, l'approche d'apprentissage d'événements reconnaît des événements associés aux objets suivis dans la scène. L'approche de suivi d'objets a été validée en utilisant des benchmarks de video-surveillance libres d'accès. L'approche complète de compréhension de vidéo a été evaluée en utilisant des vidéos obtenues d'une application réelle de maintien de personnes âgées à domicile. L'approche a été capable d'apprendre avec succès des événements associés aux trajectoires (e.g. le changement dans la position 3D et la vitesse), la posture (e.g. se lever, s'accroupir), et l'interaction entre objets (e.g. une personne s'approchant d'une table), parmi d'autres événements, avec un effort minimal de configuration.

Mots clés:
Compréhension de vidéo, répresentation 3D des objets, suivi des objets, mesures de fiabilité, apprentissage incrémental, apprentissage des événements.

Abstract

The goal of this thesis is to propose a general video understanding framework for learning and recognition of events occurring in videos, for real world applications. This video understanding framework is composed of four tasks:
First, at each video frame, a segmentation task detects the moving regions, represented by bounding boxes enclosing them.
Second, a new 3D classifier associates to each moving region an object class label (e.g. person, vehicle) and a 3D parallelepiped described by its width, height, length, position, orientation, and visual reliability measures of these attributes.
Third, a new multi-object tracking algorithm uses these object descriptions to generate tracking hypotheses about the objects evolving in the scene. Reliability measures associated to the object features are used to perform a proper selection of valuable information.
Finally, a new incremental event learning algorithm aggregates on-line the attributes and reliability information of the tracked objects to learn a hierarchy of concepts describing the events occurring in the scene. Reliability measures are used to focus the learning process on the most valuable information. Simultaneously, the event learning approach recognises the events associated to the objects evolving in the scene.
The tracking approach has been validated using video-surveillance benchmarks publicly accessible. The complete video understanding framework has been evaluated with videos for a real elderly care application. The framework has been able to successfully learn events related to trajectory (e.g. change in 3D position and velocity), posture (e.g. standing up, crouching), and object interaction (e.g. person approaching to a table), among other events, with a minimal configuration effort.

Keywords:
Video understanding, 3D object representation, object tracking, reliability measures, incremental learning, event learning.

Contents

Chapter 1

Introduction

One of the most challenging problems in the domain of computer vision and artificial intelligence is the automatic interpretation of image sequences or video understanding. The research in this area concentrates mainly on the development of methods for the analysis of visual data to extract and process information about the behaviour of physical objects in a real world scene.

The advance in low-level visual data extraction in video, has allowed researchers to focus on higher level analysis involving temporal aspects, as event recognition and learning. In the latest years, video event analysis has become one of the biggest focus of interest in the video understanding community [Hu et al. 2004a], even if the number of studies in this area is still low, compared with other areas in video understanding. The extraction of event information in video generally implies the proper processing of low-level video processing tasks, as motion detection, object classification, and tracking, in order to generate the appropriate input for the event analysis tasks.

The goal of this thesis is **to propose a video understanding framework for general event learning and recognition addressing real world applications**.

An increasing number of event analysis approaches have been proposed in the latest years. The interest of researchers has been mainly focused on the recognition of pre-defined events [Howarth and Buxton 2000], [Medioni et al. 2001], off-line learning of the relations between pre-defined events [Hongeng et al. 2004], [Chan et al. 2006a], [Hamid et al. 2005], [Toshev et al. 2006]), and off-line learning of events [Fernyhough et al. 2000], [Remagnino and Jones 2001], [Hu et al. 2006], [Niebles et al. 2006], [Xiang and Gong 2008]. To date, very little attention has been given to incremental event learning in video [Mugurel et al. 2000], [Piciarelli and Foresti 2006], which should be the natural step further of real-time applications for unexpected event recognition, or anomalous behaviour detection.

The analysis of events in video has several interesting applications. Video surveillance is one of the most important application domains. For the safety of the public places,

video camera surveillance is commonly used, but the dramatical increase of the number of cameras has lead to the saturation of the transmission and analysis means, as it is difficult to supervise simultaneously hundreds of screens. For assisting in this difficult task, video understanding techniques can be utilised for filtering and sorting the scenes which can be interesting for a human operator. For example, the AVITRACK project for video surveillance in airports [AVITRACK 2002], reports to the operators the apron activities occurring (e.g. refuelling operation), and generates alarms in case of undesired situations (e.g. collision between a cargo vehicle and and an aircraft). As another example, the CARETAKER project for behaviour analysis in public spaces [CARETAKER 2006], [Carincotte et al. 2006], generates alarms in case of undesired situations (e.g. persons fighting in a parking lot), and performs data mining on long duration video sequences for analysing patterns of behaviour of the objects evolving in the scene.

Another interesting application domain is health-care monitoring. It consists in monitoring the activity of a person through cameras and sensors in order to ensure her/his physical and mental integrity. For these applications, video understanding techniques can be utilised for automatically generating alarms in case that the health of the monitored person is in danger. For example, GERHOME project for elderly care at home [GERHOME 2005], [Zouba et al. 2007]), utilises heat, sound and door sensors, together with video cameras for monitoring elderly persons. The video understanding system proposed at GERHOME project is able to alert the family or to demand for medical support in case that an accident is detected (e.g. person falling down), and to monitor the behaviour of the person alerting her/him if some necessary action is not performed (e.g. the person did not that her/his medication, or did not take water for a long period in a hot season).

The utilisation of incremental event learning in video understanding allows to obtain the probability of occurrence of the events in a video scene, which can be utilised for the detection of abnormal situations based on an adaptive model of the event frequency in a video scene. The detection of abnormal situations can be an interesting characteristic for many video surveillance and health-care applications, as it allows to alert an operator about the occurrence of a new unknown situation, which could be undesirable or dangerous.

This thesis centres its interest in applications for incremental event learning, where several objects of diverse type can interact in the scene (e.g. persons, vehicles). The events of interest are also diverse (e.g. events related to trajectories, human posture) as the focus of interest is learning events in general. The objects simultaneously evolving in the scene can be many, but the interest is centred in objects which can be individually tracked in order to be able of recognising the events each object is participating.

For achieving the goal of this thesis, a new video understanding framework for general event learning and recognition is proposed. This approach involves a complete framework for event learning including video frame segmentation, object classification, object

tracking, and event learning tasks:

1. First, at each video frame, a segmentation task detects the moving regions, represented by bounding boxes enclosing them.

2. Second, to each moving region, a new 3D classifier associates a object class label (e.g. person, vehicle) and a 3D parallelepiped described by its width, height, length, position, orientation, and visual reliability measures of these attributes.

3. Third, a new multi-object tracking algorithm uses these object descriptions to generate tracking hypotheses about the objects evolving in the scene. Reliability measures associated to the object features are used to perform a proper selection of valuable information.

4. Finally, a new incremental event learning algorithm aggregates on-line the attributes and reliability information of the tracked objects to learn a hierarchy of concepts describing the events occurring in the scene. Reliability measures are used to focus the learning process on the most valuable information. Simultaneously, the event learning approach recognises the events associated to the objects evolving in the scene.

Next Section 1.1 presents the hypotheses and objectives for this thesis work. Then, Section 1.2 describes the structure of this thesis, where a short description of the contents for each chapter is presented.

1.1 Thesis Hypotheses and Objectives

The framework assumes the following hypotheses:

- **Mono-camera application:** The framework has been conceived for considering as input only one camera. This framework infers 3D information of the physical objects evolving in the scene by using a priori knowledge about the objects expected to be present in the scene. Even if the mono-camera constraint seems very limiting, in real world applications it is often the case to process separately the cameras of a large network.

- **Fixed-camera hypothesis:** The framework considers a fixed camera configuration. This hypothesis implies the availability of a model for transforming 2D image referential points to 3D scene referential points. The process of finding this mapping transform is known in the video processing domain as calibration. In the scope of this thesis, a pinhole camera model is utilised, which considers the mapping between 2D image points and 3D scene points as a linear transform represented by a projection matrix. For performing the calibration process, an off-line process called the Direct Linear Transform (DLT) algorithm [Abdel-Aziz and Karara 1971] is utilised. DLT algorithm consists in finding the projection matrix by solving the

linear problem $X = AY$, where each column $x_k \in X$ corresponds to a 2D image point, the column $y_k \in Y$ to the corresponding 3D point in the scene referential, and A to the transform to be found. The mentioned projection matrix is often referred as perspective matrix.

- **Available 3D object models:** This hypothesis is more desirable than compulsory, as the availability of 3D object models allows the different tasks of the video understanding framework to perform a better analysis of the objects evolving in the scene. The availability of 3D object models allows the classification task to feed the tracking process with a more precise description of the mobile objects present in the scene, allows the object tracking task to perform a more detailed analysis of the possible configurations for the tracked objects, and allows the event learning task to learn from more interesting object attributes.

- **Real world applications:** The video understanding framework application must be suitable for learning events from video. This suitability implies that several factors must be considered:

  - *Video sequence quality:* The quality of the analysed video sequence must be sufficient for detecting the objects evolving in the scene with an acceptable level of reliability. Excessive video noise, too low video frame rate, or a big lack of contrast between the objects and the background of the scene, among others, can be the factors which prevent the right detection of an object. This constraint does not mean that the interest is only centred in video sequences of high definition and quality, as mechanisms are provided to control several of these factors if their consequences in the video sequence are not severe.
  - *Crowding level:* The number of objects simultaneously evolving in the scene is not limited, but it is a fact that it can affect the performance, then it is an aspect to be considered. The separability of objects evolving in the scene is a more important factor, as the video understanding framework requires the information of events for each individual object. This factor does not mean that dynamic object occlusion can not occur, as mechanisms for cope with occlusion exist in the framework, and will properly work according to the reliability obtained for the object attributes in the previous frames.
  - *Real-time performance:* The real-time computing performance is a desirable factor of the proposed framework. Several aspects can prevent the framework to accomplish this factor, as for example an excessive number of objects evolving in the scene, a highly demanded precision for object attributes, or a huge number of possible object classes to analyse. Depending if an application requires or not an on-line response from the video understanding framework, this factor becomes more or less desirable.

Given the complexity of the problem to be solved, this thesis work tries to respond to several general questions arising:

1. **How to diminish the gap between low-level video processing tasks and event learning?** Currently, general complex event recognition and learning is performed by pre-defining the basic events of interest for the user. When the interest is also focused in learning these basic events, current studies centre their attention in particular event types (e.g. trajectories).

2. **How can generic frequent events occurring in a scene be learnt and recognised on-line, keeping a computing time performance adequate for real world applications?**

3. **How can the information needed for event learning be robustly extracted from noisy videos?**

For responding to these questions, the proposed video understanding framework establishes two global objectives:

1. **To propose a general approach for frequent event learning, able to properly work in real world applications**. For this purpose, an incremental learning approach is proposed in order to be able to learn simple events on-line directly from mobile object attribute information, with minimal learning processing time when new information arrives to the system. The learnt events can be used to bridge the gap between low-level video processing tasks and high-level complex event analysis for generic events, by considering these simple events as building blocks of the complex events.

2. **To propose a learning approach able to robustly handle noisy information**. For achieving this robustness, a complete framework has been proposed, which utilises reliability measures for accounting the quality and coherence of the acquired data. The reliability information is associated to the tracked object features, and computed for the different tasks of the video understanding framework.

This way, the global contributions of this approach are the following:

1. **A new incremental event learning approach able to learn the frequency of generic events from a video scene.** This approach proposes an automatic bridge between the low-level data obtained from objects evolving in the scene and higher level information which considers the temporal aspect. Incremental learning of events can be useful for abnormal event recognition and to serve as input for higher level event analysis.

2. **A new global way of managing noisy information.** The video understanding framework proposes to associate reliability measures to obtained information, in order to be able of accounting for the quality, coherence, and reliability of this information. This way, most valuable information can be identified in order to increase the robustness on tracking by focusing the object tracking process on most coherent and certain object features, and to focus the learning process on the most reliable information.

1.2 Thesis Structure

First, Chapter 2 describes the state of the art related to the presented video understanding framework. As the proposed framework addresses the whole issue of video understanding, this chapter has been separated in five sub-parts covering: object representation, multi-object tracking, the utilisation of reliability measures in video understanding, incremental concept formation, and event learning from video.

Second, Chapter 3 presents a global view of the proposed video understanding framework, giving a detailed description of the problem to be solved. This chapter gives a general description of the proposed video understanding framework for event learning and recognition. Also, the solutions proposed for the problems present at each task of the video understanding framework are introduced. The possible user interactions with the framework are also described. The three following chapters give a detailed description of each component of the proposed framework.

In Chapter 4, the utilised object representation is described in detail. This description includes the mathematical formulation of the parallelepiped model, the calculation of different alternative models, the detection of static occlusion situations, and the validation of the representation for its utilisation in real world applications.

In Chapter 5, the proposed multi-object tracking approach is described in detail. This description includes a framework for hypotheses modelling, the tracking algorithm and methods for hypothesis generation.

In Chapter 6, the proposed event learning and recognition algorithm is described in detail. This description includes the framework for input, state and event concept representation, and the incremental algorithm for event recognition and learning.

After, in Chapter 7 the complete video understanding framework is evaluated. Evaluation for the classification and tracking tasks have been also performed. A full evaluation of the video understanding framework has been performed, focused on different aspects as the capability of event learning and recognition, the processing time performance, and the influence of reliability measures, among other studies.

Finally, Chapter 8 presents the conclusion of this thesis work and the future research perspectives for the different contributions emanating from this work.

Chapter 2

State of The Art

This chapter has as main objective to perform a proper justification of the choices made for the proposed approach. Also, another important objective is analysing the current state of the art of event analysis in video to be able to highlight the contributions of the proposed approach. As this thesis work is involved in the resolution of several aspects of video understanding, each section of this chapter will be dedicated to the different aspects considered in the approach.

This way, first Section 2.1 explores the state of the art of object representation. Second, Section 2.2 presents the related work for the Multi-target Tracking (MTT) problem. Third, a review of the utilisation of reliability measures in video understanding is performed in section 2.3. Fourth, section 2.4 presents a review of previous work in incremental concept formation techniques. Fifth, section 2.5 presents the related work in the topic of incremental learning of events in video. Finally, in Section 2.6 the most important aspects of the state of the art with respect to this thesis work are discussed.

2.1 Object Representation for Video Understanding

This section explores different object representations utilised in video understanding, in order to establish the proper representation fitting with the objectives of the proposed video understanding framework. The choice of the right object representation plays a critical role, as it defines the precision and availability of object information to be utilised in a video understanding approach and has a direct incidence in the processing time performance of the approach.

Different object representations have been used for video understanding, normally defined by the objective or application domain. They comprise shape and appearance representation of objects, and also combinations of these representations. Appearance model includes colour, texture template, or local descriptors information which can characterise a given object or globally an object class [Quack et al. 2007]. Usually these

appearance models are either too dependent on the object appearance (i.e. colour-based techniques need a discriminative colour distribution for the tracked object), or require an extensive learning stage. As the interest of the thesis in object representation is focused in obtaining 3D features from tracked objects, appearance models are also not suitable because they base their representation on 2D image features.

In the following, commonly used shape representation of objects to be tracked are presented, describing some representative tracking approaches from the state of the art for each of these representations. These representations can be separated in general representations (Section 2.1.1), able to give a generic description for different object classes, and specific representations (Section 2.1.2), able to precisely describe a single object class.

2.1.1 General Object Representations

These representations give a general description of object classes. The main advantages of these representations are their capability of describing several object classes with the same model and their processing time performance. The main limitation is their lack of precision. According to the state of the art, these representations can be classified as:

- *Point-based representation:* The object is represented by a single point. In general, this representation is suitable for tracking objects that occupy small regions in an image. For instance, in [Veenman et al. 2001] objects are represented by their centroid (Figure 2.1). They have been extensively used in radar applications. For instance, in [Arambel et al. 2004], authors use a point representation for tracking multiple objects in a radar system application, as depicted in Figure 2.8.

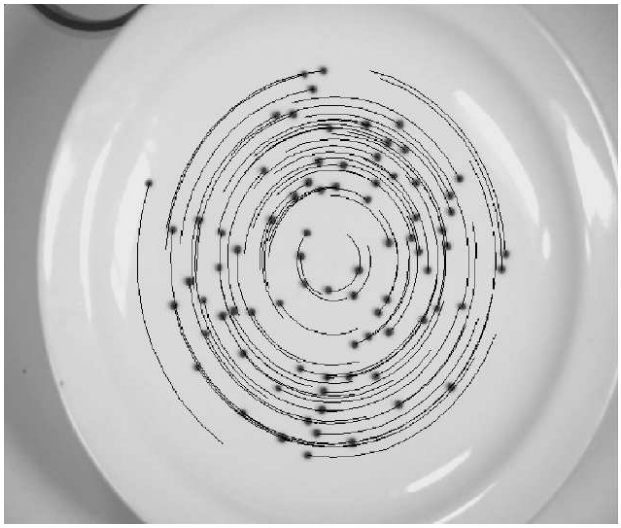
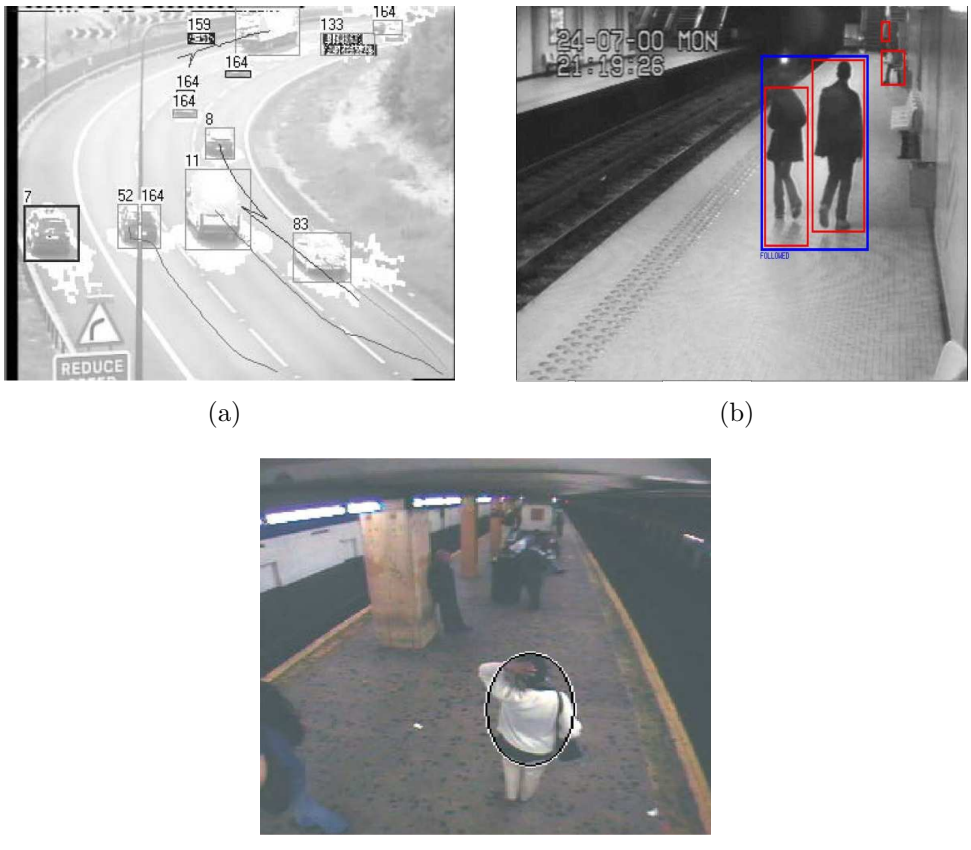


Figure 2.1: Example of point representation, where seeds in a dish are represented by their centroid [Veenman et al. 2001].

- *2D Primitive geometric shapes:* It consists in enclosing the object to be represented with a 2D primitive geometric shape. These representations have been found in the literature with several different 2D shapes (e.g. rectangles, or ellipses). For instance, in [Cucchiara et al. 2005b] objects are represented with a rectangle (Figure 2.2(a)), while in [Comaniciu et al. 2003] persons are represented by an ellipse (Figure 2.2(c)). Though primitive geometric shapes are more suitable for representing simple rigid objects, they are also used for tracking non-rigid objects. For example, in [Cupillard et al. 2001] authors track groups of people in a metro scene using a rectangular representation ((Figure 2.2(b)). Because of their simplicity they are suitable to complex real world applications with multiple targets. The main drawback of these representations is their lack of precision, specially dealing with objects as people which shape does not fit properly with a simple geometric shape.



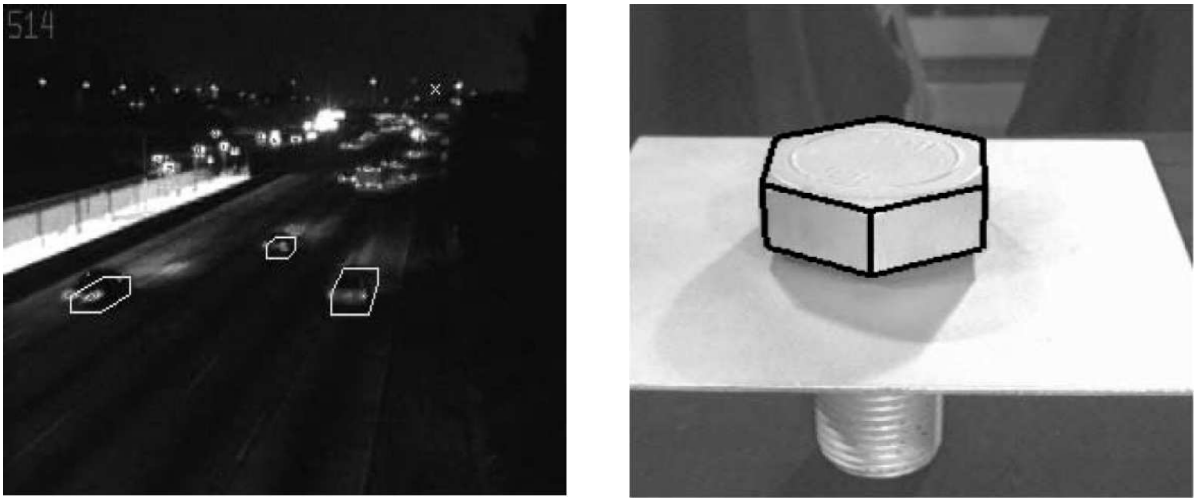
(a) (b)

(c)

Figure 2.2: Examples for 2D primitive shape representations for tracking. In Figure (a), tracked vehicles are represented by a rectangle [Cucchiara et al. 2005b]. In Figure (b) an example of a non-rigid object (group of people) represented by a rectangular shape [Cupillard et al. 2001]. Figure (c) shows elliptic shape representation for a person [Comaniciu et al. 2003]).

- *3D Primitive geometric shapes:* It consists in enclosing the object to be represented with a 3D primitive geometric shape. These representations have also been found

in the literature with several different 3D shapes (e.g. parallelepipeds, or cylinders). For instance, in [Isard and Maccormick 2001], [Kong et al. 2005], [Kong et al. 2006], [Kelly et al. 2006] tracked pedestrians are represented with a cylindrical shape, while in [Scotti et al. 2005], the cylinder shape representation is used for modelling both vehicles and persons. In [Lai et al. 2001], and [Yoneyama et al. 2005] vehicles are represented by a parallelepiped (Figure 2.3(a)). Also, polyhedral shape representations for diverse objects can be found in [Marchand et al. 2001] (Figure 2.3(b)). As the 2D primitive shapes, the 3D primitive geometric shapes are more



(a) (b)

Figure 2.3: Examples for 3D primitive shape representations for tracking. In Figure (a), tracked vehicles are represented by a parallelepiped [Yoneyama et al. 2005]. In Figure (b) a polyhedral shape is used for tracking a nut [Marchand et al. 2001].

suitable for representing simple rigid objects, but they are also used for tracking non-rigid objects. These representations gain in precision with respect to 2D primitive shape representations, but they are more expensive in terms of processing time performance, as the number of degrees of freedom of the 3D shapes is higher than the 2D shapes. However, they are still suitable for real world applications with multiple targets. they can be seen as the intermediate step between 2D primitive shapes, and more complex specific object representations.

2.1.2 Specific Object Representations

These representations give a specific description of an object class. The main advantage of these representations is their precision in the object description. Their main drawbacks are their inability of describing other object classes and their high processing time. According to the state of the art, these representations can be classified as:

- *Articulated models:* These models are used to represent articulated objects, composed of body parts that are held together with joints. To represent an

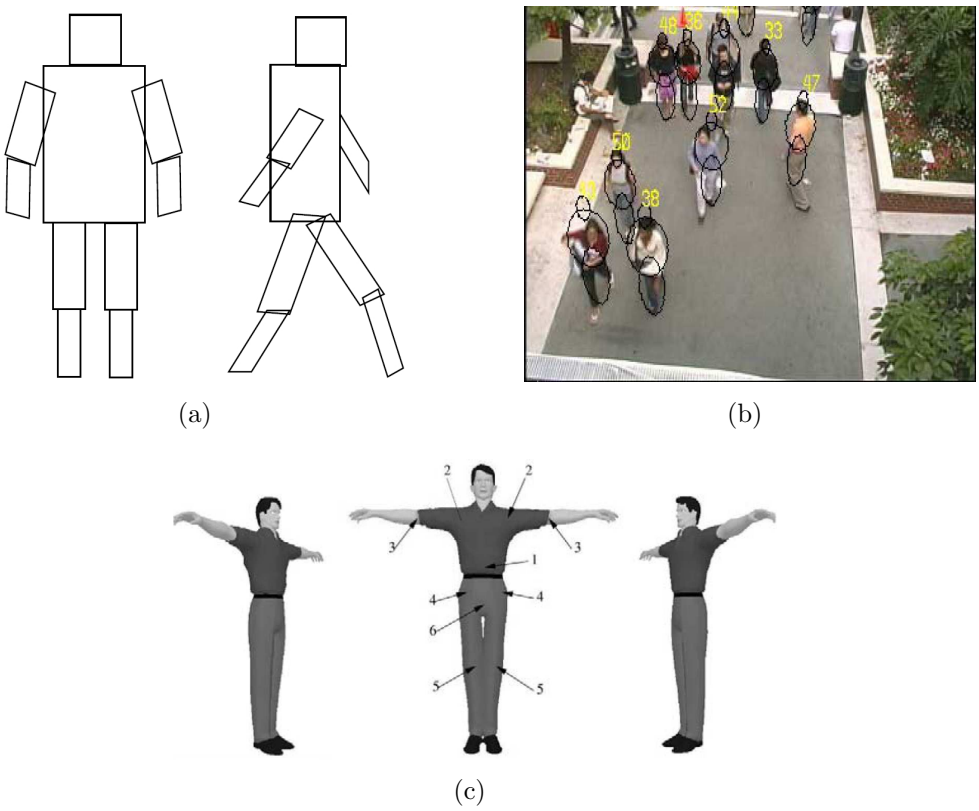


Figure 2.4: Examples for articulated models. In Figure (a), a front and lateral articulated model for a person, using rectangular patches [Black et al. 1997] is depicted. Figure (b) depicts a model for humans standing and walking consisting of a set of ellipsoids. Figure (c) depicts a complex model of human posture described by a set of 23 parameters, subject to bio-mechanical constraints [Boulay et al. 2006].

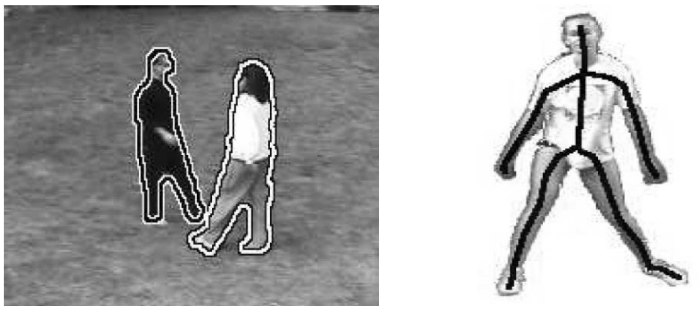
articulated object, one can also model the constituent parts using geometric shapes. For instance, [Black et al. 1997] use a 2D model of each human body part represented by planar patches (Figure 2.4(a)). Also, in [Zhao and Nevatia 2004] a 3-ellipsoid representation (for head, for torso, and for legs) is utilised for representing walking and standing humans.(Figure 2.4(b)).

In [Boulay et al. 2006] a very precise 3D model of human is utilised to detect postures. In this work, a human posture is described by a set of 23 parameters, subject to bio-mechanical constraints. This human model enables to generate 2D silhouettes to be compared with the one detected for a person in the scene (see Figure 2.4(c)). This representation is specific for one type of object class and, in general, very dependent on the application. Depending on the complexity of the model, the processing time for this type of model can be very high.

- *Contour-based representation:* This type of representation defines the boundary of an object. The region inside the contour is called the *silhouette* of the object.

Silhouette and *contour* representations are suitable for tracking complex non-rigid shapes [Yilmaz et al. 2004] (Figure 2.5(a)). Their drawback is their high processing time due to the border detection process, and then, they are not well suited for real-time applications.

- *Skeletal models* are commonly used as a shape representation for recognising objects (e.g. posture detection for humans from lateral view [Ali and Aggarwal 2001]). Figure 2.5(b) depicts a skeletal model for a person. An object skeleton can be extracted by applying medial axis transform to the object silhouette [Ballard and Brown 1982], which is very time consuming. This model can be used to model both articulated and rigid objects. They are not well suited for real-time applications, because of their high processing time.
  In [Foresti and Regazzoni 1997, Foresti 1999], the authors calculate a statistical morphological skeleton [Regazzoni et al. 1995] for classifying unknown objects and estimating their 3D orientation. This is done by comparing the calculated skeletons with those of object models stored into a database.



(a) (b)

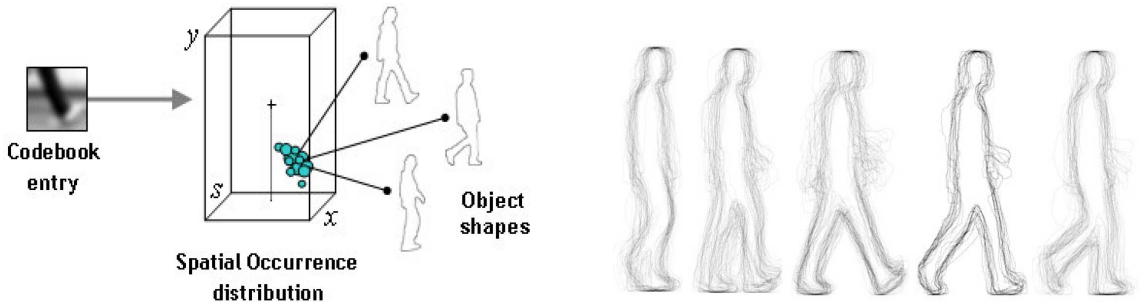
Figure 2.5: Examples of object-specific representations. Figure (a) shows a contour representation for a person [Yilmaz et al. 2004]. Figure (b) depicts a skeletal representation of a person.

In another completely different way of representing objects, other authors train classifiers with examples of the objects they expect to find in their applications. One of the precursors of this type of approach are [Viola and Jones 2001]. The authors propose to train a system in the detection of object basic features (e.g. Haar wavelets, Histograms of Oriented Gradients (HOG)), and to combine these basic features to construct strong classifiers, based on Adaboost algorithm. They present their method for an application of frontal view face detection, with high detection rates. A considerable number of studies have taken this kind of approach.

The problem of these methods is their dependence on a determined object orientation and camera position relative to the object position, as the detection is restricted to objects similar to the training samples. For example, in [Viola and Jones 2001], the face of a person seen from one side would not be detected, as their classifier was trained to detect

persons facing the camera. Then, for having a complete enough representation of an object, the size of the training set can become prohibitively large for a given application.

One of the latest contributions on this type of approaches, is the work proposed in [Leibe et al. 2005], and [Seemann et al. 2006]. In this work, authors propose a general approach for multi-aspect detection of pedestrians. They utilise an approach for multi-scale object categorisation using scale-invariant interest points called Implicit Shape Model (defined in [Leibe and Schiele 2004]).
Their approach performs a global classification based on learnt object silhouettes, for then performing another verification stage comprising locally learnt features (Figure 2.6(a)) representing articulations (Figure 2.6(b)) and viewpoints, which can be shared among these representations. This way, authors argue that their two-stage recognition approach is more robust and that their approach needs less training examples, than other similar approaches.



(a) (b)

Figure 2.6: Object recognition approach presented in [Seemann et al. 2006]. In Figure (a), for each local descriptor of typical object structures (referred as codebook entry), their approach stores the spatial occurrence distribution, as well as the associated shape. Figure (b) shows an example of shape clusters found on a training set for the right-left walking direction.

Even if their work improves the performance of this type of approaches, the limitations with respect to general models remain the same, as the recognition is limited to the training samples. For example, in [Seemann et al. 2006], authors test their approach for pedestrians walking in an environment with low camera angle. Their approach requires two annotated test sets for learning different viewpoints and postures. Moreover, the processing time is still an issue for this type of approaches.

2.2 Multi-target Tracking

This section analyses the related work for the resolution of the Multi-target Tracking (MTT) problem, in order to highlight the interesting elements from the state of the art used in the proposed tracking approach and to study the open issues for the tracking

problem. Section 2.2.1 describes the Multiple Hypothesis Tracking (MHT) algorithms addressing the MTT problem. Then, Section 2.2.2 describes other tracking algorithms addressing this problem.

2.2.1 Multiple Hypothesis Tracking

One of the first approaches focusing on MTT problem is the Multiple Hypothesis Tracking (MHT) algorithm [Reid 1979], which maintains several correspondence hypotheses for each object at each frame. 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. For reducing the processing time, they propose hypothesis elimination methods according to the likelihood of hypotheses.

In Reid's original implementation, the same dynamic model applies to all targets. In [Cox and Leonard 1994], the authors extend the MHT to a broader class of applications by allowing multiple behaviour models for different targets.

To reduce the computational load, in [Streit and Luginbuhl 1994] a probabilistic MHT (PMHT) has been proposed, in which the associations are considered to be conditionally independent random variables and thus there is no requirement for exhaustive enumeration of associations. In this work, the states of targets are modelled as continuous random variables and measurement associations to targets are modelled as discrete random variables.

Also to overcome the exponential processing time limitation of MHT, [Cox and Hingorani 1996] use an algorithm to determine the k-best hypotheses in polynomial time (proposed by [Murty 1968]) for tracking interest points. These MHT approaches are known in the literature as Hypothesis-Oriented MHT (HOMHT), as the MHT algorithm maintains and expands hypotheses from one frame to the next one, without feedback from the object measurements.

For controlling the combinatorial explosion of hypotheses in MHT all the unlikely hypotheses have to be eliminated at each frame. Several methods have been proposed to perform this task (for details refer to [Kurien 1990], and [Pattipati et al. 2000]). These methods are classified in two classes:

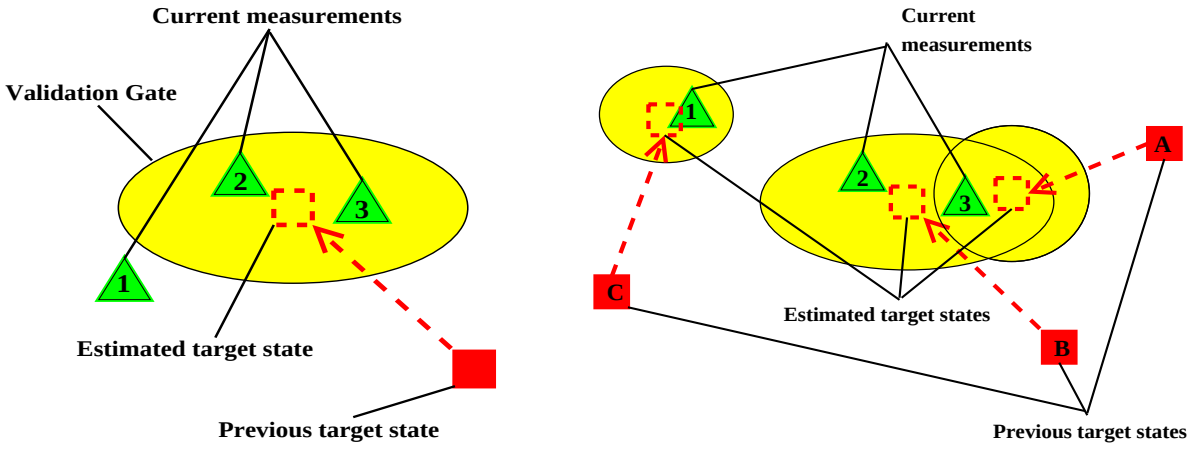
- ***Screening:*** Selective generation of hypotheses. These methods are applied prior to hypothesis generation and allow to slow the exponential growth of the number of hypotheses. In [Kurien 1990], three screening methods are described:

– **Gating:** Consists in constructing for each target a region or gate in the measurement space, which defines a validation zone for the association of measurements to the target. The shape and size of the gate may be defined in several ways. Figure 2.7(a) shows an example of the gating method.

– **Clustering:** Consists in partitioning targets into separate *clusters*. If the intersection of measurements that can be associated to a set of targets is not empty, those targets can be clustered. Figure 2.7(b) shows an example of the clustering method.

– **Classification:** Consists in grouping targets according to their confidence level. These confidence levels may be defined in several ways. For instance, the confidence levels can be proportionally defined by the *age* of the target (number of frames since the target was detected for the first time). This grouping scheme allows different criteria to be applied for screening and pruning targets with different confidence levels:

  * Enforcing stricter pruning requirements for targets with lower confidence levels. For example, a born target ($age = 1$) is allowed to fewer misdetections compared to that allowed for a higher confidence level target.
  * Imposition of restrictions on the number of associations to measurements for targets with lower confidence levels.



(a) (b)

Figure 2.7: Two screening methods for hypothesis generation. Figure (a) shows the gating method. Here, the validation gate allows to ignore the association of measurement 1 to tracked target. In Figure (b) a clustering method example is shown. Validation gates for targets A and B define the possibility of association for the same measurement 3. Thus, targets A and B are clustered.

- ***Pruning:*** Elimination of hypotheses after their generation. The two most common methods are described:

- **Lower probability:** Consists in eliminating hypotheses which probability is lower than a pre-defined threshold.
- **n-Scan Approximation:** Consists in examining a finite but variable number n of subsequent frames for assigning the measurements to targets in a particular frame, in contrast of examining measurements for all frames since the birth of targets. This method is performed in two steps:
  * Perform all feasible associations between target hypotheses from the previous frame with the measurements of the current frame.
  * Identify the most likely set of hypotheses in the n frames earlier and eliminate the rest.

Another approach for MHT is presented in [Kurien 1990], and is called the Track-Oriented MHT (TOMHT). This approach recomputes the hypotheses using the newly updated tracks with the measurements extracted in each frame. Rather than maintaining, and expanding, hypotheses from frame to frame, TOMHT discards the hypotheses formed on the previous frame. The tracks that survive pruning are predicted to the next frame where new tracks are formed, using the new observations, and reformed into hypotheses.

In [Blackman et al. 2001] processing time results for a difficult scenario with 100 closely spaced targets and a high radar update rate are presented, indicating the feasibility of real-time operation for a TOMHT. This study was performed using a single 866 MHz Pentium computer. Newer computers and/or parallel processing with several computers would allow real-time tracking for even more difficult scenarios. Interesting theoretical aspects of both HOMHT and TOMHT are discussed in [Bar-Shalom et al. 2007].

MHT methods have been extensively used in radar (e.g. [Arambel et al. 2004], [Rakdham et al. 2007]) and sonar tracking systems (e.g. [Moran et al. 1997]). Figure 2.8 depicts an example of MHT application to radar systems [Arambel et al. 2004]. In [Blackman 2004] a good summary of MHT applications is presented. However, most of these systems have been validated with simple situations (e.g. non-noisy data).

MHT is an approach oriented to single point target representation (see section 2.1), 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. Also, 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.

In the case of valid assumptions on distributions, MHT gives optimal solutions. 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 of interest for this thesis because of their simplicity. For further details on classical dynamics models used in MHT refer to [Reid 1979], [Kurien 1990], [Cox and Leonard 1994], [Streit and Luginbuhl 1994], [Cox and Hingorani 1996], and [Bar-Shalom et al. 2007].

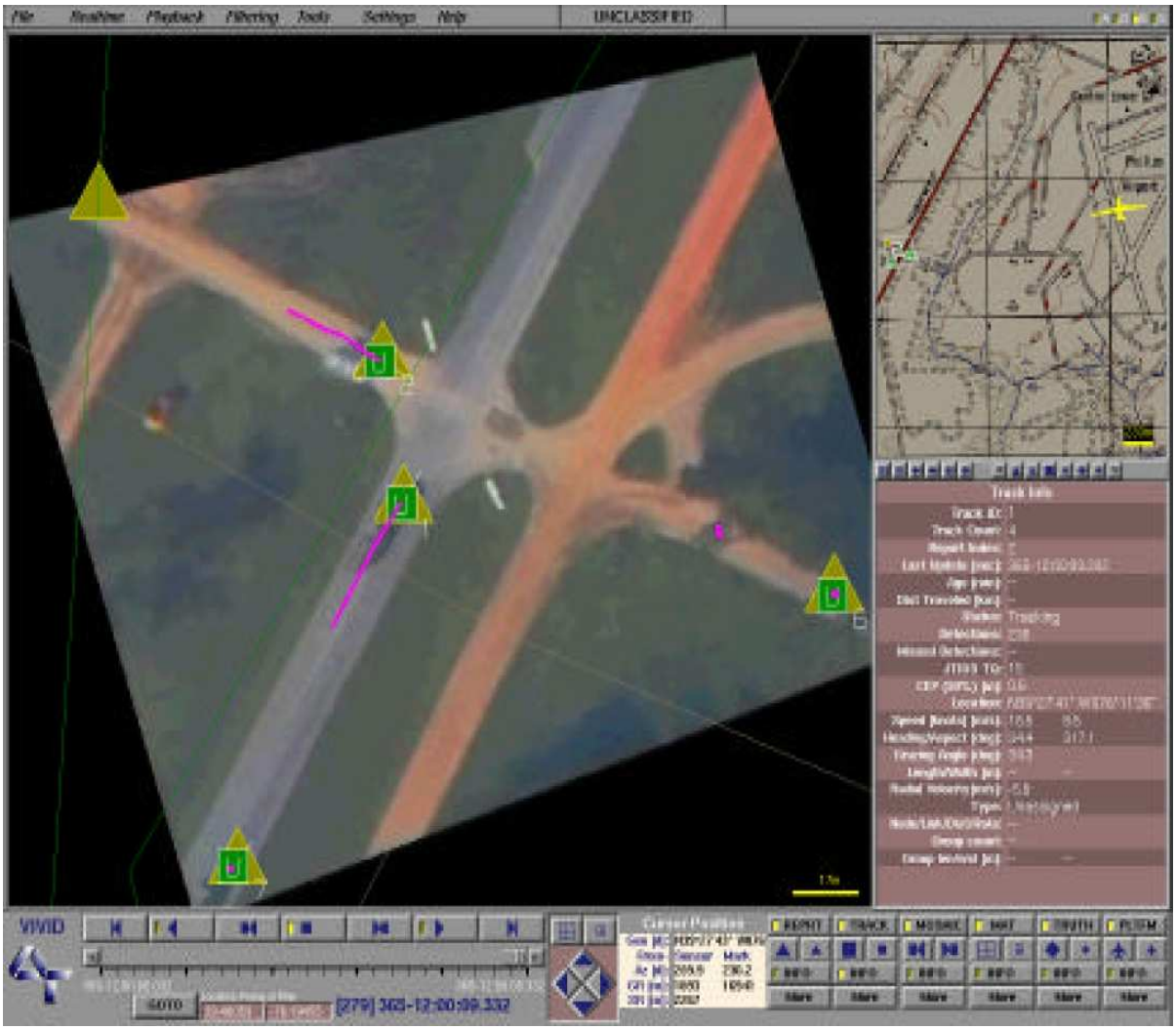
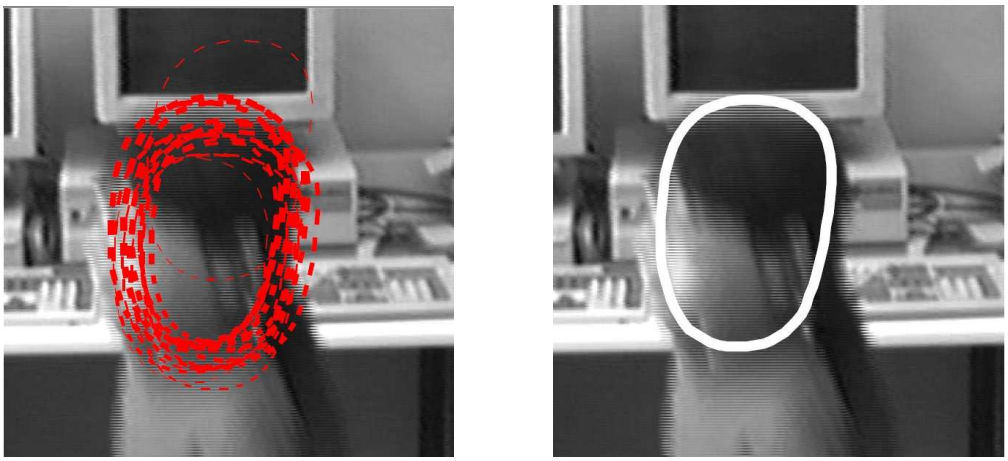


Figure 2.8: Example of a Multi-Hypothesis Tracking (MHT) application to radar systems [Arambel et al. 2004]. This figure shows the tracking display and operator interface for real-time visualisation of the scene information. The yellow triangles indicate video measurement reports, the green squares indicate tracked objects, and the purple lines indicate track trails.

2.2.2 Alternatives to Multiple Hypothesis Tracking

An alternative to MHT methods is the class of Monte Carlo methods. These methods have widely spread into the literature as bootstrap filter [Gordon et al. 1993], CONDENSATION (CONditional DENSity PropagATION) algorithm [Isard and Blake 1998], Sequential Monte Carlo method (SMC) [Doucet et al. 2001], and particle filter [Hue et al. 2002a], [Hue et al. 2002b], [Jin and Mokhtarian 2007]. They represent the state density distribution by a set of weighted hypotheses, or particles (Figure 2.9).



(a) (b)

Figure 2.9: Illustration of a sample-set representation of shape distributions for a Monte Carlo method (CONDENSATION algorithm [Isard and Blake 1998]). In Figure (a) samples of a curve distribution are displayed. Their thickness represents the weight associated to a sample. Figure (b) depicts an estimator of the distribution mean, as the weighted mean of the samples.

Monte Carlo methods have the disadvantage that the required number of samples grows exponentially with the size of the state space (perhaps as many as several thousands when the motion is poorly defined). As a consequence, an accurate dynamic model is required in practise to reduce the number of samples needed for accurate modelling. For on-line applications, the system must provide a state estimate in each frame, usually taken to be the mean or the median of the particles. This estimate is not particularly accurate. This lack of smoothness in tracking results is a major drawback for Monte Carlo methods for many applications. These factors make non-parametric techniques less attractive for objects which have both a large state space and complex dynamics. Also, as these techniques keep a non-parametric distribution of joint state probability, they scale poorly as the dimensionality increases due to a large number of objects to be tracked. Often, these techniques do not have enough information to select the proper hypotheses, specially in case of noisy videos and too simple tracking features (e.g. position, speed, height, and width).

Point trackers are suitable for tracking very small objects which can be represented by a single point representation (see section 2.1). When objects to track are represented as regions or multiple points other kinds of issue must be addressed to perform tracking. For instance, in [Brémond and Thonnat 1998a], 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 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 work, 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 [Zhao and Nevatia 2004]. Authors use a set of ellipsoids to approximate the 3D shape of a human (see Figure 2.4(b)). 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 in tracking adult humans with slight variation in posture (just walking or standing).

Another important issue in the context of multi-target tracking is the handling of missing or noisy observations. To address these problems, Monte Carlo methods explicitly handle noise by modelling uncertainty. These uncertainty measures are usually assumed to be in the form of normally distributed noise. However, the assumption that measurements are normally distributed around their predicted position may not hold. Moreover, in many cases, the noise parameters are not known.

Another possible approach for handling noise and missing observations is to enforce constraints that define the 3D structure of the object. This is addressed for non-rigid objects in [Bregler et al. 2000], [Torresani et al. 2001], [Torresani and Bregler 2002], [Torresani et al. 2004], where the authors first define a set of shape bases from a set of reliable tracks which has minimum or no appearance error on the trajectory points. Authors consider a feature as reliable if it contains a distinctive high contrast pattern with 2D texture, such as corner features [Torresani et al. 2001]. Computed shape basis then serves as a constraint on the remaining trajectory points that are labelled as unreliable. The drawback of this method is its processing time performance, far slower than frame

rate.

2.3 Reliability Measures in Video Understanding

This section analyses the way reliability measures have been used in video understanding, to establish how these measures can be used in the video understanding framework proposed in this thesis. Reliability can be defined as the confidence or degree of trust we have on a measurement. In this general sense, reliability measures can be interpreted, modelled and calculated depending on the attributes we want to measure or the observer that we want to evaluate. The observer can be a sensor (e.g. camera) or a video processing task (e.g. classifier, tracker, event analyser).

In the context of video analysis, the interest is focused on the visible attributes of objects to be analysed or in the process itself. The confidence of a reliability measure is a subjective concept. It is mostly related to the certainty in terms of error in the measurement or to the repetition of the measurement of an attribute throughout time. In the context of sensors, it is related to the weighted merge of information from different sources.

Several applications of reliability measures on video analysis can be found in the literature:

- For example, in [Irani et al. 1994], the authors use a reliability measure to determine which pixel can be reliably considered as stationary. They have proposed a method for detecting and tracking occluding and transparent moving objects, using models of optical flow. They have defined a reliability measure of the motion at each pixel which has been determined by the numerical stability of two optical flow equations proposed by [Bergen et al. 1992]. These equations represent the minimisation of the error of the incremental flow vector for general flow fields. The reliability measure is expressed by $R = \lambda_{min}/\lambda_{max}$, where λ_{max} and λ_{min} are the largest and smallest eigenvalues. This expression represents the inverse of the *condition number*[1] for the coefficient matrix of the linear system formed by the optical flow equations.

  In a similar way in the context of optical flow methods, [Tsai et al. 1999] use the smallest eigenvalue of a singular value decomposition for a similar system of equations, as a reliability measure for the motion of a pixel. The pixel estimate is considered unreliable when this eigenvalue is less than a threshold.

- Another example can be found in [Loutas et al. 2002]. The authors propose a reliability measure for tracking under occlusion, representing the efficiency of a selected region for tracking. Selected regions are represented as feature point sets

[1]The condition number associated to a linear system $Ax = b$ gives a bound on how inaccurate the solution x will be after approximating the solution. It can be roughly described as the rate at which the solution will change with respect to a change in b. Thus, if the condition number is large, even a small error in b may cause a large error in x. On the other hand, if the condition number is small then the error in x will not be much bigger than the error in b.

and the reliability measure for these sets is defined using the sum of the entropy of the feature points belonging to the set. They use this reliability measure to determine when an object is partially or totally occluded.

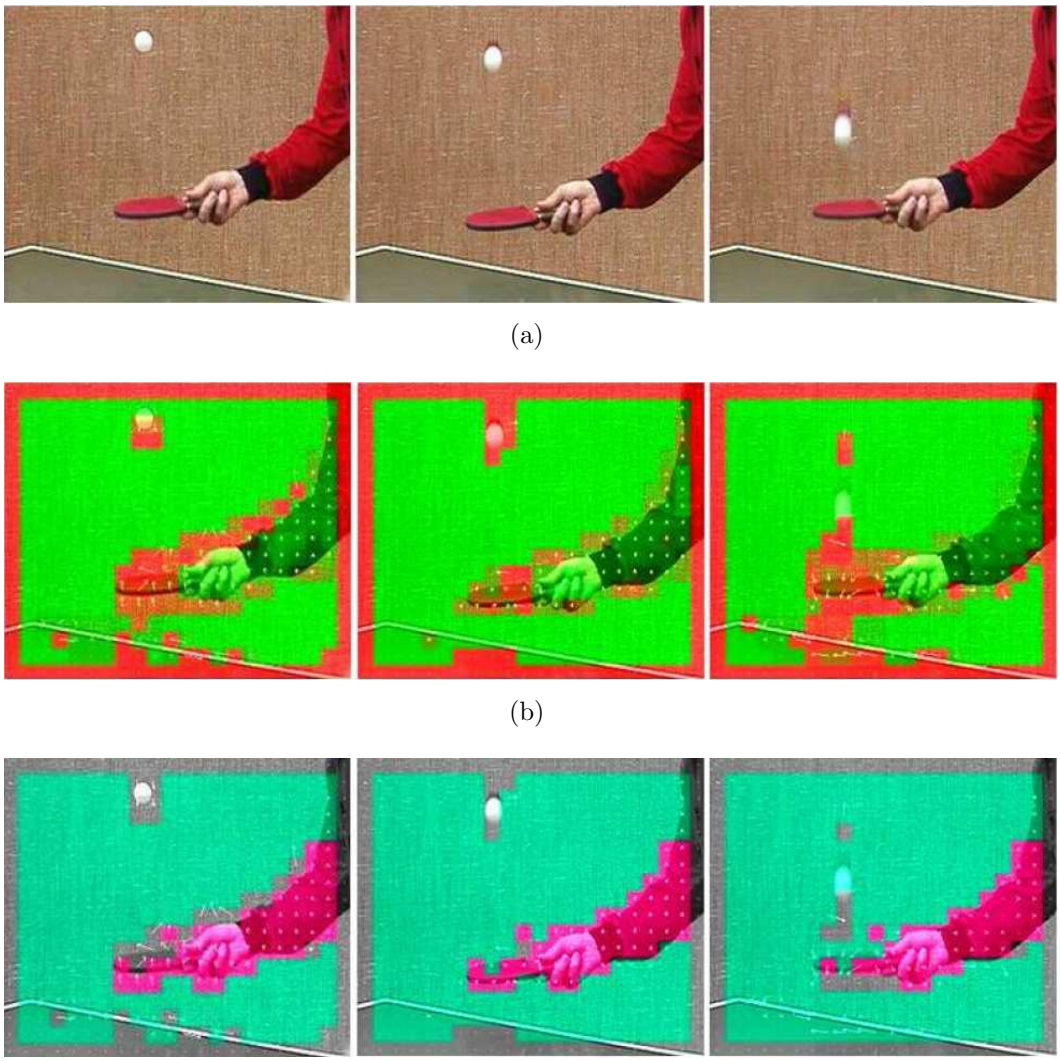
- Also, in [Ben-Ezra et al. 1994] the authors propose a reliability measure to determine the most dominant or influential pixels in terms of the gradient of the intensity of pixel. The reliability measure is computed as the module of the gradient vector.

- In [Treetasanatavorn et al. July 2005], the authors use reliability measures as weights for displacement vectors between features to be tracked. In this case, a reliability measure for a displacement vector is calculated by the local motion coherence of the vector with respect to the predicted displacement of the region (described in [Treetasanatavorn et al. August/September 2005]). Then, these reliability measures are used to reinforce the utilisation of the most reliable displacement vectors in terms of displacement coherence, in order to find a correspondence between detected and predicted regions. Figure 2.10 depicts one of the experimental results obtained in [Treetasanatavorn et al. July 2005] for frames 4, 7 and 10 (Figure 2.10(a)) of a Tennis Table sequence.

- In the context of observers, in [Kukar and Kononenko 2002] authors present a framework to calculate the reliability in classification for a single new unlabelled example for a Machine Learning Algorithm. First, the classifier is trained with a set of labelled training examples. The resulting classifier is referred as the inductive classifier, because the training phase is associated with an inductive step. Then, another classifier is obtained by training the machine learning algorithm including the unlabelled example in the training set, labelled with the result obtained using the inductive classifier. This second trained classifier is referred as a transductive classifier, because this training phase is associated with a *transductive inference*[2] process. The reliability measure is determined using the difference between the inductive classifier result for an unlabelled example and the transductive classifier result for the same unlabelled example. This reliability measure is computed as 2^{-diff}, where *diff* corresponds to the difference between the results obtained from both, inductive and transductive classifiers. These results in high reliability values for a small difference in results, and low reliability for a big difference, representing the reliability as the stability of the classification result for a new unlabelled example.

- Also in the context of observers, in [Nordlund and Eklundh 1997] and [Nordlund and Eklundh 1999], authors propose to use a reliability measure for segmentation algorithms in order to decide which segmentation algorithm to use according to the obtained segmentation results. They base their proposal in the fact that more than one algorithm supporting the same hypothesis can increase the reliability of

[2]Transductive inference consists in using both labelled and unlabelled data to predict the labels of the known unlabelled examples. In logic, statistical inference, and supervised learning, transductive inference is reasoning from observed: specific training cases to specific test cases. In contrast, induction is reasoning from observed training cases to general rules, which are then applied to the test cases.



(a)

(b)

(c)

Figure 2.10: Reliability assessment and segmentation results from sequence Table Tennis at frames 4, 7 and 10 for results presented in [Treetasanatavorn et al. July 2005]. Figure (a) displays the original frames for the sequence. Figure (b) shows the reliability measure results. Green colour corresponds to high reliability, and red colour to low reliability. In Figure (c) segmentation results are displayed. Each colour illustrates a different tracked region. No colour is used at the blocks of unreliable displacement vectors.

all involved algorithms. The reliability in the segmentation will increase if the algorithms results continue to coincide over time. When the reliability of the used algorithm goes below a threshold, another more reliable algorithm is considered for segmentation. The reliability measure for each algorithm is calculated considering the detected area of moving pixels for an object, with respect to the detected area currently used by the segmentation algorithm in time. If the currently used algorithm reliability has been high and suddenly passes below a threshold, the method switches to another segmentation algorithm.

- In [Heisele 2000], the author defines a reliability measure for the correspondence problem of cluster trajectories. This reliability measure is based on the current distance between clusters. The measure corresponds to a linear distance relation between clusters in the colour/position feature space. The measure for the reliability of a trajectory increases linearly with the mean distance of a cluster centroid to its nearest neighbours in the mentioned space. Clusters which reliability measure is lower than a pre-defined threshold are eliminated. An example for this reliability measure is shown in Figure 2.11. Figure 2.11(a) shows the result of clustering in colour/position feature space. Figure 2.11(b) shows the reliability measures for each cluster. Bright values indicate high reliability.

(a) (b)

Figure 2.11: Illustration of the measure for the reliability of trajectories used in [Heisele 2000]. The result of clustering is shown in Figure (a). The reliability measures for each cluster are shown in Figure (b). Brighter values indicate high reliability.

- Also, in [Erzin et al. 2006] authors use reliability measures as weights to combine hypotheses from different biometric sensors. The authors propose a method for person recognition in a vehicle using multiple biometric sensors. For this method, the likelihood ratio of person detection corresponds to a weighted sum of the likelihood

of detection associated to the different biometric sensors, where the weight for each sensor is defined as a reliability measure. This likelihood ratio has been defined as the Reliability Weighted Summation (RWS) rule by [Erzin et al. 2005]. The reliability measure for each sensor is based on the difference of likelihood ratios of the best two candidate person classes from a set of pre-defined person classes set. Then, the reliability measure associated to a biometric sensor is calculated considering the summation of the ratios for the true accept and true reject decisions with respect to the two best candidate person classes. Hence, the reliability measure increases when there is an evidence of either true accept or true reject, otherwise stays low.

Thus, in general terms, reliability measures are utilised:

- to combine results according to the degree of trust on different measurements weighted accordingly,
- to select the most reliable measurement, or
- to obtain a measurement of the degree of trust of an attribute or observer.

These measures allow the approaches to focus on the relevant information, allowing the achievement of higher robustness.

According to the literature, the video understanding approaches utilising reliability measures focus on computing these measures only on specific tasks of the video understanding process, defining specific measures for them. A generic mechanism is needed to compute in a consistent way the reliability measures of the whole video understanding process.

2.4 Incremental Concept Formation

In this section, the evolution of models for incremental concept formation is analysed in order to present the main concepts utilised by the proposed incremental event learning algorithm.

The objective of Machine Learning (ML) is building machines that can significantly learn for a wide variety of task domains. A computer program is said to *learn* from experience E with respect to some class of tasks T and performance P, if its performance at tasks t, as measured by P, improves with experience E [Haipeng 2003].

Machine learning can be either *supervised* or *unsupervised*. In supervised learning, there is a specified set of classes and each example of the experience E is labelled with the appropriate class. The goal is to generalise from the examples so as to identify to which class a new example should belong. This task is also called *classification*. For further details in supervised learning techniques, refer to [Kotsiantis et al. 2006].

In contrast with supervised learning, the goal of unsupervised learning is often to decide which examples should be grouped together, i.e., the learner has to figure out the classes on its own. This is usually called *clustering*. The learning approach proposed in this thesis is based on *unsupervised* learning techniques for *conceptual clustering* [Michalski and Stepp 1983].

Conceptual clustering developed mainly during the 1980s, as a unsupervised machine learning paradigm. **Categorisation** is the process in which ideas and objects are recognised, differentiated and understood. Categorisation implies that objects are grouped into categories, usually for some specific purpose. Conceptual clustering derives from attempts to explain the categorisation process, and consists in generating classes (clusters or entities) by first formulating their conceptual descriptions and then classifying the entities according to the descriptions.

In [Michalski and Stepp 1983], Michalski and Stepp provide a definition of the conceptual clustering task: given a set of instances, to place those instances into disjoint clusters and formulate descriptions for each category. Conceptual clustering systems do not only evaluate clusters based on some metrics, but also evaluate the goodness of the concepts represented by those clusters. In order to do that, these systems explicitly deal with concept descriptions and not only with extensional summaries of the clusters.

For the scope of this thesis, the interest is focused in *incremental concept formation models* [H. 1989]. This approach has the same goal than the *conceptual clustering* approach, with the added constraint that learning must be incremental. Incremental does not only mean that the process is able to create a new concept dynamically with the arrival of a new instance, but also that it does not extensively reprocess previously encountered instances, while incorporating the new one. This concept leads to the integration of learning with processing time performance [Gennari et al. 1990].

The hierarchical organisation of the acquired concepts is a distinctive feature of the methods for concept formation and conceptual clustering. Knowledge is represented by a set of nodes partially ordered by generality. Each node represents a concept, and contains intentional description of the concept. Similar hierarchical structures have been utilised in other learning approaches, but with a different purpose, as for instance the *version spaces*[3].

In the incremental concept formation models, when a new instance arrives, the process begins at the most general node and sorts the instance down through the hierarchy. Once the instance has finished its descent, one can use the concept description at the selected

[3]A version space is a hierarchical representation of knowledge used for inductive concept learning (learning general rules from positive and negative samples) to represent an unknown concept to be determined. A version space corresponds to the set of all concept descriptions within the given language which are consistent with those training instances. Each instance and description is represented by a set of symbolic attributes [Mitchell 1979], [Winston 1992].

node to make predictions about unseen aspects of the instance.

Concept formation systems have an unsupervised nature, meaning that they must decide the number of classes. The concept formation task has an incremental nature, meaning that the agent accepts instances one at a time, and it does not extensively reprocess previously encountered instances while incorporating the new one.

The described models can all be characterised as *incremental hill-climbing learners*. One can view concept formation as a search through the space of concept hierarchies, and hill climbing is one possible method for controlling that search. The most important difference between incremental hill climbing learners and the traditional hill climbing method lies in the role of input. In incremental hill-climbing systems, each step through the hypothesis space occurs in response to some new experience.

Several methods for conceptual clustering have been presented in the literature. In next sections, the most relevant methods are presented for the proposed learning approach (for further details see [Gennari et al. 1990]).

2.4.1 The beginning: Feigenbaum's EPAM

Feigenbaum's Elementary Perceiver And Memoriser (EPAM) can be viewed as an early model of incremental concept formation [Feigenbaum 1959], [Feigenbaum and Simon 1962]. The system was intended as a psychological model of human learning on verbal memorisation tasks.

EPAM represents each instance as a conjunction of attribute-value pairs, along with an optional ordered list of component objects. Each component is in turn described as a conjunction of attribute-value pairs, with its own optional components. A primitive object is an object having only attributes and no component.

EPAM represents and organises its acquired knowledge in a *discrimination network*. Each nonterminal node in this network specifies some test, and each link emanating from this node corresponds to one possible result of that test. Each nonterminal node also includes a branch marked *other*, which lets EPAM avoid specifying all possible results of the sets at the outset. Each terminal node contains a partial set of attribute values (and component categories) expected to hold for instances sorted to that node. This node structure is known as the *image* of a stimulus.

As several incremental concept formation systems, the classification of an instance is completely integrated to the learning process. When a new instance arrives, the system sorts it through the discrimination network from the root node, until reaching a terminal node. Each non-terminal node defines a test, which is performed for the instance. If the tested attribute value equals a value associated to the branches emanating from the node, EPAM sends the instance down that branch, else the instance is sent down the *other*

branch. This process is repeated until the instance arrives to a terminal node, and at this moment EPAM has *recognised* an object as an instance of the terminal node. The results of all the tests leading to the generation of the node from the instance are associated to a structure called *image*. The image for the node generated from the mismatching image contains the original image plus the value for the discriminating test. After an instance has been recognised, EPAM invokes one of the following two learning mechanisms:

1. **Familiarisation:** It happens when the image of the node matches with the new instance (no attribute-value pair differs). This process adds to node's image the value of an attribute that occurs in the instance but not in the image. This way, EPAM makes its images more specific as more instances are processed. For example, when the instance $\{COLOUR = YELLOW; NUCLEI = ONE; TAILS = ONE\}$ is the input of the process, familiarisation would produce the discrimination tree of the Figure 2.12(b).

2. **Discrimination:** It happens when the image of the node fails to match the new instance (if any attribute-value pair differs). This process sorts the instance through the discrimination network a second time, looking for the first node at which the image and instance differ.

   - If such node is found, two new branches are created, one based on the instance's value for the test and the other based on the value for the test of the image which has differed from the instance in the first sort of the instance through the discrimination network.
     For example, when the instance $\{COLOUR = YELLOW; NUCLEI = THREE; TAILS = ONE\}$ is the input of the process, discrimination would produce the discrimination tree of the Figure 2.12(d). In the first sort of this instance through the discrimination tree, the new instance has differed in the $NUCLEI$ attribute with the image $\{COLOUR = YELLOW; NUCLEI = TWO\}$. In the second sort through the discrimination tree, the first test where images differ with the instance is the $NUCLEI$ test, with the image $\{NUCLEI = ONE; TAILS = ONE\}$. Then, from the $NUCLEI$ test node, two new branches are created: $\{NUCLEI = TWO\}$ from the first sort differing image, and $\{NUCLEI = THREE\}$ from the new instance.
   - If no such node exists, the system eventually sorts the instance back down to the terminal node where the mismatch originally occurred. Two new branches are created, with the mismatching image and the new node. The discrimination process selects a test on which the image and instance differ and which has not yet been examined. The value of this test, becomes the label for one branch and label for the second branch becomes $OTHER$.
     For example, when the instance $\{COLOUR = GREEN; NUCLEI = ONE; TAILS = TWO\}$ is the input of the process, discrimination would produce the discrimination tree of the Figure 2.12(c). In the first sort of this instance through the discrimination tree, the new instance has differed in the $TAILS$ attribute with the image $\{NUCLEI = ONE; TAILS = ONE\}$.

Then, at this image position, the test $TAILS$ is added, with two new branches: $\{NUCLEI = ONE; TAILS = ONE\}$ from the differing image, and $\{NUCLEI = ONE; TAILS = TWO\}$ from the new instance. Note that the $COLOUR$ attribute value of the new instance is not considered in the created image, as the $COLOUR$ test is not considered in the path leading to the new node.

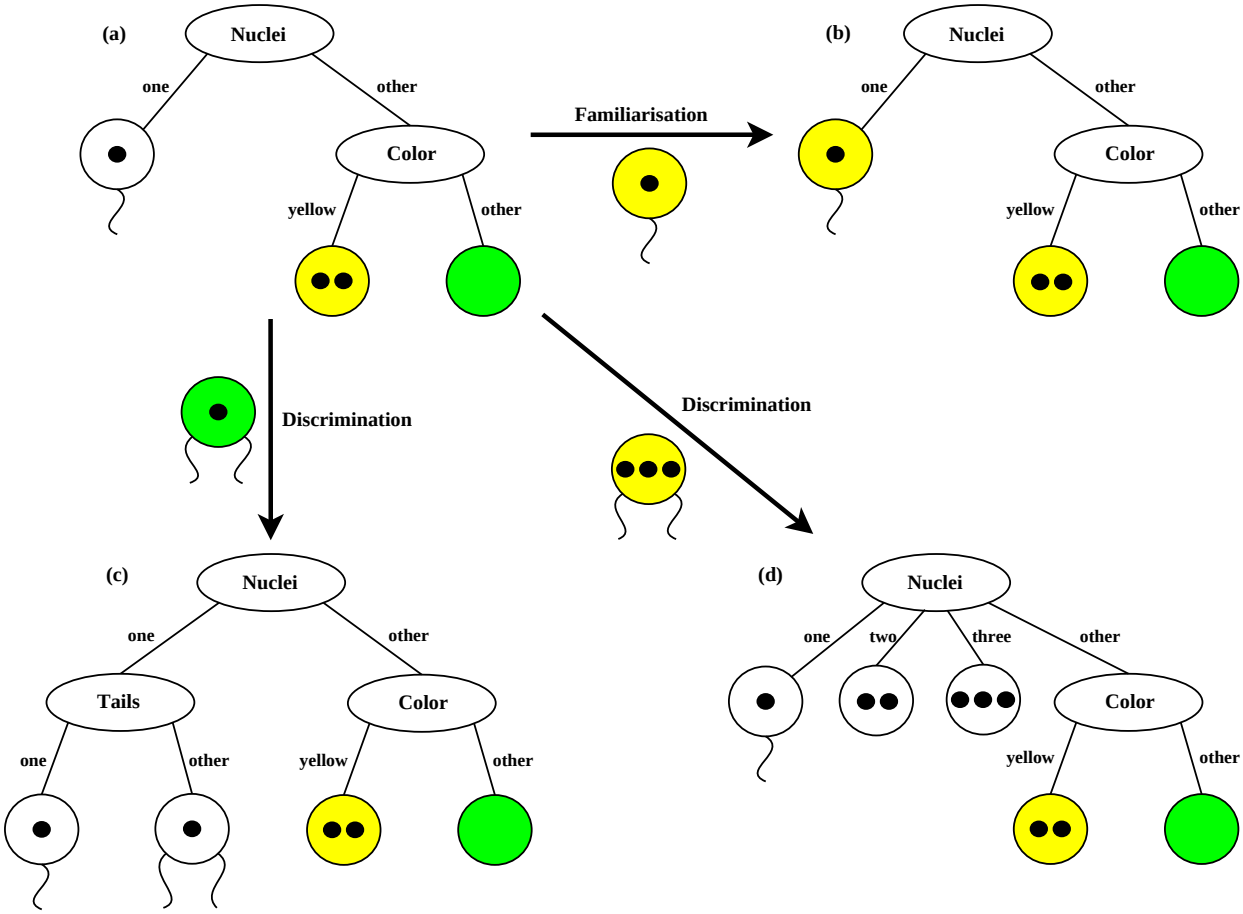


Figure 2.12: Examples of EPAM's learning method [Gennari et al. 1990].

The EPAM model introduces very important ideas into the machine learning community:

- Introduction of the notion of discrimination network. These networks can be seen as the precursors of the concept hierarchies used in later work, and images as precursors of concept descriptions.
- Distinction between instances and images.
- Distinction between the classification and prediction processes. The prediction process establishes the knowledge that can be inferred from the concept descriptions.
- Two incremental learning mechanisms were introduced.

Even if EPAM goal was the explanation of aspects of human learning and memory intention, in terms of incremental concept formation models it had some significant shortcomings:

- The concept descriptions (images) were just retained at the terminal nodes, so the discrimination network could not be considered as a true concept hierarchy.
- Concepts were considered as all or none entities, instead of considering more continuous structures.
- Just symbolic attributes were considered.

2.4.2 Methods inspired by EPAM

Three main approaches have been inspired by EPAM: UNIMEM ([Lebowitz 1983], [Lebowitz 1985], [Lebowitz 1986], [Lebowitz 1987]), CYRUS ([Kolodner 1983]), and CLUSTER/2 ([Michalski and Stepp 1983]).

UNIMEM (UNiversal MEMory model) represents each instance in the same manner as EPAM: as a conjunctions of attribute-value pairs, but it can not handle component objects. This method can handle numerical attributes in addition to symbolic ones. Symbolic attributes can represent sets. In UNIMEM, both terminal and nonterminal nodes contain concept descriptions. Each description consists of conjunction of attribute-value pairs, with each value having an associated integer representing the *predictability* in the feature (attribute-value pair).

UNIMEM also organises knowledge into a concept hierarchy through which it sorts new instances. However, the details of this hierarchy differ from EPAM's discrimination network. Nodes high in hierarchy represent general concepts, with their children representing more specific variants, and so on. Each concept has an associated set of instances stored with it, viewed as terminal nodes in the hierarchy.

As in EPAM, UNIMEM's network consists of nodes and links, with each of the node's links leading to a different child. However, UNIMEM allows each link to specify the results of multiple tests. This redundancy lets to handle missing attributes and a very flexible sorting strategy.

UNIMEM's classification system is also completely integrated with its learning method. As UNIMEM descends through the hierarchy, it uses the features on each node and its emanating links to sort the instance. If the instance matches the description of the node closely enough (parameter), then it sends the instance down those links that contain features in the instance, and it continues the process with the relevant children.

Eventually, UNIMEM reaches a node that matches the instance but none of whose children match. In this case, the system examines all instances currently stored with the node,

comparing each of them in turn to the new instance. If an old instance shares enough features (parameter) with a new one, the model creates a new more general node based on these features and stores both instances as its children. If none of the existing instances are similar enough to the new one, the system simply stores it with the current node.

Note that UNIMEM can place instances in more than one category, so these categories overlap, not forming disjoint partitions over the instances. In the literature on cluster analysis, this strategy has been called *clumping*.

Another system that independently incorporated many of the same advances as UNIMEM is CYRUS. Kolodner [Kolodner 1983] conceived this approach for modelling the organisation of episodic memory. It uses a concept representation scheme similar to the one used by UNIMEM. The clumping strategy is also shared by CYRUS. In Kolodner's work, the images are referenced as *E-MOPs* (Episodic Memory Organisation Packets), and instances are referred as *events*.

As UNIMEM, CYRUS resembles EPAM in the way of discriminating between items and in its organisation and processes describing forgetting and retrieval. UNIMEM, CYRUS, and EPAM are restricted to attribute-value languages (also known as symbolic attributes). The main difference with EPAM is that CYRUS has a concept representation allowing concept generalisation and multiple discrimination at each level in the hierarchy.

CLUSTER/2, presented in [Michalski and Stepp 1983], has introduced the *conceptual learning* paradigm. This task includes not only clustering, but also *characterisation*. The clustering problems involve determining useful subsets of an object set. This consists in identifying a set of object classes, each defined as an extensional set of objects. The characterisation problem consists in determining useful concepts for each object class. This is simply the problem of learning from examples.

This approach is non-incremental and uses a divisive technique to generate a disjoint hierarchy of concepts. The CLUSTER/2 system operates by transforming its unsupervised learning task into a series of supervised learning tasks. Thus, CLUSTER/2 does not belong to the domain of incremental concept formation models, but it has influenced several approaches in this domain with the definition of the conceptual learning paradigm. For further details on this approach, refer to [Michalski and Stepp 1983] and [Thompson and Langley 1991].

2.4.3 Fisher's COBWEB

UNIMEM and CYRUS, along with the conceptual clustering work of Michalski and Stepp [Michalski and Stepp 1983], have inspired the COBWEB system. COBWEB is an incremental system for hierarchical conceptual clustering. The system carries out a hill-climbing search through a space of hierarchical classification schemes using operators that enable bidirectional travel through this space [Fisher 1987]. Like its predecessors,

COBWEB represents each instance as a set of attribute-value pairs:

- Each attribute takes only one value and only nominal attributes are allowed (can be extended).
- Each concept node is described in terms of attributes, values and associated weights.
- COBWEB stores the probability of each concept's occurrence based on the number of instances it represents.
- Each node includes every attribute observed in the instances.
- Associated with each attribute is every possible value.
- Each such value has two associated numbers:
    1. **Predictiveness:** The *predictiveness* of a value v for an attribute a in a category c is defined as the conditional probability that an instance i will be a member of c, given that i has a value v for the attribute a or $\mathcal{P}(c|a = v)$.
    2. **Predictability:** The *predictability* of a value v for an attribute a in a category c is defined as the conditional probability that an instance i will have a value v for the attribute a, given that i is a member of c or $\mathcal{P}(a = v|c)$.

In COBWEB's concept hierarchy each node has an associated *image*, where general nodes are higher in the hierarchy and more specific ones are below its parents. COBWEB's terminal nodes are always specific instances that it has encountered. COBWEB never deletes instances and the generated hierarchy divides these instances into disjoint classes. The basic COBWEB's algorithm can be seen in Figure (2.13). Classification and learning are intertwined, with each instance being sorted down through a concept hierarchy and altering that hierarchy while passing in the following way:

- COBWEB initialises its hierarchy to a single node, setting the values of the concept attributes as the values of the first processed instance.
- Upon encountering a second instance, COBWEB averages its values into those of the concept and creates two children, one based on the first instance and another based on the second.
- At each node, COBWEB retrieves all children and considers classifying and placing a new instance in each of these categories. Each of these constitutes an alternative *clustering* that incorporates the new instance.
- Using an evaluation function, COBWEB selects the best such clustering. This evaluation function is described later (Equation (2.3)).
- COBWEB also considers creating a new category containing only the new instance, which is included in the hierarchy if the evaluation result is better than the best clustering that uses only existing categories.

function COBWEB(N, I) **returns concept_hierarchy**
Input
N: Current node in the concept hierarchy.
I: Unclassified (attribute-value) instance.
Variables
C, P, Q, R: Nodes in the hierarchy.
U, V, W, X: Clustering (partition) scores.
Begin
If N **is** *terminal_node* **then**
Create_new_terminals(N, I);
Incorporate(N, I);
Else
Incorporate(N, I);
For Each C **in** children(N) **do**
compute_score(I, C);
End For Each
P = highest_score_node(N);
W = score(P);
R = second_score_node(N);
Q = new_node(N, I);
X = score(Q);
Y = merge_score(P, R);
Z = splitting_score(P);
If W **is** *best_score* **then**
COBWEB(P, I);
Else If X **is** *best_score* **then**
initialise_probabilities(Q, I);
place_node(Q, N);
Else If Y **is** *best_score* **then**
O = Merge(P, R, N);
COBWEB(O, I);
Else If Z **is** *best_score* **then**
Split(P, N);
COBWEB(N, I);
End If
End If
Return N;
End

Figure 2.13: Top-level COBWEB algorithm.

- If the best clustering is one considering an existing category, COBWEB updates the probability of the category and the probability of its attributes. In addition, COBWEB continues to sort the instance down through the hierarchy, recursively considering the children of the category.
- If the best clustering is the one containing the new instance, COBWEB creates this new category and makes it child of the current parent node. The system bases the values for the attributes of this new concept on those found in the instance, giving them each a predictability score of one. The new included concept is a terminal node, thus halting the classification process at this step.
- COBWEB considers two additional operators to recover from non-optimal hierarchies:
  - **Merging:** At each level of the classification process, the system considers merging the two nodes that best classify the new instance. If the resulting clustering is better than the original (according to Equation (2.3)), the two nodes are combined into a single category, retaining the original nodes as its children (Figure 2.14).

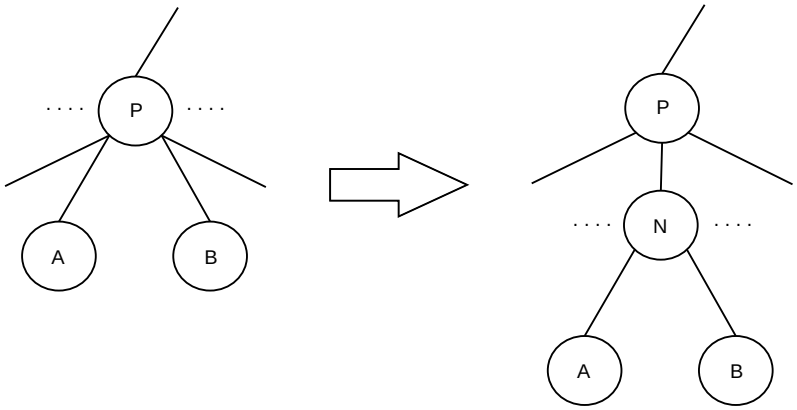


Figure 2.14: Merging categories in COBWEB.

  - **Splitting:** At each level, if COBWEB decides to classify an instance as a member of an existing category, it also considers deleting this category and elevating its children. If this action leads to an improved clustering, the system changes the structure of the hierarchy accordingly (Figure 2.15).

COBWEB does not explicitly store predictiveness scores, as they can be derived from predictability and node probability using Bayes' rule. An example of a COBWEB concept hierarchy is depicted in Figure 2.16(a), where node probability and predictability for each attribute value is displayed. Figure 2.16(b) shows a drawn representation of obtained concepts. In this representation, attributes with predictability equal to one are displayed for each category, giving an insight of what these categories are actually representing.

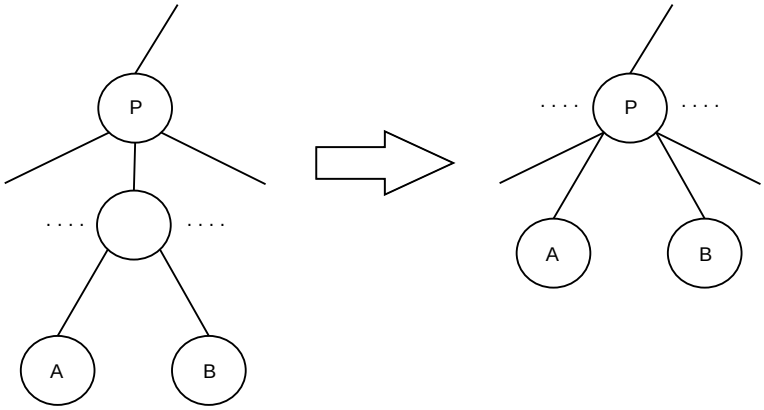


Figure 2.15: Splitting categories in COBWEB.

To evaluate the concept nodes, COBWEB uses a measure called *category utility*, which is a measure of quality for categories. This function has been derived by Gluck and Corter [Gluck and Corter 1985] by two paths, one using information theory and the other using game theory. Category utility favours clusterings that maximise the potential for inferring information. In doing this, it attempts to maximise intra-class similarity and inter-class differences, and it also provides a principled trade-off between predictiveness and predictability.

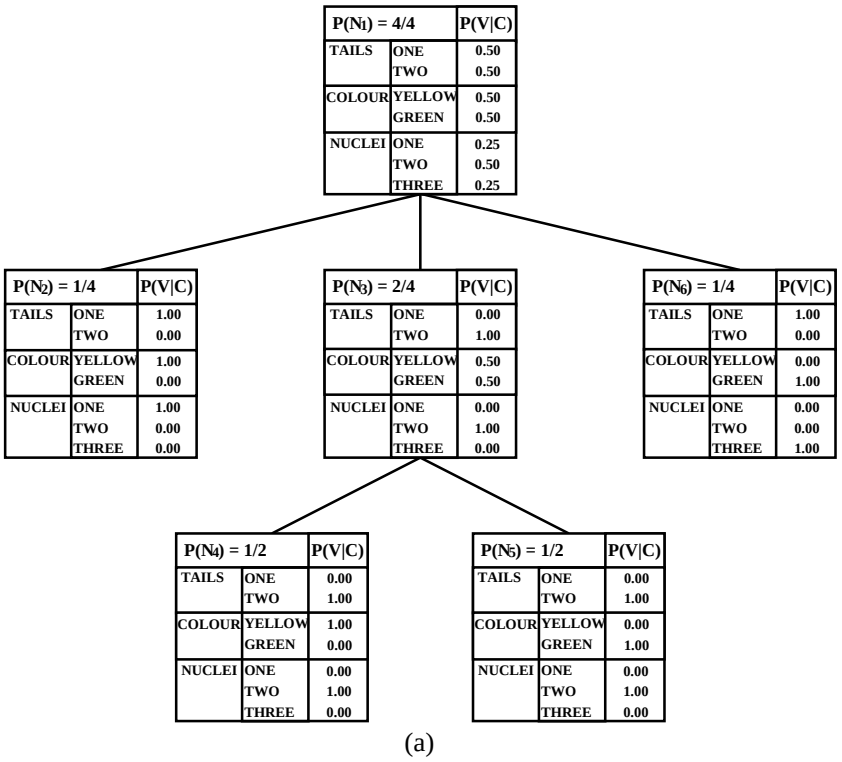
For any set of instances, any attribute value-pair $A_i = V_{ij}$, and any class C_k, one can compute $\mathcal{P}(A_i = V_{ij}|C_k)$ (predictability) and $\mathcal{P}(C_k|A_i = V_{ij})$ (predictiveness). One can combine these measures of individual attributes and measures into an overall measure of clustering quality q. Specifically:

$$q = \sum_k \sum_i \sum_j \mathcal{P}(A_i = V_{ij})\mathcal{P}(C_k|A_i = V_{ij})\mathcal{P}(A_i = V_{ij}|C_k). \tag{2.1}$$

Equation (2.1) maximises predictability and predictiveness, summed across all classes (k), attributes (i), and values (j). The probability $\mathcal{P}(A_i = V_{ij})$ weights the individual values by their occurrence frequency, giving more importance to frequently occurring values. Using Bayes', rule we have $\mathcal{P}(A_i = V_{ij})\mathcal{P}(C_k|A_i = V_{ij}) = \mathcal{P}(C_k)\mathcal{P}(A_i = V_{ij}|C_k)$. Then, expression q in Equation (2.1) can be written as:

$$q = \sum_k P(C_k) \sum_i \sum_j \mathcal{P}(A_i = V_{ij}|C_k)^2. \tag{2.2}$$

Defining category utility CU as the increase in the expected number of attribute values that can be correctly guessed, given a set of K categories, over the expected number of correct guesses without such a knowledge ($\sum_i \sum_j \mathcal{P}(A_i = V_{ij})^2$), finally we have:



$P(N_1) = 4/4$		$P(V\|C)$
TAILS	ONE	0.50
	TWO	0.50
COLOUR	YELLOW	0.50
	GREEN	0.50
NUCLEI	ONE	0.25
	TWO	0.50
	THREE	0.25

$P(N_2) = 1/4$		$P(V\|C)$
TAILS	ONE	1.00
	TWO	0.00
COLOUR	YELLOW	1.00
	GREEN	0.00
NUCLEI	ONE	1.00
	TWO	0.00
	THREE	0.00

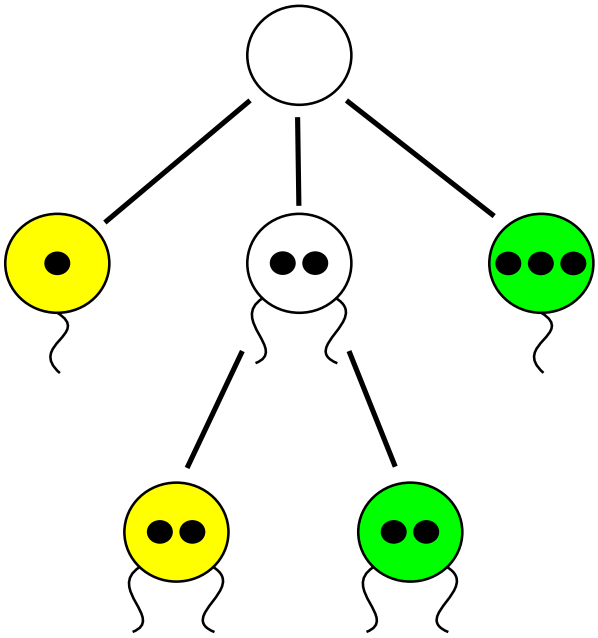
$P(N_3) = 2/4$		$P(V\|C)$
TAILS	ONE	0.00
	TWO	1.00
COLOUR	YELLOW	0.50
	GREEN	0.50
NUCLEI	ONE	0.00
	TWO	1.00
	THREE	0.00

$P(N_6) = 1/4$		$P(V\|C)$
TAILS	ONE	1.00
	TWO	0.00
COLOUR	YELLOW	0.00
	GREEN	1.00
NUCLEI	ONE	0.00
	TWO	0.00
	THREE	1.00

$P(N_4) = 1/2$		$P(V\|C)$
TAILS	ONE	0.00
	TWO	1.00
COLOUR	YELLOW	1.00
	GREEN	0.00
NUCLEI	ONE	0.00
	TWO	1.00
	THREE	0.00

$P(N_5) = 1/2$		$P(V\|C)$
TAILS	ONE	0.00
	TWO	1.00
COLOUR	YELLOW	0.00
	GREEN	1.00
NUCLEI	ONE	0.00
	TWO	1.00
	THREE	0.00

(a)



(b)

Figure 2.16: Examples of COBWEB hierarchy. (a) Detailed description of nodes, numbered in order of creation [Gennari et al. 1990]. (b) Graphic representation of the concept represented on each node. Note that just determined attribute values ($\mathcal{P}(V|C) = 1.0$) are drawn or coloured.

$$CU = \frac{\sum_{k=1}^{K} \mathcal{P}(C_k) \sum_i \sum_j \mathcal{P}(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j \mathcal{P}(A_i = V_{ij})^2}{K}. \quad (2.3)$$

The division by K allows to use the measure to compare different size clusters.

The concept hierarchy in COBWEB is very similar to UNIMEM's in that each node has an image and where general nodes are positioned higher in the hierarchy, while more specific ones are below their parents. However, COBWEB terminal nodes always correspond to specific instance as, unlike UNIMEM, COBWEB never deletes instances. Also, the concept hierarchy in COBWEB divides the instances into disjoint categories, unlike UNIMEM where concept representations can overlap. Also, COBWEB differs from UNIMEM and EPAM in that it does not associate tests on attribute values to links, leading to a novel method for sorting instances through the concept hierarchy.

One of the greatest contributions of COBWEB approach to concept formation is the use of a well-defined evaluation function for categories, allowing the comparison between concept hierarchies of different structure and size. However, this evaluation function may not completely address the user requirements.

The explicit inclusion of merging and splitting operators is also an interesting contribution, allowing COBWEB recovering from non-representative samples without losing its incremental characteristic.

The main limitations of COBWEB are its inability to handle other type of attributes, instead of just symbolic ones, and, as this approach retains all the processed instances as terminal nodes, which can lead to over-fitting the data. COBWEB also suffers the *ordering effect*, which refers to the tendency of incremental systems to create different hierarchies when the same set of input instances is presented in a different sequence order.

2.4.4 Methods inspired by COBWEB

COBWEB method has been of great impact in the incremental concept formation domain, as it has served as inspiration for a huge number of approaches. Several of these methods inspired by COBWEB are now described:

- **LABERINTH:** In [Thompson and Langley 1991], the authors propose an incremental unsupervised learning method for structured objects that acquires probabilistic concepts from relational data, using the heuristic of separating the instances into components for classification. LABERINTH uses a representation for structured objects that reduces search by decomposing them into a *partonomy*[4] of

[4]The term partonomy is used for object hierarchies, to distinguish them from concept hierarchies

components.
LABERINTH extends COBWEB method by using relational data and separating an instance into components for learning in structural domains. In terms of classification and learning, the method differs from COBWEB in that LABERINTH: (a) adds an outer loop to classify each component of a structured object, and (b) introduces a new COBWEB subroutine to form predictive characterisations of structured concepts.

- **AICC:** This system, presented in [Devaney and Ram 1994], is described as an attribute-incremental concept formation system. AICC (Attribute-Incremental Concept Creator) is able of both adding and removing attributes from an existing concept hierarchy and restructuring it accordingly. The authors introduce the notion of *attribute-incrementation*, as the dynamic restructuring of an existing hierarchy of concepts as result of a change in the attribute set used to describe the instances. The idea is to be able of incorporate the information about a new attribute to be considered in the concept description without the necessity of recalculating all the concept hierarchy from scratch. This method has been implemented as an extension of COBWEB, using as input a concept hierarchy generated by COBWEB together with new descriptions of the instances used to generate this hierarchy. This method has also been conceived for symbolic attributes.

- **INC:** This system, presented in [Hadzikadic and Bohren 1997], has been proposed to cope with COBWEB method limitations about ordering effect, and learning process performance. COBWEB and INC share several assumptions: (a) probabilistic representation of concepts, (b) incremental classification process, (c) both methods are able to handle just symbolic type of attributes, and (d) existence of a numerical evaluation function responsible for estimating the quality of the generated hierarchy. In contrast, the systems are different in several key issues:

  – The structure of the evaluation function is different. INC method uses a *cohesiveness* evaluation function. Cohesiveness calculates the average *similarity* of all pairs of instances contained in a class, reflecting the similarity between all instances under a given node. Similarity is used for both classifying previous instances and predicting the class membership of new instances and is considered as a linear combination of common and distinctive attribute/value pairs.
  The main difference between COBWEB and INC evaluation functions lies in the fact that COBWEB's category utility maximises the improvement of the clustering at the global level, i.e., the parent or root level, while INC supports a more localised approach. This new proposed structure for the evaluation function results in a better performance in the number of comparisons to classify a new instance.

(taxonomy). Partonomy is a classification based on *part-of* relations, while taxonomy is based on *is-a* relations.

  - Tree-building operators. INC uses six operators during the tree building process:
    * *Create:* To form a new class for an instance found to be dissimilar to all examined classes,
    * *Extend:* To add a new instance to the most similar class found.
    * *Merge:* To form a new class by merging the most similar children and recursively classifying the new instance into the new class, if a new instance is similar to half of the children in a class.
    * *Delete:* To undo the consequences of an unsuccessful merge operation.
    * *Pull-in:* If the *cohesiveness* measure between a sibling of a class and the class itself is lower than the *similarity* measure between the class and the sibling, the sibling is pulled into the class.
    * *Pull-out:* If the *cohesiveness* measure between a child of a class and the class itself is higher than the *similarity* measure between the class and the child, the child is promoted one level, i.e., pulled out of its current class.

    Pull-in and pull-out are responsible for reversing some unwarranted decisions made by the system, which is common to any incremental system due to the ordering effect.
  - COBWEB keeps all processed instances, while INC is able to stop the prediction process when similarity between the new instance and candidate node is lower than a pre-defined threshold, avoiding over-fitting in noisy domains at the time of retrieval.

- **OLOC:** In [Martin and Billman 1994], the incremental concept formation system OLOC has been presented. The system is able to learn and use overlapping concepts, combining multiple overlapping concepts for making predictions. OLOC uses the same concept description and hierarchy structure as COBWEB. As COBWEB, OLOC is also designed for just considering symbolic attributes. The main difference with COBWEB is that the categories used by OLOC are not individual categories, but sets of mutually exclusive categories. Each of these sets represents a distinct way of partitioning instances, typically emphasising different attributes and thus supporting overlapping concepts. This set of categories representation results in modifications in classification and learning processes in order to properly update and build the hierarchy of concepts.

- **ARACHNE:** In [McKusick and Langley 1991], the authors present ARACHNE method as a concept formation system that uses explicit constraints on tree structure and local restructuring operators to produce well-formed probabilistic concept trees, in order to cope with COBWEB limitations related with the ordering effect. Like COBWEB, ARACHNE represents knowledge as a hierarchy of probabilistic concepts, and it classifies new instances by sorting them down this hierarchy. As COBWEB, ARACHNE is able just to process symbolic attributes. The system differs from COBWEB in the following aspects:

- *Evaluation function:* The system assumes that a similarity measure is available. For both classification and learning processes, the same similarity is used to compare two concepts, two instances, or an instance and a node.
- *Structure of the concept hierarchy:* Two additional operators are proposed for restructuring the concept hierarchy:
  - **Vertical placement operator:** The system checks if each child of a node is *vertically well placed*. A node is vertically well placed in a concept hierarchy when the similarity between the node and its parent is higher than the similarity between the node and the parent of its parent. If a node is not vertically well placed, the node is promoted to the parent's level in the hierarchy.
  - **Horizontal placement operator:** The system checks if each child of a node is *horizontal well placed*. A node is horizontally well placed in a concept hierarchy when the similarity between the node and its parent is higher or equal than the similarity between the node and any of its siblings. If two or more siblings are more similar to each other than their parent, the most similar siblings are merged, averaging their probabilities and placing the originally merged siblings as children of the new node. The process is repeated with the children of the new node.

More details on ARACHNE approach, as comparative analysis with other methods can be found in [Iba and Langley 2001].

- **CLASSIT:** The method CLASSIT [Gennari et al. 1990] has been proposed as an incremental concept formation model which considers only numerical attributes. This method is very important for the scope of the performed research and is extensively described in next section 2.4.5.

- **GALOIS:** In [Carpineto and Romano 1993], authors present GALOIS as an incremental concept formation approach that helps overcome COBWEB limitations due to the ordering effect. Rather than finding and updating a particular hierarchy of concepts, GALOIS keeps and updates all the classes that can be generated in a restricted concept language. This approach relies on the theory of concept or Galois lattices and, as it is restricted to symbolic attribute representations and is not extended in the literature to numerical attributes, is out of the scope of this research work.

2.4.5 Gennari's CLASSIT

In [Gennari et al. 1989], [Gennari et al. 1990], Gennary *et al.* proposed a model of concept formation named CLASSIT, which attempts to improve upon earlier work. It has been strongly influenced by COBWEB, differing mainly in its representation of instances and concepts, and its evaluation function.

2.4.5.1 Representation and Organisation

- CLASSIT only accepts numerical attributes as input.
- CLASSIT also associates a probability distribution with each attribute occurring in the concept.
- Instead of storing a probability for each attribute value, CLASSIT stores a continuous distribution (bell-shaped curve) for each attribute, expressed by a mean value and a standard deviation.
- CLASSIT organises concepts into a hierarchy in the same manner as do UNIMEM and COBWEB. General concepts representing main instances are near the top of the tree, with more specific concepts below them. In general, concepts lower in the hierarchy will have attributes with lower standard deviation, since they represent more specific classes with greater within-group regularity.

2.4.5.2 Classification and Learning

- CLASSIT includes the same basic operators as COBWEB:
  1. To incorporate an instance into an existing concept.
  2. To create a new disjunctive concept.
  3. To merge two classes.
  4. To split a class.
- As described in the COBWEB algorithm presented Figure 2.13, for every new instance, the algorithm considers all four operators, computes the score of the evaluation function in each case, and selects the choice with the highest score.
- CLASSIT makes a few important additions to the basic algorithm:
  - Rather than always descending to the leaves of the hierarchy as it classifies an instance, CLASSIT may decide to halt at some higher level node. When this occurs, the system has decided that the instance is similar enough to an existing concept, that further descent is unnecessary and that it should throw away specific information about the instance. For determining when an instance is similar enough to a concept, a parameter named *cutoff*, based on the evaluation function, has been defined. This addition intends to avoid the over-fit of data problem of COBWEB, which always considered all instances in the concept hierarchy. Also, this addition allows to control the size of the concept hierarchy, allowing a better performance of the method.
  - If instances are described as a set of components, it is necessary that the system correctly matches instance components with concept components. Using the variances for each attribute in the concept description, CLASSIT finds a match for that component with the least associated variation. Using this as

a constraint, the system then finds a match for the next most constrained component and so forth, continuing this process until all components in the concept description have been matched against components in the instance.

2.4.5.3 Evaluation Function

As CLASSIT uses numerical attributes in both instances and concepts, a generalisation of COBWEB's category utility (Equation (2.3)) is required. The terms that have to be adapted in the expression of the category utility to numerical domains are:

$$\sum_j \mathcal{P}(A_i = V_{ij}|C_k)^2 \qquad \text{and} \qquad \sum_j \mathcal{P}(A_i = V_{ij})^2. \tag{2.4}$$

Both terms in Equation (2.4) correspond to the sum of squares of the probabilities of all values of an attribute. The former uses probabilities given membership to a particular class C_k, while the later does not consider any class information. The second terms is equivalent to the attribute value probabilities in the parent node, since this node includes the information of all processed instances. For applying these terms to continuous domains, the summation must be changed to integration, and some assumptions must be made about the distribution of values. If no prior knowledge exists about the distribution of an attribute, the best assumption is to consider a normal distribution for each attribute. Thus, the summation of the square of all probabilities becomes the integral of the normal distribution squared. For the first summation in Equation (2.4), the distribution is for a particular class, while the second summation must use the distribution at the parent class. In both case, the integral evaluates the simple expression in Equation (2.5).

$$\sum_j^{values} \mathcal{P}(A_i = V_{ij})^2 \Leftrightarrow \int \frac{1}{\sigma^2 2\pi} e^{-\left(\frac{x-\mu}{\sigma}\right)^2 dx = \frac{1}{\sigma}} \frac{1}{2\sqrt{\pi}} \tag{2.5}$$

where μ is the mean and σ is the standard deviation. Finally, since the expression is used for comparison only, the constant term $1/2\sqrt{\pi}$ can be discarded. Then, the revised evaluation function used by CLASSIT is:

$$CU = \frac{\sum_k^K \mathcal{P}(C_k) \sum_i^I \frac{1}{\sigma_{ik}} - \sum_i^I \frac{1}{\sigma_{ip}}}{K} \tag{2.6}$$

where I is the number of attributes, K is the number of classes in the partition, σ_{ik} is the standard deviation for a given attribute in a given class, and σ_{ip} is the standard deviation for a given attribute in the parent node.

Unfortunately, this transformation introduces a problem when the standard deviation is zero for a concept. For any concept based on a single instance, the value of $1/\sigma$ is infinite. In order to solve this problem, it has been introduced the notion of **acuity**, a system parameter that specifies the minimum value for σ. This limit corresponds to the notion of a *just noticeable difference* in psychophysics, the lower limit on our perception ability. This parameter can be provided by the user.

Because acuity strongly affects the score of new disjuncts, it directly controls the breadth, or branching factor of the concept hierarchy produced, just as the *cutoff* parameter controls the depth of the hierarchy.

The greatest contribution of CLASSIT approach to concept formation is the adaptation of a well-defined evaluation function for categories, to numerical domain attributes. Another contribution is the consideration of the *cutoff* parameter, which serves to diminish the risk of data over-fit.

The main limitation of CLASSIT is inherited from COBWEB, as CLASSIT also suffers from the *ordering effect*. Another limitation of CLASSIT is its inability to handle other types of attributes, instead of just numerical ones. Authors give an insight on how a mixture of symbolic and numerical attributes can be used, but they do not concretely formulate the solution.

2.4.6 From CLASSIT to present

A first extension from COBWEB and CLASSIT is presented in [McKusick and Thompson 1990]. This extension, called COBWEB/3, has been proposed to handle both numerical and symbolic attributes in the category utility measure. As in CLASSIT, COBWEB/3 assumes that the numerical attribute values are normally distributed. Then, for the set of numerical attributes, the category utility CU_k, for a given class C_k, is defined as:

$$CU_k(numerical) = \frac{\mathcal{P}(C_k)\sum_{i=1}^{I}\left(\frac{1}{\sigma_{ik}} - \frac{1}{\sigma_{ip}}\right)}{2 \cdot I \cdot \sqrt{\pi}}, \tag{2.7}$$

where σ_{ik} is the standard deviation for a given numerical attribute i in a given class, with I corresponding to the number of numerical attributes, and σ_{ip} is the standard deviation for the attribute i in the parent node.

As in COBWEB approach category utility definition (Equation (2.3)), for the set of symbolic features, the category utility CU_k, for a given class C_k, is defined as:

$$CU_k(symbolic) = \frac{\mathcal{P}(C_k)\sum_{l=1}^{L}\sum_{j}^{J_l}\left(\mathcal{P}(A_i = V_{ij}|C_k)^2 - \mathcal{P}(A_i = V_{ij})^2\right)}{L}, \tag{2.8}$$

with L corresponding to the number of symbolic attributes, and J_l to the number of possible values for the attribute l.

Then, for a set of mixed symbolic and numerical attributes, the overall category utility CU_k, given a class C_k, is the sum of the contributions of both sets of features:

$$CU_k = CU_k(symbolic) + CU_k(numerical). \tag{2.9}$$

Finally, the category utility CU for a class partition of K classes is defined as:

$$CU = \sum_{k=1}^{K}\frac{CU_k}{K} \tag{2.10}$$

Dividing the sum of class category utilities by K allows the comparison of class partitions of different size.

In response to some disadvantages in COBWEB/3 approach, ECOBWEB is proposed as part of a larger system, Bridger([Reich and Fenves 1991], [Reich 1991]). ECOBWEB attempts to remedy some of the disadvantages inherent in the COBWEB/3 interpolation scheme: (a) the normal distribution assumption for the pdf of numerical attributes, and (b) the acuity value for bounding the category utility contribution from numerical attributes.

In the ECOBWEB approach, the probability distribution for numerical attributes is approximated by the probability distribution about the mean for that feature. This probability is calculated in a designated interval range around the mean of the feature value distribution, adding parameters to be pre-defined for the interval determination.
Similar limitations due to parameter pre-definition possesses GCF method, proposed in [Talavera and Béjar 2001], which is a symbolic hierarchical clustering model that uses parametrised measures, for allowing users to specify both the number of levels and the degree of generality of each level.

In other extension for CLASSIT, in [Iba 1991] (also well described in and [Iba and Langley 2001]) the method OXBOW is proposed. OXBOW extends CLASSIT to deal with structured objects with differing numbers of components and to form concepts in temporal domains. This method represent movements as sequences of state descriptions

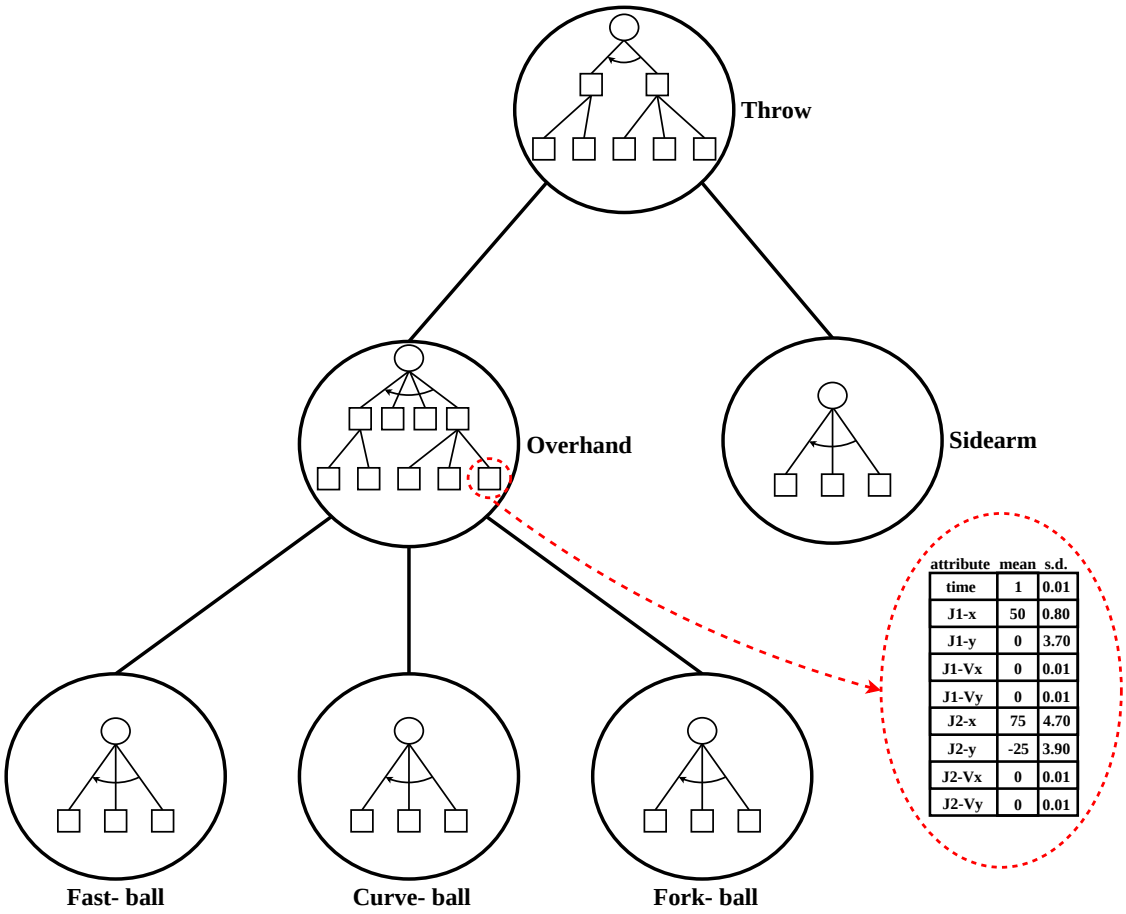


Figure 2.17: An example of hierarchy of motor schema for baseball pitches, with one node shown in detail, as presented in [Iba 1991]. Note that the top level of the state description hierarchy is ordered sequentially in time. State descriptions are described by time attribute, and position and velocity of two joints of an arm (J1 and J2).

with temporal relations among them, called *motor schema*. These motor schema are capable of capturing and summarising the original movement.
OXBOW represents a single movement concept using a probabilistic hierarchy of states, where its top-level partition is organised with respect to time only, and the nodes at this level are ordered by time to yield the state sequence of movement.
At the same time, OXBOW organises movement concepts in a probabilistic movement hierarchy, where each of these concepts points to a state hierarchy in which the top level consists of an ordered AND tree that represents the sequence of states for the movement concept. An example of this hierarchical structure is depicted in Figure 2.17. This example shows a possible hierarchy of baseball-pitching schema. The leaf nodes of the global tree represent the motor schema from specific observed pitches. The node labelled as *Overhand* represents a generalisation of the three specific throws stored below it in the hierarchy. This generalisation is also a motor schema, but instead of specific values, the generalisation stores means and variances for each of the attributes in its state descriptions

(for further details refer to [Iba 1991]).

The learning method of OXBOW is similar to CLASSIT's. A variation has been introduced to handle structured objects having temporal components. OXBOW learning process sequentially incorporates a new state sequence into an existing movement concept. For each state, OXBOW extends CLASSIT category utility function to consider structured objects, as shown in Equation (2.11).

$$CU = \frac{\sum_{k}^{K} \mathcal{P}(C_k) \sum_{j}^{J} \mathcal{P}(S_{kj}) \sum_{i}^{I} \frac{1}{\sigma_{kji}} - \sum_{m}^{M} \mathcal{P}(S_{pm}) \sum_{i}^{I} \frac{1}{\sigma_{pmi}}}{K}, \tag{2.11}$$

where $\mathcal{P}(C_k)$ is the probability of class C_k, K is the number of classes at the current level of the hierarchy, $\mathcal{P}(S_{kj})$ is the probability of the j-th state description in C_k, $\mathcal{P}(S_{pm})$ is the probability of the m-th state description in the parent of the current partition, σ_{kji} is the standard deviation of attribute i in the j-th state of the k-th class, and σ_{pmi} is the standard deviation for attribute i in the m-th state of the parent node.

For each state, OXBOW employs the category utility function presented in Equation (2.11) accounting for all movement attributes (positions and velocities), except for the first level of the hierarchy of state descriptions, where just time attribute is considered in order to organise the partition structure by temporal aspects of the movement.

Several other concept formation models have been proposed in the literature to date, improving some limitation of the presented methods, but not adding any other important feature to the incremental concept formation domain. For example, in [Li and Biswas 2002] Similarity Based Agglomerative Clustering (SBAC) algorithm is presented, which uses Goodall similarity measure [Goodall 1966]. This algorithm works well with mixed numerical and symbolic attributes, though is computationally expensive. Another example is CAS algorithm, presented in [Alomary and Jamil 2006]. CAS builds a clustering hierarchy incrementally, with each cluster node containing frequency information that maps an instance to that cluster. The representation language takes into account the current ignorance while incorporating an instance into the cluster. It combines a number of different paradigms such as constraint satisfaction, evidential reasoning, inference maximisation, and entropy maximisation. The main limitation of CAS system is that this approach allows just symbolic type attributes, preventing its utilisation for several application domains.

2.4.7 Global Scope of Incremental Concept Formation

As stated in previous sections, models of incremental concept formation have evolved in literature over years of studies. This evolution is depicted in Figure 2.18. The most important approaches for the presented learning approach are highlighted. The depicted

structure does not intend to list all the existing incremental formation approaches in the literature (that would be impossible), but to highlight the evolution line of incremental concept formation converging to the scope of the proposed learning approach in this thesis.

In the specific scope of this thesis, the interest is focused in the evolution of incremental

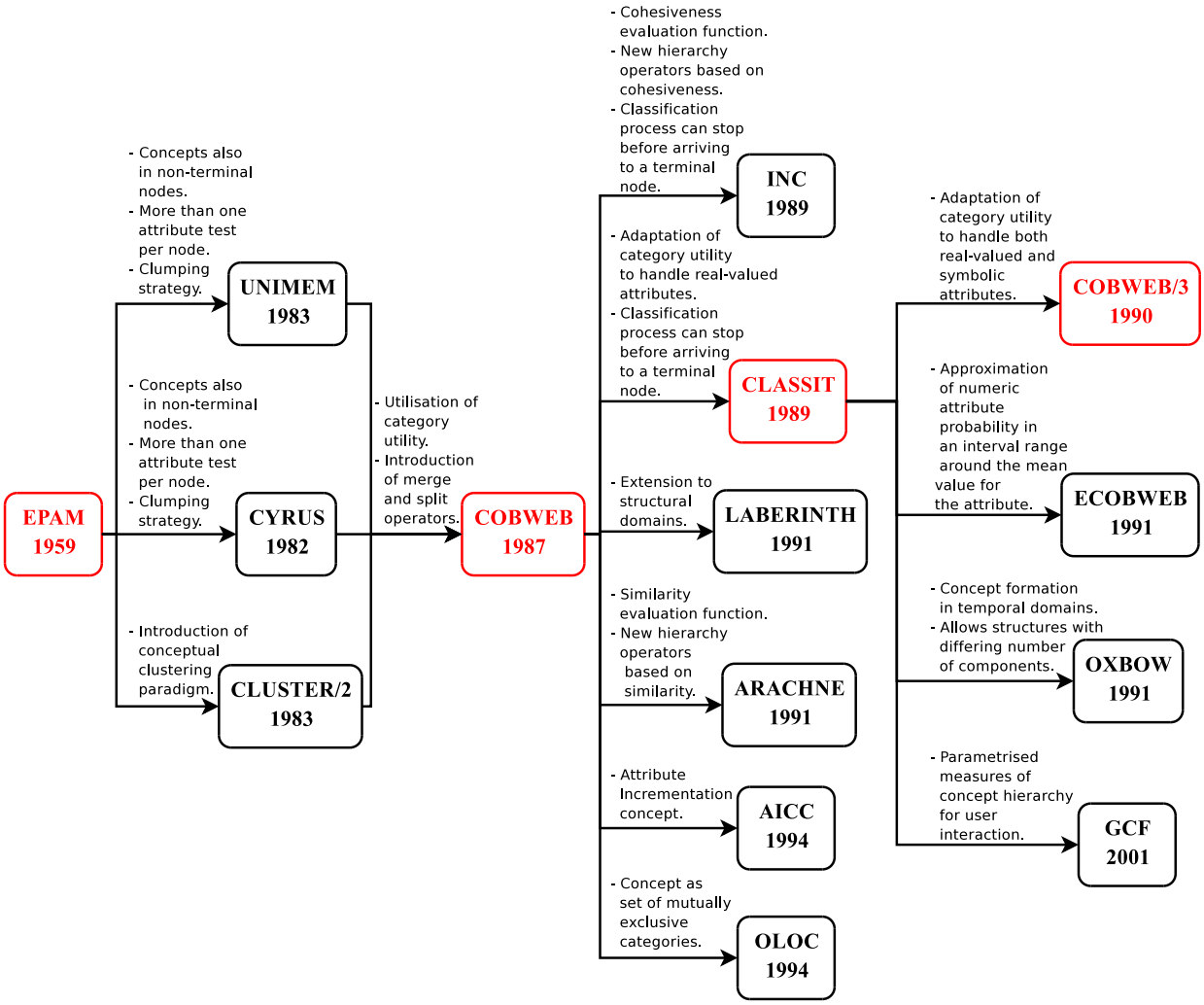


Figure 2.18: Evolution of Incremental Concept Formation domain. Most important contributions in the scope of the proposed learning approach are highlighted in red.

concept formation methods able to learn from concepts containing both numerical and symbolic attributes.

The precursor of the incremental concept formation domain is the EPAM algorithm for hierarchical clustering ([Feigenbaum 1959], [Feigenbaum and Simon 1962]), proposing a representation of instances or concepts as a set of attribute-value pairs (symbolic attributes). The concepts structure representation corresponded to a discrimination network, where different individual attributes where tested at different levels, and leaves represented the concepts.

In 1989, COBWEB has been proposed. COBWEB is an incremental concept formation approach based on three methods inspired in EPAM (UNIMEM [Lebowitz 1983], [Lebowitz 1985], [Lebowitz 1986], [Lebowitz 1987]), CYRUS [Kolodner 1983], and CLUSTER/2 [Michalski and Stepp 1983]). Its most important contribution with respect to previous work is the utilisation of the category utility function proposed by Gluck and Corter [Gluck and Corter 1985], to evaluate the quality of the obtained concept hierarchies. COBWEB has also introduced merging and splitting operators for concepts in the hierarchy.

As depicted in Figure 2.18 and presented in section 2.4.4, COBWEB has served as inspiration for several incremental concept formation models. For the scope of this thesis work, the most interesting method derived from COBWEB is CLASSIT because it adapts category utility for numerical attributes. Another interesting feature of CLASSIT is the introduction of the *cutoff* parameter to control the size of the hierarchy and to avoid the problem of data over-fit by having the possibility of halting the classification process in a node higher than terminal nodes level. Following the scope of this thesis, the most interesting extension of CLASSIT is COBWEB/3, which extends category utility to handle both numerical and symbolic attributes.

2.5 Event Learning from Video

This section explores the existing methods proposed for event learning in video. The analysis is focused in establishing the common learning techniques used in event learning, and to explore the approaches which have faced the challenge of bridging the gap between low-level video processing data and high-level complex event information.

In the latest years, video event analysis has become one of the biggest focus of interest in the video understanding community [Hu et al. 2004a], even if the number of studies in this area is still low. Several approaches have been proposed for the recognition and learning of events in video. The extraction of event information in video implies the proper processing of low-level video processing tasks, as motion detection, object classification, and tracking, in order to generate the appropriate input for the event analysis tasks.

Several approaches for video analysis have been focused in the recognition of pre-defined composite events, using a set of events extracted from visual features ([Howarth and Buxton 2000], [Medioni et al. 2001], [Piater et al. 2002], [Vu et al. 2006]). These methods have pre-defined ad-hoc methods for extracting events from low-level video tasks information. Then, these recognised events serve as building blocks for the recognition of also pre-defined composite events.

For example, Medioni *et al.* [Medioni et al. 2001] proposed an event recognition approach for an airborne moving platform. To make the link between low-level object features and the high-level behavioural events to detect, the authors propose an intermediate layer for

the extraction of object properties serving as input to the high-level behaviour analysis task. In this approach, behaviours to be recognised are pre-defined according to the application. The general structure of the approach is shown in Figure 2.19.

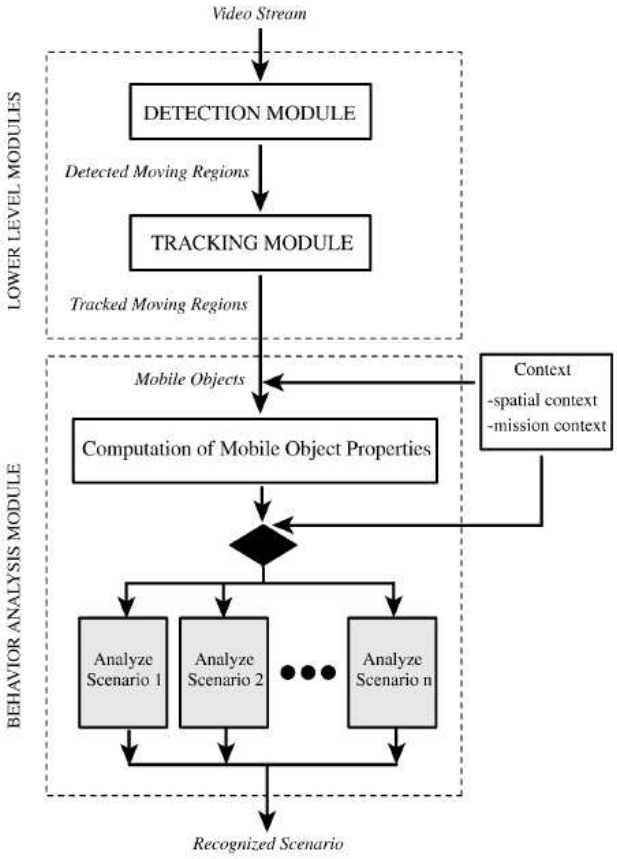


Figure 2.19: Overview of the behaviour analysis approach proposed in [Medioni et al. 2001].

Similarly, in [Ma et al. 2005], [Ma et al. 2006], the authors propose a method for detecting pre-defined primitive and composite events, which uses a feature vector accounting for instantaneous and temporal information about the objects evolving in the scene for representing an event. The authors also propose an unsupervised method for updating the set of defined events, by comparing the distance of a new event with the clusters formed with the existing events. If the new event is considered dissimilar, a new cluster is formed, and reported to the user for proper labelling.

In the context of event learning, several approaches have focused their interest on learning different elements of the events:

- Some approaches have focused in unsupervised learning of composite events, utilising pre-defined events (see Section 2.5.1).

- Other approaches have focused in unsupervised learning of both primitive and composite events, for specific object feature sets (see Section 2.5.2).

- The utilisation of incremental learning of events in video is almost inexistent in the video understanding literature (see Section 2.5.3).

2.5.1 Composite Event Learning

Several approaches have centred their attention on learning composite events in an unsupervised way ([Howarth and Buxton 2000], [Hongeng et al. 2000], [Hongeng et al. 2004], [Chan et al. 2004], [Chan et al. 2006b], [Chan et al. 2006a]). These methods search to enhance the recognition of composite events by an off-line training phase for learning the probabilistic and temporal parameters of these representations. A common method for representation of composite events is the Dynamic Bayesian Network (DBN) [Ghahramani 1998], which is a Bayesian Network [5] that represents sequences of variables in time.

More specifically, the Hidden Markov Models (HMM) are often utilised for this purpose (and variants of HMM), which can be considered as the most simple DBN. In a HMM, a sequence of observations is modelled by assuming that each observation depends on a discrete hidden state, and that the sequences of hidden states are distributed according to a Markov process[6].

For example, in [Hongeng et al. 2000], the authors use a variant of HMM for recognising composite events in a parking lot application. Also, in [Chan et al. 2004], [Chan et al. 2006b], [Chan et al. 2006a], authors propose an algorithm that solves event recognition. In this approach, event detectors are made ad-hoc, but composite events are learnt as DBNs trained with a standard Expectation Maximisation (EM) algorithm [Murphy 2002]. Authors apply semantic modelling early in the data processing chain, through the use of spatio-temporal semantic relations. This way, DBN representations are based on these relations rather than on low-level object attribute data, such as position (e.g. constructed with relational predicates such as *CloseTo* or *ContainedIn*, or with unary predicates such as *Moving*).

Other approaches are more interested in learning the sequences of events forming composite events ([Hamid et al. 2005], [Toshev et al. 2006]). In [Hamid et al. 2005], the authors propose an approach for learning composite events, which are described using

[5] A Bayesian Network is a probabilistic graphical model for representing conditional independences between a set of random variables. These networks are directed acyclic graphs whose nodes represent variables, and whose arcs encode conditional independences between the variables. They represent a particular factorisation of a joint distribution, where each variable is represented by a node in the network. A directed arc is drawn from node A to node B if B is conditioned on A in the factorisation of the joint distribution [Ghahramani 1998].

[6] A Markov process is a stochastic process in which the conditional probability for a state at any future instance, given the present state, is unaffected by knowledge of the past history of events.

a histogram account for the occurrence of pre-defined event sequences of length n, called n-grams. These composite events are clustered using a similarity measure made ad-hoc for the composite event representation, and used for anomalous activity detection. Also, in [Toshev et al. 2006], the authors have adapted a data mining algorithm called APRIORI [Agrawal and Srikant 1995], for automatically deducing the frequent composite events of a video, from a set of pre-defined events. Composite events are considered as patterns represented by a sequence of these events. The approach has been tested in parking lot sequences taking into account simple relational events of the type *Person_N in Zone_M*.

Until now, the presented methods, even if they perform learning at the composite events level, need to pre-define the events in order to construct the composite ones.

2.5.2 Primitive Event Learning

One step further is the application of machine learning techniques for also learning the events. This way, the task of pre-definition of events can become easier or even disappear.

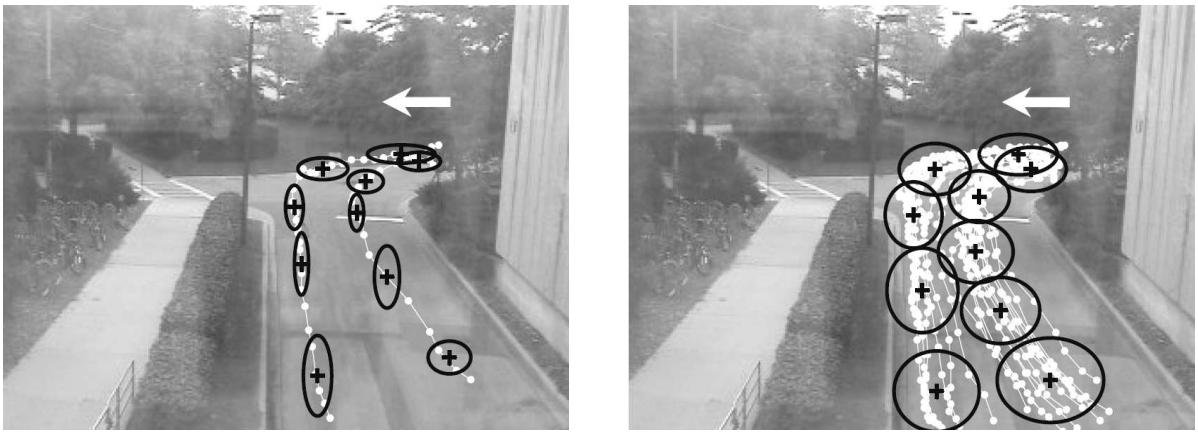
Existing approaches for primitive event learning have mainly focused in specific events to learn. The main motivations for these studies are the automatic generation of building blocks for composite events and the detection of unexpected events based on their frequency of occurrence in a video scene.

Mainly, the focus of research has been centred in learning trajectories ([Fernyhough et al. 2000], [Owens and Hunter 2000], [Remagnino and Jones 2001], [Hu et al. 2004b], [Hu et al. 2006], [Jiang et al. 2007], [Gaffney and Smyth 1999] , [Reulke et al. 2008], [Piciarelli et al. 2008]). A recent survey on trajectory learning for video surveillance applications can be found in [Morris and Trivedi 2008].

In this context, the work presented in [Remagnino and Jones 2001] utilises a HMM approach to model trajectory events occurring in a car-park environment. In their approach, a HMM behaviour representation is composed of states (to be in a region in the image), prior probabilities measuring the likelihood of an event starting in a particular region, the transitional probabilities capturing the likelihood of trajectory progressing from one region to another across the image, and the probability density function of each state. An expectation maximisation (EM) algorithm [Cadez et al. 2000] is employed to fit a number of Gaussian probability distributions, representing the states of trajectories to recognise, which are trained off-line from a set of all trajectory positions in a training dataset.

In [Jiang et al. 2007], a method for unusual event detection is proposed. First trajectories from a training dataset of trajectories considered normal, are grouped and fitted to an HMM, where the states are fitted to a Gaussian model of the position, obtaining a representation of trajectories as shown in Figure 2.20. After training these HMMs off-line unsupervised clustering is performed on them, merging HMMs considered similar. Those

clusters containing large number of samples (e.g., more than the average number) are chosen as normal pattern groups.



(a) (b)

Figure 2.20: HMM modelling of object trajectories as presented in [Jiang et al. 2007]. A 5-state HMM with Gaussian emission probability is fitted to the 2D trajectory feature vector $\{(x_1, y_1), (x_2, y_2), \ (x_T, y_T)\}$, where $\{x, y\}$ denotes the coordinate of object centre at every frame and T is the length of the trajectory. The black ellipses and crosses show the means and variances of every state.

Some other unsupervised event learning approaches have been proposed considering attributes different from those related to trajectory learning.

In [Galata et al. 2002], the authors propose an approach for detecting interaction events between pairs of objects. These interactions are represented as a sequence of feature vectors. A feature vector of a sequence representing an interaction between two objects corresponds to the velocity magnitude of the reference object, the vector representing the relative distance between the two objects and the velocity vector of the other object. A set of prototypical interactions is learnt off-line from these sequences of feature vectors by using a variant of the *Vector Quantisation* (VQ) algorithm [7], proposed in [Johnson and Hogg 1996]. Then, these prototypes become the events of a variant of a HMM which automatically infers the high level structure of typical interactive behaviours. The learnt behaviour model is then capable of recognising typical or atypical composite events within a scene.

Also, in [Xiang and Gong 2008], the authors propose a method for unusual event detection. For this purpose, the method performs clustering using a Gaussian Mixture Model (GMM)

[7] Vector Quantisation is a classical quantisation technique from signal processing which allows the modelling of probability density functions by the distribution of prototype vectors (also referred to as codebook vectors). It works by dividing a large set of vectors into groups having approximately the same number of points closest to them. Each group is represented by its centroid point. For further details, refer to [Gray 1984].

over a 7D features training set. Each feature is represented as the centroid $(\bar{x}, \bar{y})$ of the blob enclosing a detected object in the scene, the blob dimensions $(W; H)$, the filling ratio of foreground pixels within the bounding box associated with the blob, and a pair of first-order moments of the blob $(M_p x, M_p y)$. Then, behaviours are represented as Multi-observation HMM (MOHMM)[8] [Gong and Xiang 2003], using the cluster set resulting from the 7D features as the states of the HMM. Both learning methods are performed off-line.

In [Niebles et al. 2006], the authors propose a different way of performing unsupervised event learning. A video sequence is represented as a collection of spatial-temporal words by extracting space-time interest points. The algorithm learns from a set of video sequences the probability distributions of the spatial-temporal words and intermediate topics corresponding to human action categories automatically using a probabilistic Latent Semantic Analysis (pLSA) model[9] [Hofmann 1999]. The learnt model is then used for human action categorisation and localisation in a novel video.

2.5.3 Incremental Event Learning

Approaches which use incremental learning techniques are of special interest for the scope of this thesis. These techniques are currently almost inexistent in the literature.

In [Piciarelli and Foresti 2006], the authors propose a method for incremental trajectory clustering. Each input trajectory is represented by a list of vectors, which correspond to the spatial positions along the x and y axes in the 3D referential of the scene, ordered by the time of occurrence of these object positions. Trajectory clusters are represented in a similar way, but now the coordinates represent mean position at a given instant, and an approximative variance parameter is associated to each mean position coordinate. For performing trajectory clustering, the authors propose a distance measure, where the distance of a trajectory from a cluster is the mean of the normalised distances of every trajectory element from the nearest cluster element found inside a temporal window centred at the instant the trajectory element occurs. If a match occurs, the cluster is incrementally updated with the input trajectory data, considering a pre-defined update rate weighting the new data.

The authors propose an interesting tree representation of trajectory cluster prefixes, useful for abnormal trajectory detection. The authors refer to a cluster of similar trajectories as a class. Each class is split in a concatenation of clusters representing the initial common prefixes, so that classes can be represented with a tree structure, as in Figure 2.21.

[8] Compared to an HMM, in MOHMM the observational space is factorised by assuming that each observed feature is independent of each other. Consequently, the number of parameters for describing a MOHMM is much lower than that for an HMM.

[9] Probabilistic latent semantic analysis (pLSA) is a statistical technique for the analysis of general co-occurrence data which models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions.

The tree construction method is as follows: when a new trajectory is detected, a matching

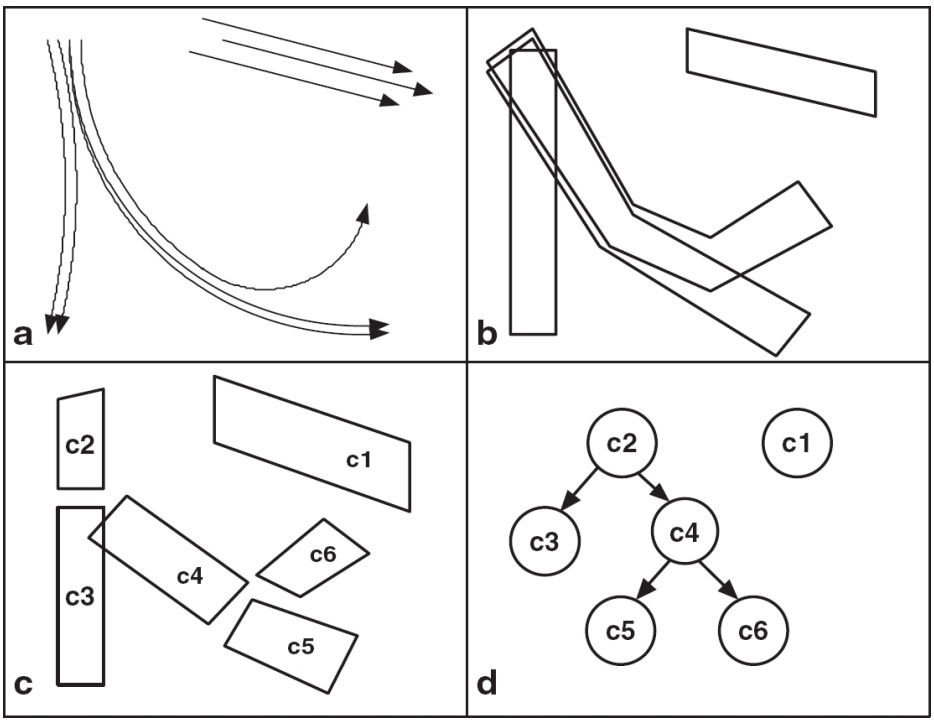


Figure 2.21: Representation of cluster prefixes for trajectories as in [Piciarelli and Foresti 2006]. (a) Trajectories; (b) classes; (c) cluster prefixes; (d) trees.

step is performed, in order to see if the trajectory matches any of the candidate prefix clusters. If the trajectory matches a prefix cluster, the prefix cluster is updated. If no match is found a new single-node tree is created. The prefix cluster associated to the new node is initialised with the data of the trajectory, while the variance is initially set to a fixed, pre-defined value. In the updating phase, the distance between the trajectory and the matched cluster is continuously checked to detect if the trajectory is moving too far from the cluster. If this happens near the end of the prefix cluster, no action is performed, otherwise the prefix cluster is split in two parts, the first one representing the part of the cluster that matches the trajectory and the second one representing the rest of the prefix cluster. In the tree view, the split of a node is modelled with the creation of a new child node, possibly inheriting all the children of the old node. In both cases, if the trajectory no longer matches a prefix cluster, a new matching step must be performed as described before, but this time the candidate nodes are all the children of the node from which the trajectory comes. This approach works only in specific structured scenes (e.g. road, path), performs learning only on spatial information (can not take into account time information), and do not handle noisy data.

Another approach for incremental event learning in video is the work presented in [Mugurel et al. 2000]. In this work, the authors propose an incremental event learning algorithm for classifying and learning the pattern of multiple tracked objects moving in a dynamic scene.

The pattern to be learnt is represented as a sequence of symbolic spatial relations among objects (e.g. in-front, behind, left-of, and right-of) at each time instance. For performing the incremental learning task, a variant of ILF(Incremental Learning with Forgetting)[10] algorithm is utilised, which is an approach for incremental concept formation similar to UNIMEM and COBWEB, that eliminates noisy instances. Unfortunately, the authors perform testing for just two aircrafts, which does not allow to appreciate the potentialities of the approach. Besides, the approach seems not scalable for a big number of objects, as spatial-relations will grow exponentially. Another limitation arises with the number of objects in the scene, which must be static during the analysed video scene, as relations are described according to pair of objects.

2.6 Discussion

Each task of the video understanding process has to solve specific problems. Each of these problems presents interesting and complex issues to the scientific community.

The first problem is to proper represent the detected moving regions in the scene, according to the objectives of a real world application. As discussed in Section 2.1, the choice of the right object representation plays a critical role and has a direct incidence in the processing time performance. Given the objective of this thesis, 3D primitive shape representation fits appropriately the processing time required by real world applications. The 3D shape representation also allows to easily define a large variety of objects with a better precision compared to the 2D primitive shape representations. Therefore, the issue is to find a representation which offers a good trade-off between precision and processing time.

One of the most challenging problems in video understanding is the MTT problem, as described in Section 2.2. This problem has been subject of thousands of publications proposing different approaches for its resolution. The main limitation of the MHT approaches (and also Monte Carlo methods, such as particle filtering) is the impossibility of considering the association of several moving regions corresponding to the same real object, as these methods were first designed for objects represented as a single point. Hence, it is necessary to address this problem in order to cope with situations related to more complex representations, as for example, when an object is visually detected as a set of separated moving regions or overlapping with another object. The tracking requirements in this thesis imply the generation of hypotheses similarly as the TOMHT, together with screening and pruning methods to achieve performances adequate for real world applications. Moreover, the dynamics models of multi-target tracking approaches do

[10]ILF [Lazarescu et al. 1999] is an incremental learning algorithm that builds compact conceptual hierarchies and tracks concept drift. The concept drift means that the statistical properties of the target changes over time in unforeseen ways.

not handle properly noisy data. Therefore, the shape representation information should be combined with reliability measures to generate a new dynamics model which takes advantage of these measures.

In the context of incremental concept formation, described in Section 2.4, the interest is focused on their capability of incrementally learning a hierarchy of concepts according to the arrival of new information. This characteristic is suitable with the objective of the proposed video understanding framework for incremental learning events in real world applications. These techniques were not conceived for representing temporal relations between concepts, then an extension of these methods is necessary for considering these relations. Also, the incremental concept formation models found in the literature do not consider explicitly the quality of the information utilised in the learning process, even if some techniques consider the possibility of missing object attributes.

Moreover, the design of a complete video understanding framework for event learning implies the resolution of a wide variety of complex problems. For the complete framework, the problem of obtaining reliable information from video concerns the proper treatment of the information in every task of the video understanding process. For solving this problem, each task has to measure the quality of the information in order to evaluate the overall reliability of a framework. As presented in Section 2.3, the reliability measures have been used in the literature for focusing on the relevant information, 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. Then, the problem of globally estimating the quality of the information utilised by a video understanding framework is still an open problem.

The currently existing event learning approaches in video understanding, presented in Section 2.5, show the increasing interest of the community in this area. The efforts are mostly focused on unsupervised learning rather than supervised learning and they only handle specific events. Moreover, the few existing unsupervised techniques often perform learning off-line. Thus, the problem of incremental learning of general events remains an open problem in the video understanding domain.

Next Chapter 3 presents the video understanding framework for incremental event learning, in order to understand the process as a whole, its interactions with the user, and the resulting output from the learning approach. This section will also serve as an introduction of next sections, explaining the different processes involved in the proposed video understanding framework.

Chapter 3

Thesis Overview

In this thesis, a new video understanding framework for incremental event learning is proposed. As depicted in Figure 3.1, the proposed video understanding framework is conceived for obtaining a hierarchical description of the events induced by the objects evolving in the scene, together with the recognised events in which these objects participate, starting from noisy image-level data.

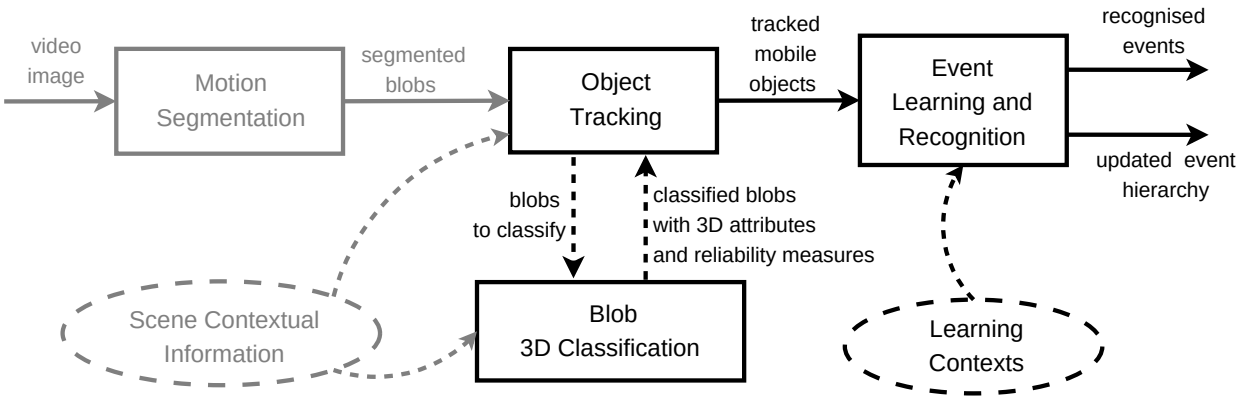


Figure 3.1: Proposed video understanding framework for event learning. Black elements correspond to the thesis contributions. Gray elements correspond to elements used by the proposed framework, but not forming part of the contributions.

The video understanding process begins with a segmentation task which processes a video frame and returns the motion regions occurring in the scene. These regions are processed by a tracking task in order to extract the information of mobile objects present in the video scene, and reliability measures are associated to the extracted information in order to account for the quality and coherence of this information.

Then, the obtained information and reliability measures are utilised by an event learning and recognition task in order to incrementally update a hierarchical structure of learnt states and events, and to recognise the events for each object evolving in the scene. The learning process is performed without any prior information about the events to be detected in the scene. A state represents the occurrence of a set of object attribute values, while an event represents the transition between two states.

The object attribute information consists of 2D and 3D position and dimensions, where 3D attributes are inferred from generic 3D object models of the objects expected in the scene (e.g. a person, a car). The tracking task obtains this 3D information by interacting with a blob classification task, which associates 3D model instances to the blobs, together with reliability measures associated to the attributes describing an instance of these 3D models.

The objective of this chapter is to give a general overview of the proposed video understanding framework, describing the different problems to be solved by the approach, and also giving a first insight about the proposed solution at each task of the video understanding framework.

This chapter is organised as follows. First, Section 3.1 defines the terminology utilised in this thesis. Second, Section 3.2 focuses on describing the proposed video understanding framework, the possible interactions between the framework and the user, and a description of the platform utilised for the development of this framework. Third, Section 3.3 introduces the proposed blob classification approach utilised to associate a generic 3D object representation to a blob, focusing on the issues arising from the classification problem. Fourth, in Section 3.4, the proposed multi-object tracking approach is presented, giving a first insight about the way of solving different issues for the tracking problem. Fifth, Section 3.5 introduces the proposed incremental event learning algorithm. Sixth, the different possible interactions of the user with the video understanding framework are described in Section 3.6. Finally, in Section 3.7, general remarks about the video understanding framework are discussed.

3.1 Terminology

In the context of this thesis, several concepts must be appropriately defined in order to clarify some discrepancies in the event and machine learning terminology utilised in the literature. First, in the context of events, the following concepts are defined, using the event ontology presented in [Brémond et al. 2004]:

Definition 3.1 *A **state** is a spatio-temporal property valid at a given instant or stable on a time interval. A state can characterise several mobile objects.*

Definition 3.2 *An **event** is one or several state transitions at two successive time instants or on a time interval.*

Definition 3.3 *A **primitive state** is a spatio-temporal property valid at a given instant or stable over a time interval which is directly inferred from the visual attributes of physical objects computed by vision routines.*

Definition 3.4 *A **primitive event** is a primitive state transition. It represents the finest granularity of events.*

Definition 3.5 *A **composite state** is a combination of primitive states.*

Definition 3.6 *A **composite event** is a combination of primitive states and events. This is the coarsest granularity of events. Composite events are also known in video understanding literature as complex events, behaviours, and scenarios, among other names.*

In the context of machine learning, the following fundamental concepts for this thesis are defined, based on the differences established in [Gennari et al. 1990]:

Definition 3.7 ***Supervised learning** is a machine learning approach in which an algorithm generates a function that maps inputs to desired outputs. To describe the desired outputs, the algorithm is trained using a labelled training set (i.e. output is known for each instance).*

Definition 3.8 ***Unsupervised learning** is a machine learning approach in which an algorithm models a set of inputs. It differs from supervised learning in that information about the output of instances is not known (i.e. the training dataset is unlabelled). Instead, similarity or distance measures between instances are defined to guide the learning process.*

Definition 3.9 ***Incremental learning** is an unsupervised learning approach in which an algorithm models a set of inputs, with the information obtained so far as an ongoing process. This approach can dynamically generate new concepts, and interleave learning and performance, as it is intended to learn from instances one at time without extensive reprocessing of previously encountered instances.*

Definition 3.10 ***Concept Clustering Problem [Michalski and Stepp 1983]:***

- ***Given:** A sequential presentation of instances and their associated descriptions.*
- ***Find:***
  1. *Clusterings that group those instances into categories.*
  2. *A user-guided definition for each category that summarises its instances.*
  3. *A hierarchical organisation for those categories.*

Definition 3.11 ***Incremental concept formation models** are incremental learning approaches for solving the concept clustering problem. These formation models allow to incrementally build a concept hierarchy based on incomplete or uncertain data, by updating the hierarchical concept structure with the arrival of each new data instance. They also allow the classification of a new instance, based on the inferred concepts from previously processed data.*

For simplicity, from now, *primitive states* will be referred simply as *states*, while *primitive events* will be referred as *events*. In order to avoid misunderstandings, the *composite events* will keep their denomination.

3.2 Video Understanding Framework for Event Learning

The design of general and robust video understanding techniques is still an open problem. Providing robust information from noisy videos can be a very complex problem, as several issues of different nature can complicate this task. For instance, these issues can be associated to the quality of the analysed video (e.g. bad contrast), the complexity of the scene (e.g. illumination changes, strong shadows, cluttered scene), the number of mobile objects evolving in the scene (e.g. a crowd), or the interactions of the mobile objects with the scene and with other mobile objects (e.g. static and dynamic occlusion), among other issues.

All these factors can induce to errors a video understanding approach due to the ambiguity of the visual evidence. Therefore, in order to achieve a robust video understanding process, it is first necessary to measure the quality and coherence of the acquired information.

For coping with this problem, a new video understanding framework for learning frequent events occurring in a noisy video scene is proposed. This approach involves a complete framework for event learning, in order to cope with noisy environments.

Section 3.2.1 gives a detailed general description of this video understanding framework for event learning and recognition. Then, the video understanding platform utilised to develop this framework is presented in Section 3.2.2.

3.2.1 Video Understanding Framework Process

The proposed video understanding framework follows a bottom-up process to obtain high-level temporal information, starting from low-level image data. This process is depicted in Figure 3.2.

The video understanding framework receives as input a sequence of images. A segmentation task is applied to each image frame to detect motion in the scene, obtaining a set of moving regions represented as the bounding boxes enclosing them (called *blobs* from now on). In particular, a background subtraction method called thresholding [Heikkila and Silven 1999] is used for segmentation, which basically consists in comparing intensity and colour information on the currently analysed frame, with a reference background image (for further information about background subtraction methods refer to [McIvor 2000]). No further details about the utilised segmentation method is given in this thesis, as segmentation is not part of the contributions (as depicted in Figure 3.1). In fact, for the

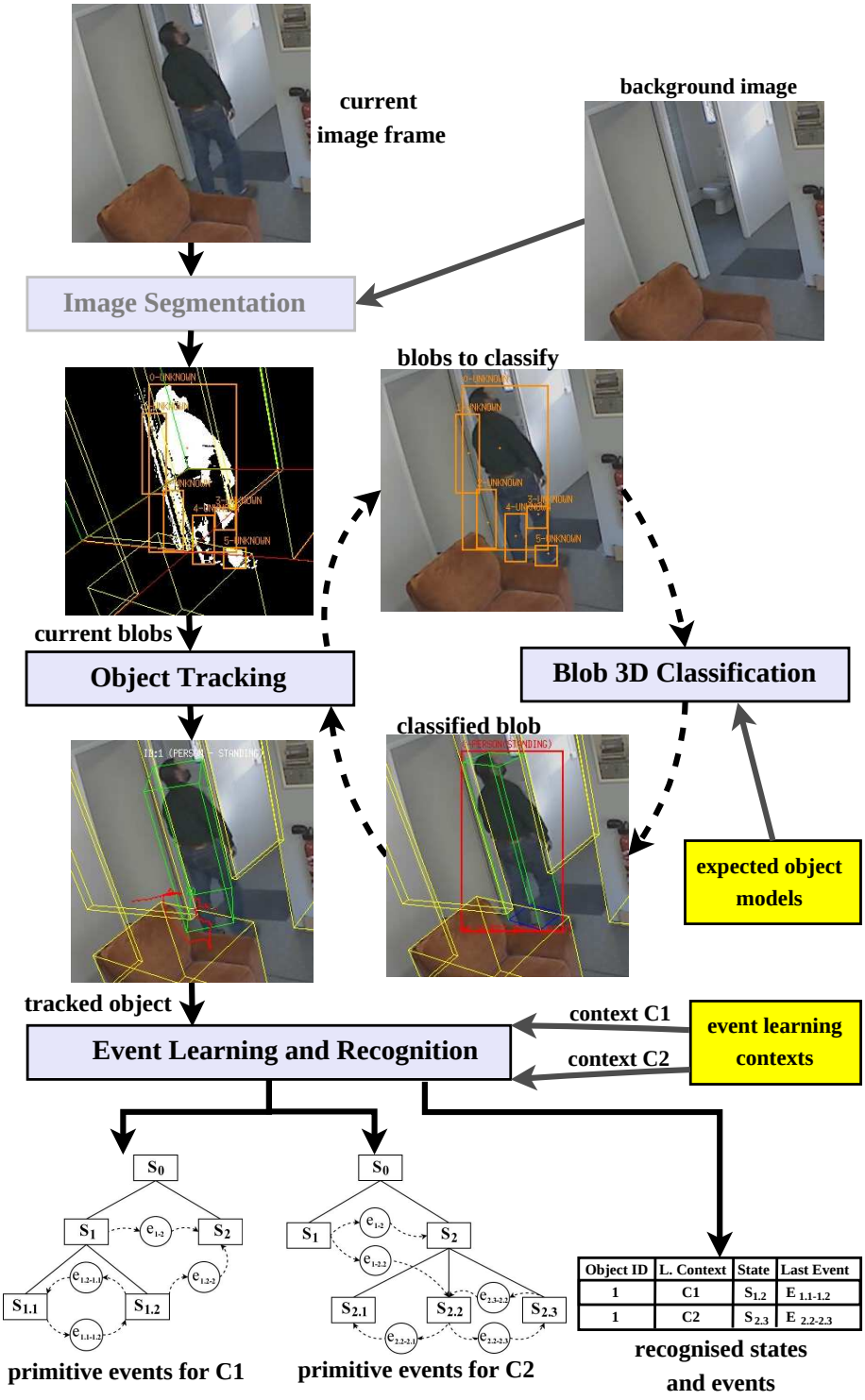


Figure 3.2: Data-flow of the video understanding process.

proposed approach, any segmentation method which gives as output a set of segmented blobs can be used in the video understanding framework.

Then, taking as input the set of obtained blobs from the segmentation task, a multi-object tracking task is performed. This task utilises a new tracking approach which combines blob 2D information, together with 3D information obtained from the parallelepiped object representation, to generate a set of mobile object configuration hypotheses. The approach efficiently estimates the most likely tracking hypotheses in order to manage the complexity of the problem with a high processing time performance. 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.

For obtaining the 3D information associated to a mobile object, the tracking algorithm interacts with a new proposed generic 3D classifier which associates to each processed blob an object class, 3D attributes, and visual reliability measures of these 3D attributes. The representation used by this classifier corresponds to a generic primitive 3D shape which consists in a parallelepiped described by its 3D width, height, length, position, and orientation with respect to the plane of the 3D referential of the scene. This representation is calculated using pre-defined camera calibration information determined by an off-line process, and pre-defined 3D models of expected objects in the scene. This classifier is described with more detail in Section 3.3.

The reliability measures obtained with the classification task are utilised by the tracking task in a new attribute dynamics model, which takes into account these measures as a way of quantifying the quality of the estimated attributes. This dynamics model utilises the visual reliability measures calculated for the parallelepiped model to weight the contribution of new attribute information in the calculation of the attribute estimation associated to a mobile object. This way, reliable information is enforced in the dynamics model, contributing to the robustness of the approach by handling noisy data.

The tracking task gives as result the set of the most likely mobile object hypotheses, including full description of object attributes and reliability measures. For more details about the multi-object tracking approach, refer to Section 3.4.

Finally, a new event learning algorithm takes as input the information about tracked objects in the scene, together with pre-defined learning contexts information, to learn the frequency of events occurring in the scene. The learning contexts define the attributes of interest to be considered for event learning. Multiple learning contexts are allowed simultaneously, to allow the analysis of more than one context of interest at the same time.

The event learning approach is based on incremental concept formation models, which give as result a hierarchy of concepts, with information about the probability of occurrence of these concepts. In the context of this thesis, a concept corresponds to a state and the

data to be learnt corresponds to the visual attributes of mobile objects present in the video scene. Then, the probability associated to each concept can be interpreted as the frequency of occurrence of a state.

For enabling event learning, the hierarchical representation proposed by the models of incremental concept formation is extended to represent the transitions between different states. The new incremental learning algorithm expands the representation of concepts to the first-order temporal relations between these concepts. This way, in the context of the proposed learning approach, concepts represented as nodes in the hierarchy become the learnt states induced by the tracked objects present in the scene, while the first-order temporal relations, representing the change of a state to another, become the learnt events. This way, the learning approach is able to incrementally generate a hierarchical representation of the occurrence of states and events in the scene, together with information about their frequency of occurrence.

A hierarchy of states and events is learnt for each pre-defined learning context, and the current state and event for each object evolving in the scene is recognised. The process of learning and recognition occurs simultaneously, as the utilised learning approach is incremental. The information about the frequency of occurrence of the states and events allows the system to detect abnormal events occurring in the scene. This event learning approach is introduced in more detail in Section 3.5.

Next Section 3.2.2 describes the platform utilised for the development of the proposed video understanding framework for event learning.

3.2.2 Video Understanding Platform

In order to develop the proposed video understanding framework, the platform for Video Understanding SUP (Scene Understanding Platform) [Avanzi et al. 2005] has been utilised. SUP has been developed by PULSAR Team (former ORION Team) at INRIA (Institute National de Recherche en Informatique et Automatique), Sophia Antipolis, France. SUP is a generic environment for video processing algorithms which allows to flexibly combine and exchange various techniques at the different stages of the video understanding process.

SUP platform has been initially conceived as an implementation of a two-steps approach, consisting of a vision module, followed by a behaviour patterns detector. In this two-steps approach, the visual module is used to extract visual cues and events. Then, this information is used in the second stage for the detection of more complex and abstract behaviour patterns [Toshev et al. 2006].

By dividing the problem into two sub-problems simpler and more domain-independent techniques can be used at each step. The first step makes usually extensive usage of stochastic methods for data analysis while the second step conducts structural analysis

of the symbolic data gathered at the preceding step (see Figure 3.3(a)). Examples of this two-level architecture can be found in the work of [Ivanov and Bobick 2000] and [Vu et al. 2003].

At the first level, SUP extracts primitive geometric features like moving regions. Based

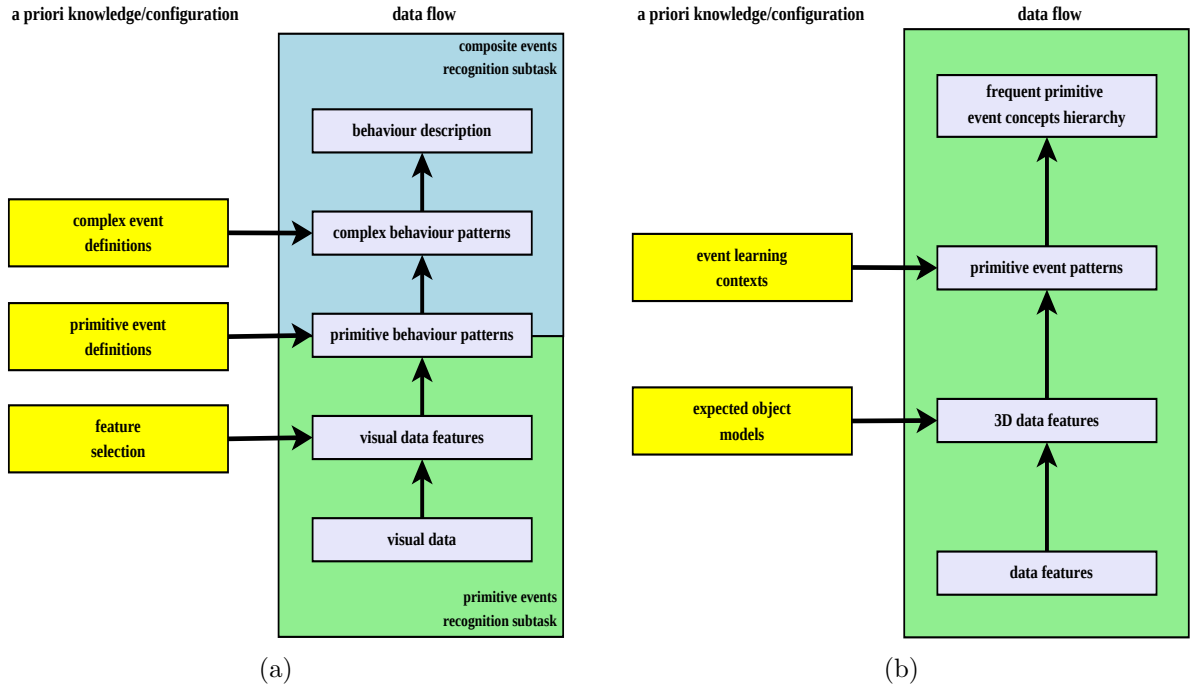


Figure 3.3: Contrasted video understanding architectures for video understanding. The steps depicted in each figure describe the data flow during the video understanding process. Figure (a) depicts a two-level architecture of a video understanding system. Figure (b) depicts the proposed video understanding framework, which bridges the gap present in the two-level architecture.

on them, objects are recognised and tracked. At the second level those events in which the detected objects participate, are recognised. For performing this task, a special representation of events is used which is called event description language [Vu et al. 2003]. This formalism is based on an ontology for video events presented in [Brémond et al. 2004] which defines concepts and relations between these concepts in the domain of human activity monitoring. The major concepts encompass different object types and the understanding of their behaviour from the point of view of the domain expert.

Two-level architectures introduce a gap between low-level information associated to visual data and high-level information associated to events. For this type of architecture, this gap has been often bridged utilising events pre-defined by the user. This way, low-level data are carried to a higher conceptual level, defined by the knowledge of the user.

In contrast, the proposed video understanding framework, defines an architecture which

automatically bridges the gap between visual and conceptual information, by learning the frequent events occurring in the analysed video scene (see Figure 3.3(b)), which are represented as attribute distributions. These events can be seen as primitive temporal concepts which can be used as building blocks for the detection or learning of more complex events.

Next Section 3.3 introduces the method for 3D object classification, utilised by the proposed video understanding framework to associate 3D information and reliability measures to a set of 2D blobs.

3.3 3D Object Classification

The main issues arising from the classification problem are both inherent to the classification problem itself and to the issues carried out from the segmentation problem. The main issues of the classification problem are related to the *ambiguity* of visual evidence, for instance, the object appearance changes with respect to its orientation, position relative to the camera, and posture in the case of persons and animals, or the same visual evidence can represent more than one object if no sufficiently discriminant information is available.

Binocular visual perception allows human beings to perceive depth of their environment. At the same time, a person can shut one of his/her eyes and still preserve the depth sensation, without loosing too much of precision on depth estimation of the focused object. This capability is a consequence of the interpretation that the brain performs about the new visual information, by associating it to similar environments or objects previously observed, and then concluding on its nature and 3D shape. This means that the brain uses a priori knowledge to conclude about the attributes (e.g. position, dimensions) of an observed object.

Following this idea, a new object representation using a simple generic 3D primitive shape model of the expected objects in the scene is proposed. This model allows to represent objects of different nature in a way that is independent from the relative position between the object and the camera. More specifically, the proposed representation corresponds to a parallelepiped model 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 3.4. Also, visual reliability measures of the three estimated dimensions are proposed, which represent a measure of their visibility with respect to the camera and static occlusion. These measures have been mainly proposed to aid the tracking and learning tasks of the video understanding framework.

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

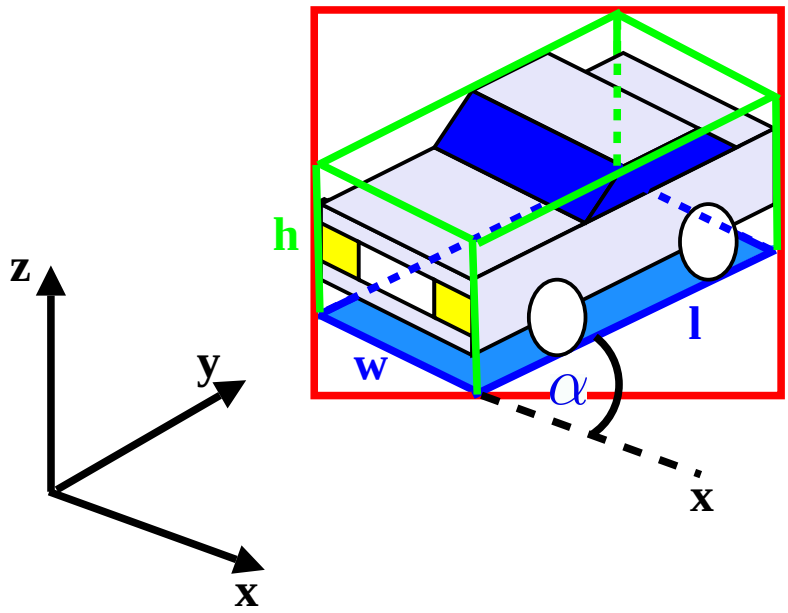


Figure 3.4: Example of a parallelepiped representation of an object. The figure depicts a vehicle enclosed by a 2D bounding box (coloured in red) and also by the parallelepiped representation. The base of the parallelepiped is coloured in blue and the lines projected in height are coloured in green. 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.

of the analysed scene. Starting from the basis that a moving object will be detected as a 2D blob, 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 parallelepiped are defined as the parallelepiped dimensions w, l, and h described by a Gaussian distribution representing their probability of occurrence for a given object class, together with a minimal and maximal value for each dimension.

The initial problem of determination of a parallelepiped enclosing a moving object has six degrees of freedom (d.o.f.): two d.o.f. for parallelepiped position with respect to the ground of the 3D referential of the scene, three d.o.f. for 3D dimensions of the parallelepiped, and one d.o.f. for the parallelepiped orientation.

The four 2D constraints imposed by the blob (bottom, top, left, and right limits with respect to the image frame) allow to finally describe the four (x, y) base points of the parallelepiped in terms of h and α attributes. Then, an optimisation step based on the pre-defined parallelepiped models of expected objects in the scene is performed, obtaining as result the most likely parallelepiped models for each class represented by the pre-defined models.

These parallelepiped models consist in a Gaussian representation for the w, l, and h

attributes of the parallelepiped, representing the probability of different 3D dimension sizes for a given object. In the case of objects changing their posture (e.g. a person), a set of parallelepiped sub-models is defined representing each posture of interest for the represented object. Then, for the optimisation step, the likelihood of a parallelepiped instance with respect to a pre-defined model is calculated as the multiplication between the dimensional probabilities.

The utilised representation tries to cope with several limitations imposed by 2D representations, but keeping its capability of being a general model able to describe different objects, and a performance adequate for real world applications. Object 2D primitive shapes can be efficiently computed, and then they are the most suitable representation for real world tracking applications. These representations have several advantages which justify their use. For certain applications, two dimensions are enough to describe the objects involved in the analysed scene, because:

1. the relative position between the camera and the observed object hides one dimension (e.g. tracking groups of people in a metro scene [Cupillard et al. 2001]), meaning that it can be enough to model a 3D object with a 2D model,

2. the estimation of the another dimension is performed by merging information from different cameras (e.g. human posture detection [Cucchiara et al. 2005b], apron monitoring application on an airport [Borg et al. 2006]), and

3. object detection can be more interesting than classification for certain applications (e.g. detection of stopped vehicles in a highway [Cucchiara et al. 2005a]). Certainly, the processing time spent in calculating the attributes associated to 2D representations is inexpensive, allowing to obtain a good performance for real world applications. These 2D models are sufficient to find the 3D position of an object, which is enough for certain applications.

Nevertheless, 2D representations present also several drawbacks, which make them useless for many situations:

1. If the 2D moving region considerably changes its appearance depending on its position relative to camera (see Figure 3.5), dimensional estimation becomes unreliable.

2. If the 2D representation considerably changes when the object rotates (see Figure 3.6), dimensional estimation becomes also unreliable.

3. For deformable objects (e.g. persons changing their posture), it would become a very hard task to define a 2D representation for each possible deformation of an object of this nature, considering that it can also change according to different positions relative to camera and different object orientations.

On the other extreme, different models have been proposed for specific objects (e.g. persons, vehicles), which are application and object dependent. Some authors use

(a) (b)

Figure 3.5: 2D moving region changes by different positions of an object relative to the camera. Here, the same person (with same posture) is represented by very dissimilar 2D regions in the same video sequence. In Figure (a) the person is far from the camera and it is possible to see her/his height, while in Figure (b) the person is seen almost from top and almost nothing can be said about his height.

precise models of a specific object to perform detection. These models allow generally to obtain quite good detection rates and attribute estimations, but the computational cost associated to its utilisation is often too expensive to be used in real world applications. [Black et al. 1997] uses a 2D model of each body part of a human constrained by image motion parameters to perform tracking of walking persons and human gestures. [Boulay et al. 2006] uses a very precise 3D model of human to detect postures. In this work, a human posture is described by a set of 23 parameters, subject to bio-mechanical constraints. This human model is used to generate silhouettes to be compared with the one detected for a person in the scene.

The proposed parallelepiped model representation allows to quickly determine the type of object associated to a moving region and to obtain a good approximation of the real 3D dimensions and position of an object in the scene. This representation tries to cope with the majority of the limitations imposed by 2D models, but being general enough to be capable of modelling a large variety of objects and still preserving high efficiency for real world applications.

The characteristics of this new object representation are listed below:

1. A representation independent from the camera view and the orientation of the object with respect to the 3D referential of the scene.

2. A model which instances can be quickly obtained, with better precision than 2D representations, providing 3D object features which are more interesting for event analysis tasks.

(a) (b)

Figure 3.6: 2D moving region shape changes because of a change in the object orientation. Here, the same car is represented by very dissimilar 2D regions in the same video sequence. In Figure (a) a car is seen from its back part. Later, the car rotates to park and it is seen from its right part, as seen in Figure (b).

3. A simple generic object representation model which allows users to easily define new mobile objects expected to be present in the scene.

4. Reliability measures proposed to calculate the visibility of the obtained 3D object features, accounting for occlusion situations and camera view.

For further details about this new 3D object classification approach, see Chapter 4.

In next Section 3.4, the proposed multi-object tracking algorithm in the context of the video understanding framework is introduced. This tracking method uses the proposed object representation to perform tracking of object 3D features, and to take advantage of the reliability measures associated to them.

3.4 Multi-target Tracking using Reliability Measures

The object tracking problem presents the most challenging issues to the video understanding community. Among the most known issues are dynamic and static object occlusion (partial visibility of an object), multiple objects tracking, and the problems derived from poor object segmentation.

Many approaches have been proposed to manage all the possible tracks that can occur for multiple objects tracking ([Gordon et al. 1993], [Isard and Blake 1998], [Doucet et al. 2001], [Hue et al. 2002a], [Hue et al. 2002b]). These methods often generate an exponential number of hypotheses increasing with the size of the state space. Also, they scale poorly as the dimensionality increases due to large number of objects to be tracked. As a

consequence, an accurate dynamic model is required in practise to reduce the number of samples needed for accurate modelling.

A new method for tracking multiple objects present in a video is proposed. This method is focused on monocular static cameras and it is dedicated to real world applications. This method maintains a list of likely configuration hypotheses for the mobile objects present in the scene. The proposed tracking method has been developed to cope with a wide range of typical issues present in videos with multiple objects, such as, segmentation errors (e.g. due to shadows, or weakly contrasted objects), cluttered scenes, and dynamic occlusions. These tracking issues are major challenges in the vision community [Society 2007].

The capability of the tracking approach for coping with these issues depends on the reliability of the attributes estimated in the video frames processed before the occurrence of one of the mentioned issues. This means, that the tracking approach will be able of solving these issues if the temporal coherence and quality of the object attributes previously estimated, calculated as reliability measures, is high enough to utilise this attributes in the resolution of these issues.

This new method efficiently estimates the most likely tracking hypotheses in order to manage the complexity of the problem with a good computer performance. This approach combines blob 2D information, together with 3D information obtained from the object representation presented in previous Section 3.3, to generate a set of mobile objects configuration hypotheses. These hypotheses are validated or rejected in time according to the information inferred from previous frames, and combined with the information obtained from the currently analysed frame.

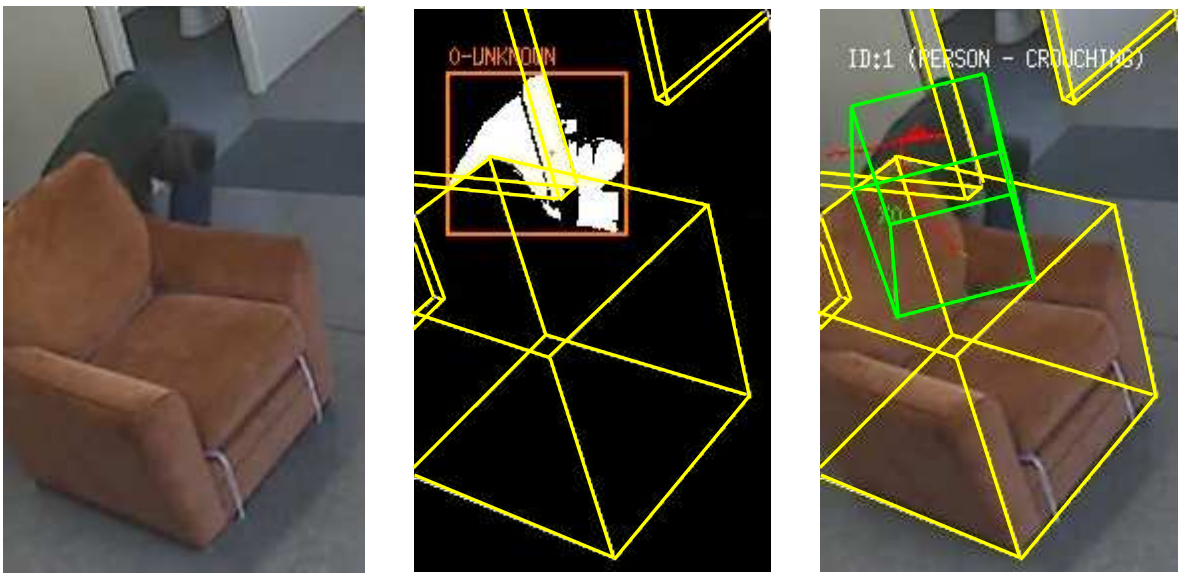
Each mobile object is represented as a set of statistics of features inferred from visual evidences of their presence in the scene. The hypotheses are grouped according to their visual relations in scene in order to separate the tracking procedure into different tracking ambiguities. Each group of hypotheses is updated according to the visual evidences obtained in later frames, expanding the hypotheses group to account for different possible mobile object tracks. The generation of new hypotheses for tracked objects has been carefully designed to immediately generate the best possible hypotheses in order to improve the processing time performance.

The reliability measures obtained with the classifier introduced in previous Section 3.3 are utilised in a new proposed attribute dynamics model, which takes into account these measures to quantify the quality of the estimated attributes. This dynamics model utilises the visual reliability measures calculated for the parallelepiped model, introduced in previous Section 3.3, to weight the contribution of new attribute information in the calculation of the attribute estimation associated to a mobile object. This way, reliable information is enforced in the dynamics model, contributing to the robustness of the approach by handling noisy data. Also, a cooling function is utilised in order to

diminish the contribution of old information, and highlight the contribution of the newest information.

The proposed tracking approach is able to cope with several issues common to multi-object tracking techniques:

- **The static occlusion problem:** This problem occurs when a tracked object is partially or totally occluded by the image frame border or static objects present in the scene (Figure 3.7). The proposed tracking approach solves this issue by maintaining the temporal coherence of the expected tracked object attributes, and determining which portion of the object can be occluded by a border or a static object (Figure 3.7(c)).

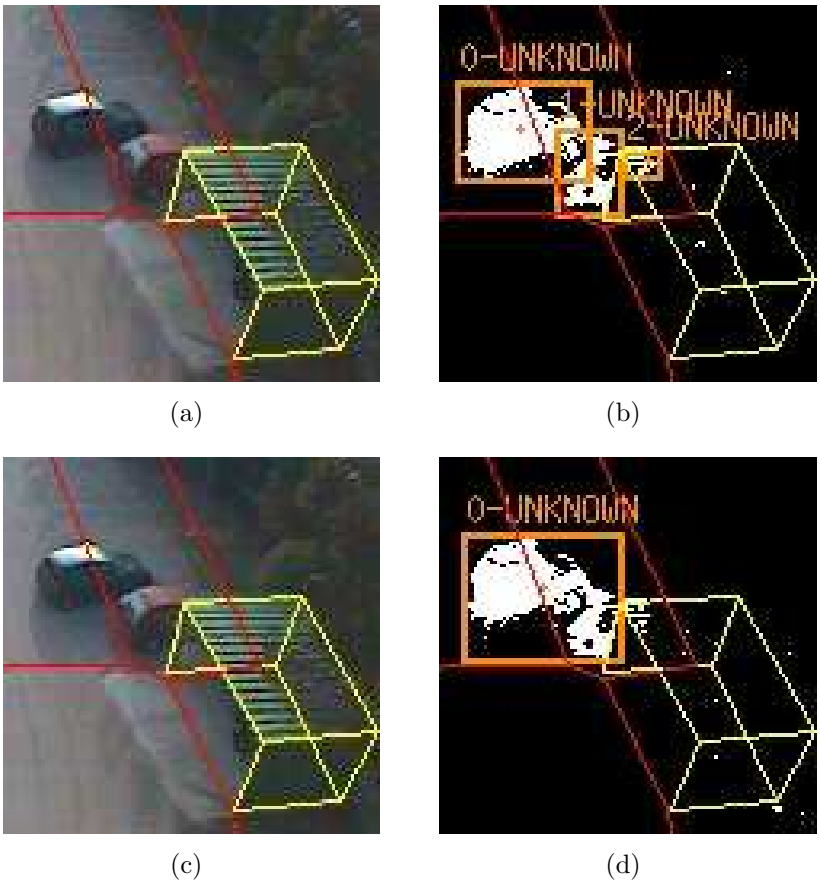


(a) (b) (c)

Figure 3.7: Example of a static occlusion situation, and how this issue can be solved with the proposed tracking approach. Figure (a) shows the original frame, where a crouched person is occluded by a couch. Figure (b) shows the segmentation result, where resulting blob is depicted in orange and moving pixels in white. Yellow lines represent the pre-defined static objects in the scene. Note that a big portion of the tracked person is occluded by the couch. In Figure (c), the solution found by the proposed tracking algorithm for this problem is depicted. The 3D parallelepiped representation is coloured in green, while object trajectory is represented by a sequence of points in red, connected by red segments. Last point in trajectory represents the expected position for next frame, which is connected by a green segment.

- **The dynamic occlusion problem:** This problem occurs when a tracked object is partially or totally occluded by another tracked object (Figure 3.8). The problem arises when the segmentation process is not able to separate a set of objects,

which are near to each other. The proposed tracking approach solves this issue by maintaining the temporal coherence of the set of occluding objects, and checking the validity of the possible solutions in terms of 3D model collisions. As the tracking approach does not use object appearance information, it can solve dynamic occlusion situations where involved objects maintain the temporal attribute coherency during the occlusion situation. One of the considered aspects in future work is the inclusion of object appearance models for coping with more complex dynamic occlusion situations.



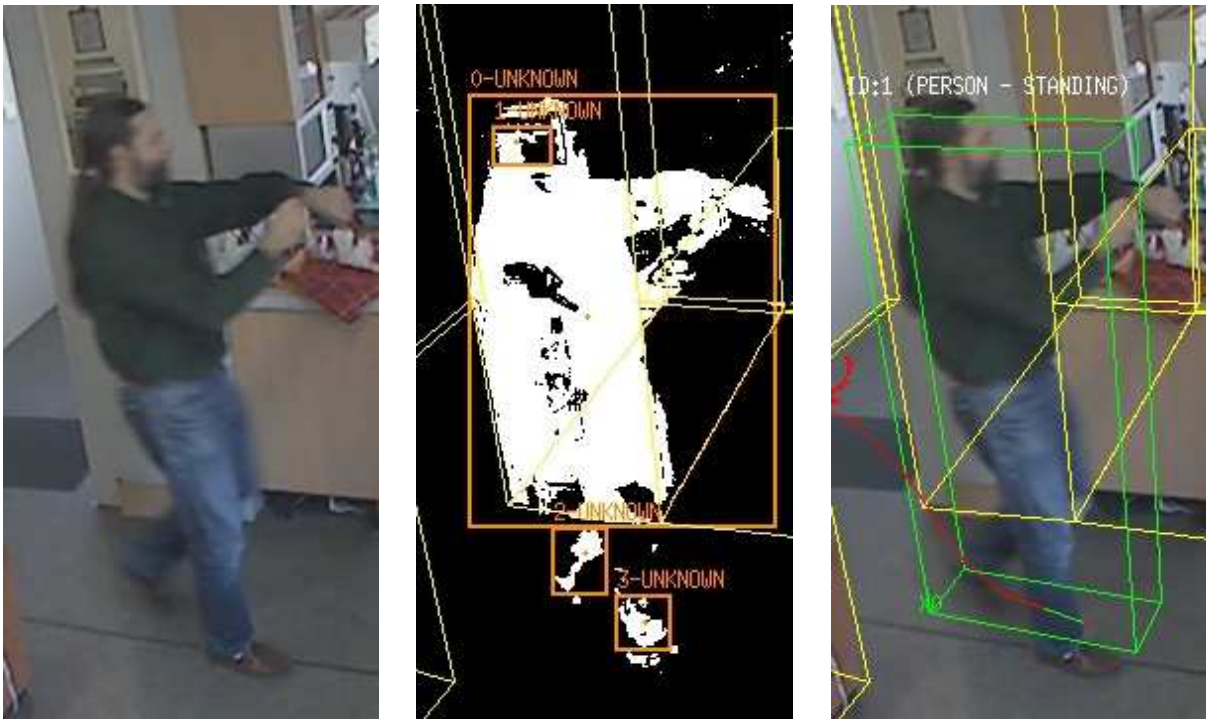
(a) (b)

(c) (d)

Figure 3.8: Example of a dynamic occlusion situation for two vehicles in a parking lot application. Figures (a) and (b) show an image frame where the two vehicles are still separately segmented. Figures (c) and (d) show the next image frame where segmentation is not able to separate the object. In this case, the temporal coherence on vehicle attributes can be exploited in order to solve this dynamic occlusion situation. In images (b) and (d), the resulting blobs are depicted in orange and moving pixels in white. For the four images, yellow lines represent the pre-defined static objects, and red lines represent the zones of interest in the scene.

- **Low contrasted objects and illumination problems:** These problems leads to missing object parts, several separated parts, or an over-segmented object (Figure

3.9). More precisely, these situations are commonly caused by illumination changes, low object contrast with respect to background, and object shadows and reflections, among other situations. The proposed tracking approach solves these issues by maintaining the temporal coherence of the tracked object, evaluating if possible hypotheses for the object in current frame are coherent with respect to the expected attribute values of the dynamics model, and eliminating incoherent hypotheses.



(a) (b) (c)

Figure 3.9: Example of a badly segmented object, and how this issue can be solved with the proposed tracking approach. Figure (a) shows the original frame. Figure (b) shows the segmentation result, where resulting blobs are depicted in orange and moving pixels in white. Yellow lines represent the pre-defined static objects in the scene. Notice that the legs of the tracked person are badly segmented, with some parts of the shoes detected as movement separately from the body. In Figure (c), the solution found by the proposed tracking algorithm for this problem is depicted. The 3D parallelepiped representation is coloured in green, while object trajectory is represented by a sequence of points in red, connected by red segments. Last point in trajectory represents the expected position for next frame, which is connected by a green segment.

Hence, the proposed multi-object tracking approach presents the following main characteristics:

1. A new multi-hypothesis algorithm for tracking multiple objects for real world applications.

2. A new dynamics model for object tracking which keeps redundant tracking of 2D and 3D object information, in order to increase robustness.

3. New methods for best object hypothesis generation in order to ensure a high processing time performance for tracking.

For further details about this new multi-object tracking approach, see Chapter 5.

Next Section 3.5 introduces the proposed event learning algorithm. It uses as input the set of mobile objects obtained by the introduced tracking approach, which are represented by a set attributes with reliability measures associated to the temporal coherence of these attributes.

3.5 Incremental Event Recognition and Learning

The proper treatment of the information by the previous tasks of the proposed video understanding framework, allows the final event learning task to obtain as input a more detailed and refined description of the mobile objects evolving in the scene, as also to identify the most valuable information contained in these object descriptions by using the reliability measures associated to the mobile object attributes.

This way, an event learning method based on models of incremental concept formation ([Gennari et al. 1990], [Carbonell 1990]) is proposed. The models of incremental concept formation allow to incrementally build a concept hierarchy based on incomplete or uncertain data, by updating the hierarchical concept structure with the arrival of each new data instance. These models also allow the classification (i.e. recognition) of a new instance, based on the inferred concepts from previously processed data. In the context of the proposed learning method, a concept corresponds to a state, and data correspond to the visual attributes of mobile objects present in the video scene.

The input data of this method correspond to object visual attribute values together with a reliability measure for each attribute, obtained from the multi-object tracking approach introduced in Section 3.4. These reliability measures represent the temporal coherence of the tracked object attributes, and are used to perform a proper selection of the relevant information for the learning approach.

The new incremental learning algorithm proposes an extension of the models of incremental concept formation, by expanding the representation of concepts to the first-order temporal relations (i.e. Markov hypothesis) between these concepts. Thus, in the context of the proposed learning approach, concepts (represented as nodes in the hierarchy) become the states induced by the tracked objects present in the scene, while the first-order temporal relations, representing the state transitions, become the learnt events. Therefore, the learning approach is able to incrementally generate a hierarchical representation

of the states and events occurring in the scene, as depicted in Figure 3.10. Information about the frequency of occurrence of these states and events is also calculated, which allows to determine if the current state and event of an object is normal or abnormal in terms of frequency. The utilised hierarchical representation presents concepts describing more general states in the top of the hierarchy, while the sibling state concepts in the hierarchy represent specifications of their parent.

Currently in the video understanding literature, several studies on event learning can be

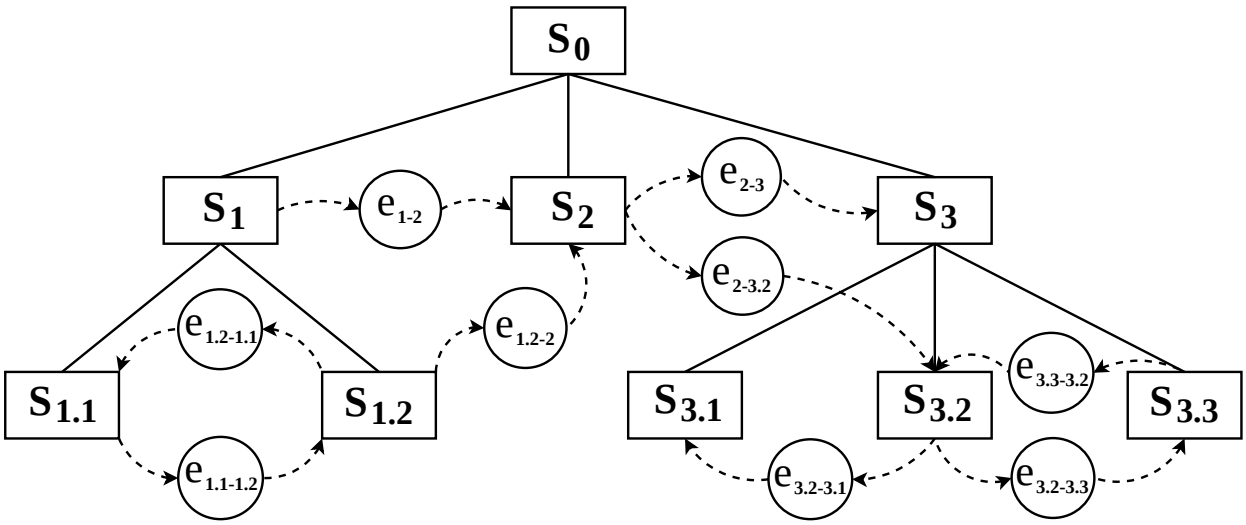


Figure 3.10: Example of a hierarchical event structure resulting from the proposed event learning approach. Rectangles represent states s, while circles represent events e. An event represents the unidirectional transition between two states.

found, which has been mainly performed for trajectories, and by applying off-line learning methods ([Fernyhough et al. 2000], [Owens and Hunter 2000], [Remagnino and Jones 2001], [Hu et al. 2004b], [Hu et al. 2006], [Jiang et al. 2007], [Gaffney and Smyth 1999] , [Reulke et al. 2008]). Very little attention has been given to incremental event learning in video ([Piciarelli and Foresti 2006]), which should be the natural step further for unexpected event recognition, or anomalous behaviour detection. Only few solutions have been proposed in the literature for bridging the gap between low-level video processing tasks (as segmentation and tracking), and high-level composite event analysis.

One of the objectives of this learning approach aims precisely at bridging this gap, by proposing a generic way of learning the frequency of events occurring in the scene, from the information obtained from low-level video processing tasks. These events can serve as building blocks for high-level event analysis.

For guiding the learning process, it is necessary to pre-define the *learning contexts*. A learning context corresponds to a description of the scope of the events of interest for the

user. It is defined as a set of object attributes, where these attributes can be numerical or symbolic. For the numerical attributes, it is necessary to associate a discrimination value, which represents the granularity of interest for this attribute. For example, in a trajectory learning context, we can be interested in learning the events associated to the 3D position (x, y) and velocity (V_x, V_y) of vehicles arriving to a parking lot.

As the attributes defined in the learning context are numerical, normalisation values have to be associated to these attributes, for corresponding to a meaningful variation of the attributes. Following the previous example, as a parking lot is a large open area, and vehicles velocity can be high, appropriate normalisation values can be *2 meters* for position attributes, and *10 km/h* for velocity attributes.

Several learning contexts can be simultaneously processed by the proposed approach, generating for each of them a different resulting hierarchy of states and events. Then, for each learning context, the event learning method extracts the appropriate available information according to the currently tracked objects in the scene. Then, state instances are created for each tracked object. These instances are classified through the hierarchy of states and the information of the instance is used to update the state hierarchy. Each state concept in the hierarchy is described by its frequency of occurrence, and by descriptions of the attribute values it represents.

Each tracked object can participate to more than one learning process at the same time, if this object is allowed according to the associated learning context. The state and event hierarchies are learnt combining the information provided by all the allowed mobile objects being tracked.

For the symbolic attributes of a state, all their possible values are listed and a frequency of occurrence value is associated, according to the number of instances which are considered for the attribute value. Numerical attributes are represented by the mean and standard deviation of the attribute values for the collected instances in the state concept.

Then, when an instance is classified, the associated state concept description is updated with the attribute information of the instance, considering the reliability measures associated to the attributes for weighting the contribution of this new information to the model of the attribute.

The learning algorithm keeps track of the current state of each mobile object. When an object changes of state, the event information is updated or created if it is the first occurrence of this event. Each event concept contains mean and variance information about the time of permanence of the mobile object in the previous state. This information can be very useful to understand the behaviour of objects evolving in the scene.

Hence, the result of the learning process corresponds to a learnt hierarchy of states and events for each pre-defined learning context, and the currently recognised state and event

for each object evolving in the scene. As the utilised event learning approach is incremental, the process of learning and recognition occurs simultaneously.

This way, the proposed incremental event learning approach presents the following main characteristics:

1. A new incremental frequent event learning approach, which learns states as a hierarchy of concepts, and also learns the frequency of occurrence of the events associated to these states.

2. The consideration of reliability measures associated to obtained data in the previous object tracking process in order to guide the learning process on the most reliable information.

3. A multiple contextual definition of the interesting attributes to be considered in the learning process.

For further details about this new incremental event learning approach, see Chapter 6.

Next Section 3.6 describes the possible interactions between the user and the proposed video understanding framework.

3.6 Framework Configuration and User Interaction

The proposed framework offers to the user different possibilities to build specific applications, as proposed by [Brémond and Thonnat 1998b]. This way, the video understanding framework allows the flexibility necessary for coping with a wide variety of objects and scenarios with an acceptable precision and time performance.

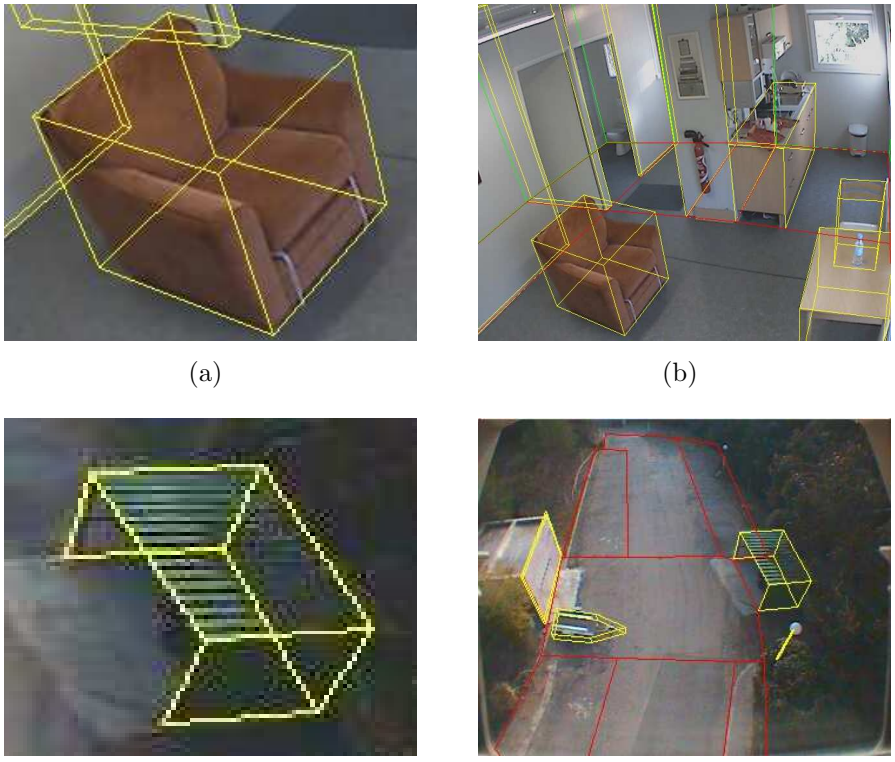
It is critical for the utilisation of 3D information in the video understanding framework, to define at least one model of the expected mobile objects in the scene. These models allow the framework to extract the 3D information of mobile objects detected and tracked in the scene. A model of expected object is defined as a probabilistic parallelepiped, described by each of its 3D dimensions (width, length and height). Each of these attributes is described by a Gaussian function accounting for the probability of occurrence of an attribute value for a given object model. As any Gaussian distribution, they are defined by the mean expected value for the attribute μ, and the expected standard deviation σ values for the attribute.

Also, minimal and maximal values for each attribute model must be provided in order to guide the search of the attribute values in a valid interval. Optionally, a velocity model can be defined in the same way as dimensional attributes in order to help the tracking task in the search of possible tracks for a mobile object which are coherent with its type.

The parallelepiped model allows the definition of a wide variety of objects in a very simple way. The description is independent to the object orientation in the scene and relative position with respect to the camera. Even if these models are not perfectly suited for objects which change their posture (e.g. a person), the model is able to handle this type of objects in an appropriate way. The definition of an object with different postures is achieved by defining a parallelepiped model for each posture of interest, plus a general parallelepiped model representing all the possible dimensional limits of this object class.

The user can also define the zones of interest to be analysed in the video scene. These zones are defined as polygons in the ground plane of the 3D referential of the scene. These zones are used by the framework to discriminate the zones where the moving object will be considered for analysis. Examples of pre-defined zones of interest can be found in Figures 3.11(b) and 3.11(d).

The user can define the static objects and walls present in the scene. Two representations



(a) (b)

(c) (d)

Figure 3.11: Example of elements pre-defined in a video scene. Figure (a) shows a couch defined as a context object in an apartment scene. The apartment scene is presented in Figure (b). Figure (c) shows a motor-park defined as a context object in an open parking lot scene. The parking lot scene is presented in Figure (d). Context objects are represented with yellow lines, context walls with green lines, while zones of interest are represented with red lines.

are allowed for modelling two different types of static objects:

- If the static object to be represented has a visible internal zone where a mobile object could possibly arrive, the static object is defined by a set of walls, with a given height. This representation allows the definition of objects with open spaces inside them, as also the definition of objects without roof. For example, Figure 3.11(c) shows the definition of a motor-park as a static object of this type. Note that there is a missing wall for the entrance of vehicles.
- If the static object has no gaps inside, or the user is not interested in the mobile object interactions inside the object, the static object can be defined as a polygon in the ground plane of the 3D scene referential, plus the height of the static object.

These static elements are used by the video understanding framework to cope with the problem of static occlusion, where a mobile object is partially occluded by a static object present in the scene.

In order to guide the learning process through the extraction of the interesting events according to the application, the user can define several **learning contexts**. As described in Section 3.5, a learning context corresponds to a description of the scope of the events of interest for the user. It is defined as a set of the possible object types involved in the context, and a set of object attributes, where these attributes are numerical or symbolic. For the numerical attributes, it is necessary to associate a normalisation value representing meaningful attribute variation. For the symbolic attributes, it is necessary to list the values of interest for this attribute. Several learning contexts can be simultaneously processed by the proposed approach, generating for each of them a different resulting hierarchy of events.

Learning Context *Trajectory* {
Involved Objects: Any
Attributes:
Numerical x : 2 [m]
Numerical y : 2 [m]
Numerical Vx : 10 [km/h]
Numerical Vy : 10 [km/h]
End

Figure 3.12: Definition of a trajectory learning context in a parking lot environment.

As defined in Figure 3.12 for a trajectory learning context, the user can be interested in learning the events associated to the 3D object position (x, y) in the ground plane of the 3D referential of the scene, together with the (V_x, V_y) for any type of object in a parking lot environment (e.g. persons and vehicles). As the attributes defined in the learning

context are numerical, normalisation values have to be associated to these attributes. As a parking lot is a large open area, appropriate normalisation values can be *2 meters* for position attributes, and *10 km/h* for velocity attributes.

In other case, as defined in Figure 3.13 for a position-posture context, the user can

Learning Context *Position_Posture* {
Involved Objects: Person
Attributes:
Numerical x : 50 [cm]
Numerical y : 50 [cm]
Symbolic Posture : { Standing,
Crouching,
Sitting,
Lying }
End

Figure 3.13: Definition of a Position-Posture learning context for persons in an office environment.

be interested in learning the events associated to the 3D object position (x, y) in the ground plane of the 3D referential of the scene, together with the human posture in an office environment. As an office is a small closed area, appropriate normalisation values for position attributes can be *50 centimetres*.

All these possibilities of customisation by the user, give a high flexibility to the proposed video understanding framework in order to cope with a wide variety of applications and typical issues present in the video understanding domain.

3.7 Discussion

As seen in this chapter, the process of learning events from object attributes can be an extremely hard task. In order to treat the general event learning problem, lots of issues have to be solved. Different levels in the event learning process impose different issues to be considered for obtaining a proper description of the events occurring in the scene, even in presence of noisy data.

The video understanding framework proposes solutions to several common issues in the video understanding domain. At a global scope, the approach proposes a unified way of measuring and controlling robustness of data. This solution corresponds to the utilisation of reliability measures associated to the data obtained at different levels of the event learning process.

At a specific scope, the approach proposes new solutions for common issues in

classification, tracking and event learning. Also, at each task of the video understanding framework, special attention has been given on processing solutions able to achieve a performance adequate for real world applications.

- The 3D classifier includes inexpensive methods for discarding the analysis of objects types impossible to associate to the processed visual evidence. Also, the classifier includes optimisation mechanisms for quickly finding the most likely instances associated to each expected object model.

- The tracking algorithm also considers the optimisation of processing time by immediately generating the most likely hypotheses, instead of generating and pruning hypotheses. Also, every function of the proposed dynamics model is designed as an incremental function, allowing to update each function with the new arriving attribute values, instead of recalculating the functions. More details on processing time considerations for tracking can be found in Chapter 5.

- Finally, the learning task considers the processing time performance by nature, as it is conceived as an incremental learning approach, avoiding the need of extensive calculation when incorporating new information.

The proposed video understanding framework has been conceived to be utilised in diverse application domains. The framework can be considered as a generic framework for event learning in several aspects:

- The possibility of inexpensively defining several 3D object models. The proposed model allows a simple way of defining any object expected to be present in the scene, even if the object can change its posture.

- The flexibility in the definition of contextual scene information. The possibility of defining static object information, as the zones of interest of the 3D scene, gives the user the possibility to better guide the video understanding framework to the elements of the scene interesting for the user.

- The possibility of defining learning contexts, gives to the user a large number of possibilities of event analysis. Moreover, new attributes, such as interaction between people and equipment, can be derived to give even more flexibility to the video understanding framework.

Briefing, the two main global contributions of the video understanding framework with respect to the video event analysis domain are:

1. A new incremental video understanding approach able to learn generic events from a video scene. This approach proposes an automatic bridge between the low-level data obtained from objects evolving in the scene and higher level information which considers the temporal aspect. Incremental learning of events can be useful for rare event recognition and to serve as input for higher level event analysis. Current related work in video analysis has paid little attention on solving the problem

of generic frequent event learning, focusing mainly on specifically learning events associated to object trajectories. Also, event learning state of the art has currently few contributions in incremental event learning.

2. A new general way for controlling the pertinence on utilisation of noisy video data. The video understanding framework proposes to associate reliability measures to the obtained information, in order to account for the quality, coherence, and reliability of this information. Reliability measures have been already used in the video understanding domain, but just for very specific tasks and features. The reliability measures are used in every level of the proposed video understanding framework:

   - In the object representation associating visual reliability measures for the estimated dimensions of the 3D parallelepiped model (i.e. length, width, and height). These measures allow to account the visibility of the obtained 3D data, associating a degree of confidence for these attributes.
   - In the multi-hypothesis tracking algorithm the reliability measures take several forms:
     - Temporal coherence measures of attribute estimations.
     - A visual quality measure of attribute estimations.
     - Global temporal coherence measure of a tracked object.
     - Global visual quality measure of a tracked object.
     - Global reliability measure for a tracking hypothesis.
   - In the incremental event learning algorithm the reliability measures obtained in previous stages of the video understanding framework are used to determine which is the most valuable information to be utilised in the learning process.

As previously detailed, the contributions of this Thesis are centred in object classification, tracking, event learning, and in the interaction between different tasks of the video understanding process. The motion segmentation task is not in the scope of this Thesis, even if a segmentation approach is utilised in the proposed video understanding framework. For the same reason, background updating, which is a very important subject related to the segmentation task, is not studied in this Thesis. Background updating can be very useful for dealing with problems as illumination changes, weak contrast, and dynamic background in the scene. For details on background updating algorithms refer to [Jain et al. 1977, Parker 1991, Stauffer and Grimson 2000, Durucan and Ebrahimi 2001, Ziliani and Cavallaro 2001, Rosin 2002, Snidaro and Foresti 2003].

Next Chapters describe in detail the proposed 3D classifier (including the proposed object model), object tracker, and event learning method. Then, the following Chapter 4 describes the 3D model for expected objects to be present in the scene, together with the methodology for obtaining these models from visual evidence in the scene.

Chapter 4

Reliable Object Classification

In order to obtain the associated 3D information to a blob, a new 3D classifier for monocular video sequences is proposed. This new method allows to classify objects modelled independently from the position relative to the camera and object orientation. For this purpose, a simple and generic 3D model has been proposed, which represents an object as a parallelepiped.

The proposed model is described by the parallelepiped dimensions (width, length and height) and orientation in the ground plane of the scene. Also, visual reliability measures of the three estimated dimensions are proposed, which represent a measure of their visibility. These measures have been proposed to estimate the object dimensional attributes in the tracking method, by better weighting the most visible attribute values. This classifier interacts with the proposed tracking algorithm on-demand, as depicted in Figure 4.1.

This chapter is organised as follows. First, in Section 4.1, the proposed parallelepiped model is formally presented, including its mathematical formulation and the visual reliability measures associated to the parallelepiped dimensions. Second, Section 4.2 describes the method for finding the most likely parallelepiped model instance associated to a blob, explaining how different video interpretation domain issues (as static occlusion and objects with changing posture) have been solved by the model. Third, in Section 4.3, the parallelepiped model is validated in both aspects, processing time performance and classification correctness, by performing a validation test over generated data. Finally, in Section 4.4, remarks of the classification issues are discussed, serving as an introduction to the next chapter.

4.1 The 3D Parallelepiped Object Model

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

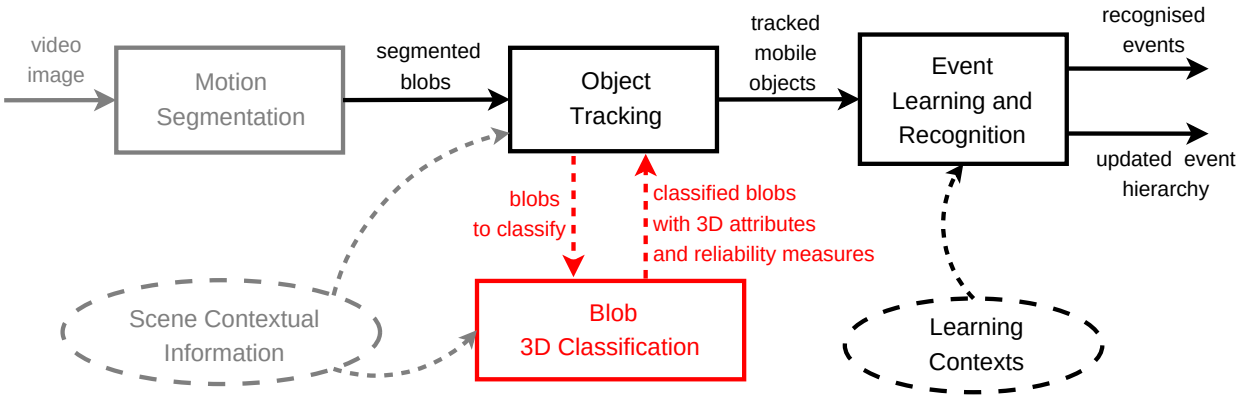


Figure 4.1: 3D Classifier as a component of the video understanding framework. Black elements correspond to the contributions of this thesis work. Gray elements correspond to elements used by the proposed framework, but not forming part of the contributions of this work. Red elements correspond to the elements analysed in this chapter, related with the 3D classifier.

on the information given by pre-defined 3D parallelepiped models of the expected objects in the scene.

An attribute model $\tilde{\mathbf{q}}$, for an attribute q can be defined as:

$$\tilde{\mathbf{q}} = (Pr_q(\mu_q, \sigma_q), q_{min}, q_{max}), \tag{4.1}$$

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 $Q_{\mathbf{C}}$ for an object class $\mathbf{C}$ can be defined as:

$$Q_{\mathbf{C}} = (\tilde{\mathbf{w}}, \tilde{\mathbf{l}}, \tilde{\mathbf{h}}), \tag{4.2}$$

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 with parameters (μ_w, σ_w), (μ_l, σ_l) , and (μ_h, σ_h), respectively.

The objective of the classification approach is to obtain a detected object model $S_{\mathbf{O}}$ for an object $\mathbf{O}$ detected in the scene, which better fits with an expected object class model $Q_{\mathbf{C}}$.

A 3D parallelepiped model $S_{\mathbf{O}}$ (see Figure 4.2) is described by:

$$S_{\mathbf{O}} = (\alpha, (w, R_w), (l, R_l), (h, R_h)), \tag{4.3}$$

where α represents the parallelepiped orientation angle (Figure 4.2(b)), 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 below.

The dimensions of the 3D model are calculated based on the 3D position of the vertexes of the parallelepiped in the world referential of the scene. Eight points $P_i^z(x_i, y_i) = (x_i, y_i, z)$ are defined, with $i \in \{0, 1, 2, 3\}$ and $z \in \{0, h\}$, as the 3D points that define the parallelepiped vertexes, with $P_i^{(0)}$ corresponding to the i-th base point and $P_i^{(h)}$ corresponding to the i-th vertex on height h, as shown in Figure 4.2(d). Also, P_i are defined (and respectively E_i), with $i \in \{0, 1, 2, 3\}$, as the 3D points (x_i, y_i) on the ground plane xy representing each vertical edge E_i of the parallelepiped, as depicted in Figure 4.2(b). The parallelepiped position (x_p, y_p) is defined as the central point of the rectangular base of the parallelepiped, and can be inferred from points P_i.

4.1.1 Mathematical Resolution

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

To find these values, a system of equations has to be solved. A first group of equations arise from the constraints imposed by the vertexes of the parallelepiped which are bounded by the 2D limits of the blob. For expressing these equations, four line segments in the 2D image referential are defined, as depicted in Figure 4.2(c):

SegLeft: Defined by points $[(X_{left}, Y_{top}); (X_{left}, Y_{bottom})]$.

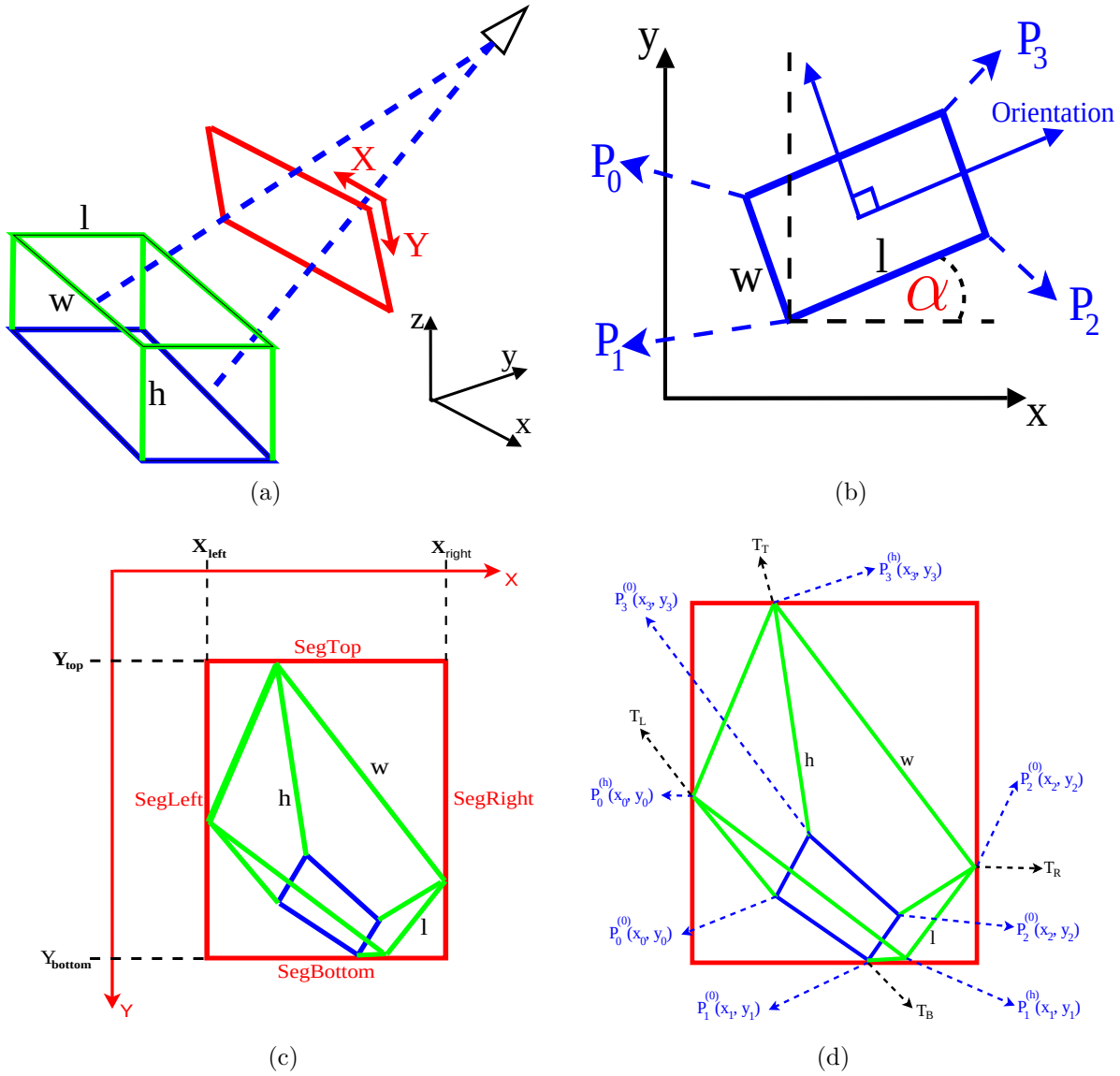


Figure 4.2: 3D parallelepiped model for detected objects. (a) 3D view of the scene. (b) Top view of the scene. (c) Point of view from the camera explaining image 2D referential variables. (d) Point of view from the camera explaining world 3D referential variables.

SegBottom: Defined by points $[(X_{left}, Y_{bottom}); (X_{right}, Y_{bottom})]$.
SegRight: Defined by points $[(X_{right}, Y_{top}); (X_{right}, Y_{bottom})]$.
SegTop: Defined by points $[(X_{left}, Y_{top}); (X_{right}, Y_{top})]$.

Then, points $T = \{T_L, T_B, T_R, T_T\} \in \{P_i^z | i \in \{0,1,2,3\}, z \in \{0,h\}\}$ are defined as the vertexes that comply with Equations (4.4). Indexes for vertexes T_j, with $j \in \{L, B, R, T\}$, stand for *left*, *bottom*, *right*, and *top*, respectively.

$$\begin{aligned} \textbf{ImageProjection}(T_L) &\in SegLeft, \\ \textbf{ImageProjection}(T_B) &\in SegBottom, \\ \textbf{ImageProjection}(T_R) &\in SegRight, \\ \textbf{ImageProjection}(T_T) &\in SegTop, \end{aligned} \tag{4.4}$$

where **ImageProjection**$(\cdot)$ is the function that projects a point from the 3D world referential of the scene onto the image plane. The problem is that the set of vertexes $T = \{T_L, T_B, T_R, T_T\}$ varies according to the orientation α of the parallelepiped and the relative position of the blob with respect to the camera.

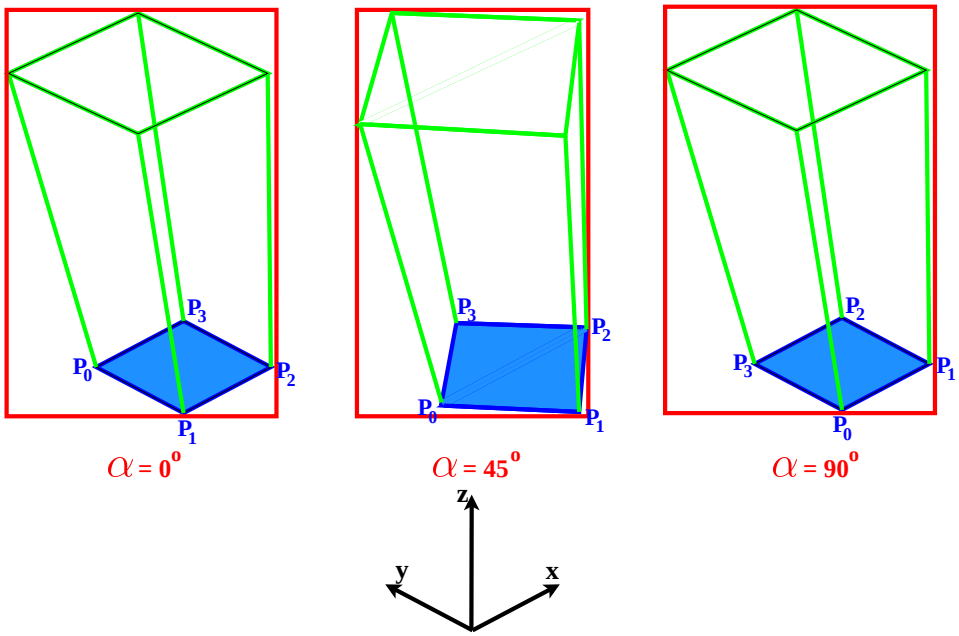


Figure 4.3: Effect of change of parallelepiped orientation in the vertexes bounded by the bounding box. Notice, for example, that from angle $\alpha = 0^o$ to $\alpha = 45^o$, the points T remain the same, while from angle $\alpha = 0^o$ to $\alpha = 90^o$, P_i points rotate, giving that $T_L \equiv P_0$ changes to $T_L \equiv P_3$, $T_B \equiv P_1$ changes to $T_B \equiv P_0$, and so on.

The change in orientation α just rotates the parallelepiped, so its effect in changing the vertexes just consists in rotating the indexes of points P_i bounded by the blob, as depicted in Figure 4.3. In this example, the set of points $T = \{T_L = P_0^{(h)}, T_B = P_1^{(0)}, T_R =$

$P_2^{(0)}, T_T = P_3^{(h)}\}$ changes to $T = \{T_L = P_3^{(h)}, T_B = P_0^{(0)}, T_R = P_1^{(0)}, T_T = P_2^{(h)}\}$, when parallelepiped orientation α changes from 0^o to 90^o, changing the indexes of the points P_i associated to each point in T, but the height of the vertex remains the same.

The relative position of the blob with respect to the camera poses a more delicate situation, as the visual perception of the parallelepiped varies according to the camera view. In a pin-hole camera model, the pinhole aperture of the camera, through which all projection lines pass, is assumed to be infinitely small, a point. In the literature this point in 3D space is referred to as camera focal point (x_f, y_f, z_f). Hence, the vertexes of the parallelepiped associated to each side of the blob depend on the relative position of the blob with respect to the projection in 2D image coordinates (X_f, Y_f) of the focal point projection on the ground of the 3D scene $(x_f, y_f, 0)$, as depicted in Figure 4.4.

This way, nine cases can be identified depending on the relative position of the blob

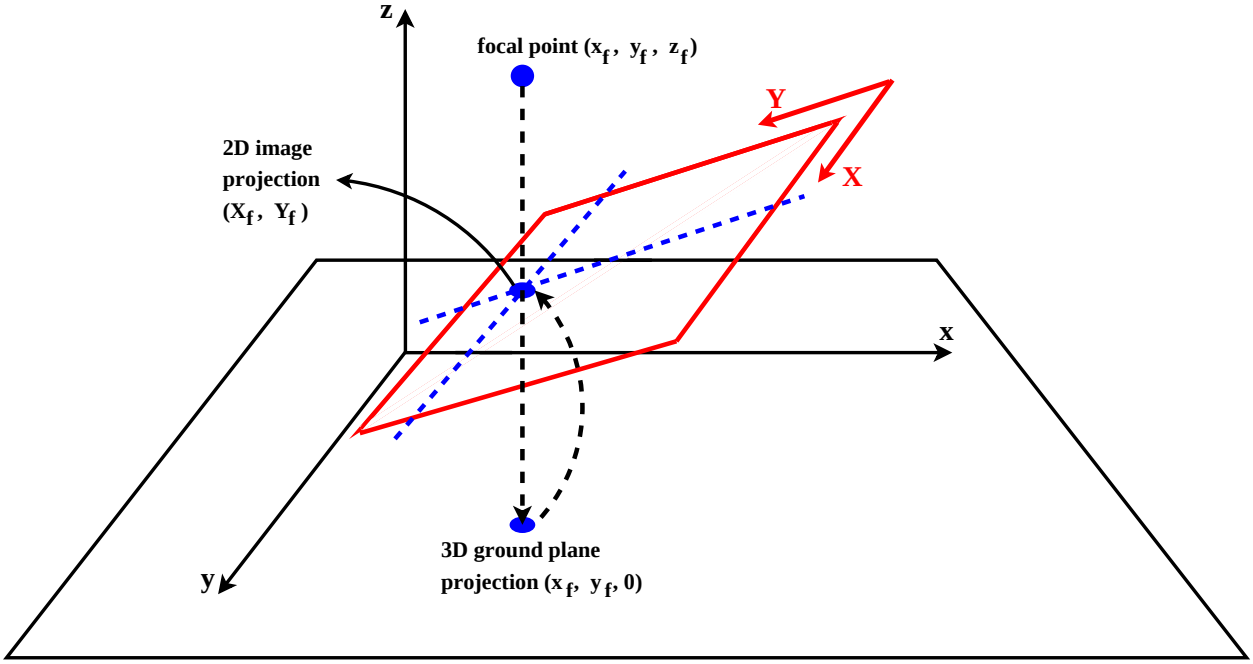


Figure 4.4: 2D projection (X_f, Y_f) of focal point (x_f, y_f, z_f), projected to the ground of the 3D scene $(x_f, y_f, 0)$. This 2D projection point is used to determine the parallelepiped case according to the camera view.

to the point (X_f, Y_f) as depicted in Figure 4.5. As can be observed, each of these cases determine the height of the vertexes of the set T. For example, for Case C0, T_L and T_T height is h, and T_B and T_R vertexes height is 0, while for Case C4, the height of every vertex in the set T is h.

Hence, considering a blob b with 2D limits $(X_{left}, Y_{bottom}, X_{right}, Y_{top})$ the height of the vertexes in set T is determined following the rules of Equation (4.5):

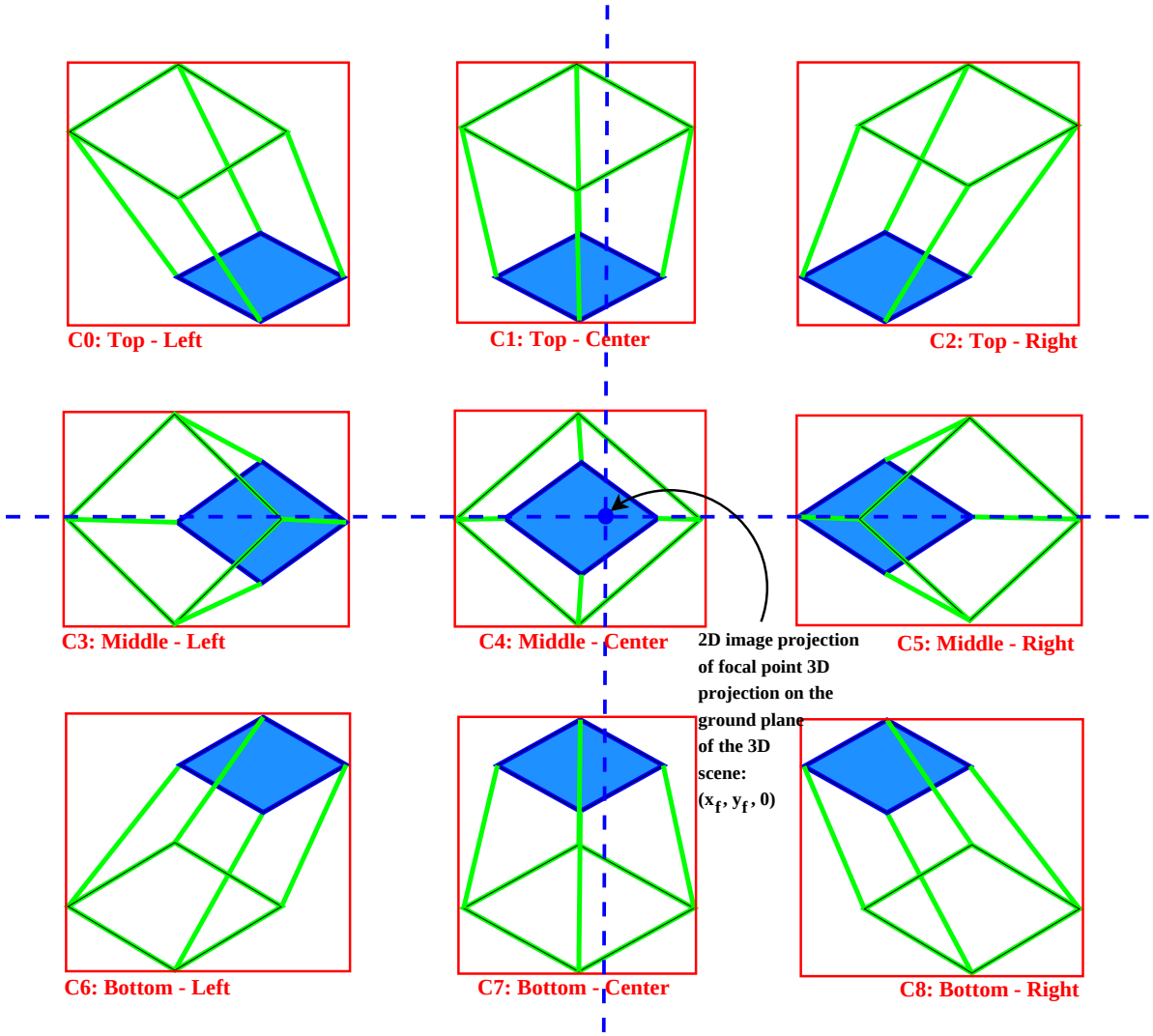


Figure 4.5: Different parallelepiped cases determined with the relative 2D position of the blob with respect to the 2D projection of the focal point (X_f, Y_f).

$$
\begin{aligned}
\mathbf{T_L} &= \begin{cases} P_l^{(h)} & if \quad X_{left} <= X_f \\ P_l^{(0)} & else \end{cases} \\
\mathbf{T_B} &= \begin{cases} P_b^{(h)} & if \quad Y_{bottom} >= Y_f \\ P_b^{(0)} & else \end{cases} \\
\mathbf{T_R} &= \begin{cases} P_r^{(h)} & if \quad X_{right} >= X_f \\ P_r^{(0)} & else \end{cases} \\
\mathbf{T_T} &= \begin{cases} P_t^{(h)} & if \quad Y_{top} <= Y_f \\ P_t^{(0)} & else \end{cases}
\end{aligned}
\tag{4.5}
$$

where l, b, r, and t correspond to the indexes of the parallelepiped vertexes bounded by the 2D limits (X_{left}, Y_{bottom}, X_{right}, and Y_{top}), respectively. Consider then, the function $In\_h(T_j)$, with $j \in \{L, B, R, T\}$ which returns 1 if the vertex is bounded by the blob b in parallelepiped height h, or 0 if the vertex is bounded by the blob b in the base of the parallelepiped at height 0. Consider also the pin-hole camera model Equation (4.6), with $\mathbf{M}$ corresponding to the calibrated perspective matrix:

$$\begin{pmatrix} X_k \\ Y_k \\ k \end{pmatrix} = M \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad with \quad M = \begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{pmatrix}, \tag{4.6}$$

where $X = X_k/k$, and $Y = Y_k/k$ correspond to image referential 2D coordinates. Then, using the pin-hole camera model Equation (4.6), and the four relations of Equation (4.4), four linear equations can be derived between each pair of variables $T_j = (x_j, y_j)$ of vertexes set T, with $j \in \{L, B, R, T\}$, as shown in Equation (4.7).

$$\begin{aligned}
&(p_{20} \times x_L + p_{21} \times y_L + p_{22} \times h \times In_h(T_L) + p_{23}) \times X_{left} \\
&\quad = p_{00} \times x_L + p_{01} \times y_L + p_{02} \times h \times In_h(T_L) + p_{03}, \\
&(p_{20} \times x_B + p_{21} \times y_B + p_{22} \times h \times In_h(T_B) + p_{23}) \times Y_{bottom} \\
&\quad = p_{10} \times x_B + p_{11} \times y_B + p_{12} \times h \times In_h(T_B) + p_{13}, \\
&(p_{20} \times x_R + p_{21} \times y_R + p_{22} \times h \times In_h(T_R) + p_{23}) \times X_{right} \\
&\quad = p_{00} \times x_R + p_{01} \times y_R + p_{02} \times h \times In_h(T_R) + p_{03}, \\
&(p_{20} \times x_T + p_{21} \times y_T + p_{22} \times h \times In_h(T_T) + p_{23}) \times Y_{top} \\
&\quad = p_{10} \times x_T + p_{11} \times y_T + p_{12} \times h \times In_h(T_T) + p_{13}.
\end{aligned} \tag{4.7}$$

These four equations are valid when each vertex P_i ($i \in \{1, 2, 3, 4\}$) associated to a variable T_j is bounded by only one blob limit. If this is not the case, we are in presence of a degenerate case, where a same vertex is bounded by two blob limits at the same time. For further details about the types of degenerate cases and their resolution, refer to Appendix A.

Other six equations can be derived from the fact that the parallelepiped base points P_i, with $i \in \{0, 1, 2, 3\}$, form a rectangle. Then, considering the parallelepiped orientation α, these equations are written in terms of the parallelepiped base points $P_i = (x_i, y_i)$, as shown in Equation (4.8).

$$\begin{aligned} x_2 - x_1 &= l \times cos(\alpha) \\ y_2 - y_1 &= l \times sin(\alpha) \\ x_3 - x_2 &= -w \times sin(\alpha) \\ y_3 - y_2 &= w \times cos(\alpha) \\ x_0 - x_3 &= -l \times cos(\alpha) \\ y_0 - y_3 &= -l \times sin(\alpha) \end{aligned} \tag{4.8}$$

These six[1] equations define the rectangular base of the parallelepiped, considering an orientation α and base dimensions w and l.

As there are 12 variables and 10 equations, there are two degrees of freedom for this problem. In fact, the problem posed this way, defines a complex non-linear system, as sinusoidal functions are involved, and the indexes $j \in \{L, B, R, T\}$ for the set of bounded vertexes T are determined by the orientation α. Then, the wisest decision is to consider 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 (4.9). 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 relations presented in Equations (4.7) and (4.8).

$$\begin{aligned} w &= M_w(\alpha; M, b) \times h + N_w(\alpha; M, b) \\ l &= M_l(\alpha; M, b) \times h + N_l(\alpha; M, b) \\ x_3 &= M_{x_3}(\alpha; M, b) \times h + N_{x_3}(\alpha; M, b) \end{aligned} \tag{4.9}$$

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

$$S_{\mathbf{O}} = f(\alpha, h, M, b) \tag{4.10}$$

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

Equation (4.10) 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.

4.1.2 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 objective is to find a measure that gives a minimal value (e.g. 0) when attribute is not visible, and a maximal value (e.g. 1) when the dimension is totally visible. The chosen function is $R_q(S_{\mathbf{O}}) \rightarrow [0, 1]$, 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 (4.11).

$$R_a = \min\left(\frac{dY_a \cdot Y_{occ}}{H} + \frac{dX_a \cdot X_{occ}}{W}, 1\right), \tag{4.11}$$

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.

In simple terms, this function accounts for the visibility of estimated parallelepiped dimensions in the image. The value of this function is between 0 and 1. 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. An exception occurs in the case C4 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.

The concept of visibility is not necessary for describing the reliability of the parallelepiped orientation α and parallelepiped position (x_p, y_p), because these attributes depend on dimensions w and l. Hence, no dimensional reliability measure associated to the visibility is proposed for these attributes. In Section 5.2.2, a reliability measure for attributes α and (x_p, y_p) is proposed as the mean between the visual reliability of w and l. These reliability measures are used in the object tracking task of the video understanding framework to weight the contribution of new attribute information.

For each class C of pre-defined models
  For all valid pairs (h, α)
    $S_O \leftarrow F(\alpha, h, M, b)$;
    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 4.6: Classification algorithm for optimising the parallelepiped model instance associated to a blob.

4.2 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 presented in section 4.1. 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 (4.12)), 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.

$$PM(S_{\mathbf{O}}, C) = \prod_{q \in \{w,l,h\}} Pr_q(q|\mu_q, \sigma_q) \tag{4.12}$$

Given a perspective matrix $\mathbf{M}$, object classification is performed for each blob b from the current frame as shown in Figure 4.6.

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.

4.2.1 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 video understanding framework, static objects can be modelled as part of the context of the 3D scene, as described in Chapter 3. Then, a static object is defined as a set of points delimiting the base of the object, together with the 3D height of the object.

The possibility of occlusion with the border of the image is easy to determine as it 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. Then, the direction and limit of blob bounds possible growth for the image referential directions *left*, *bottom*, *right*, and *top* are determined, according to the position of the possibly occluding elements. 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 direction of possible growth is to the left.

As stated before, the possibility of occlusion by the border of the image for a given blob is determined by the proximity of the blob to the image border. For determining the possibility of occlusion by a static object several tests are performed:

1. First, the 2D proximity to the static object 2D bounding box is analysed as a first filter for occlusion possibility.

2. If 2D proximity test is passed, the next step is to evaluate the blob proximity to the 2D projection of the static object in the image plane.

3. Finally, 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 [Georis et al. 2004].

When a possible occlusion exists, the maximal possible growth for the possibly occluded bounds of the blob is determined. First, in order to establish an initial limit for the possible growth of blob bounds caused by occlusion, the largest possible expected objects in the scene are considered at the blob position, and the 2D bounds of the blob enclosing these largest expected objects are taken into account if they exceed the blob initial bounds in the direction of possible occlusion. If all possible largest expected objects do not impose a larger bound to the blob, the hypothesis of possible occlusion is discarded.

Then, the obtained limits of growth for blob bounds are adjusted for static objects, by analysing the hidden faces of the object polygon which possibly occludes the blob. The growth of a blob bound is then limited by the 2D projection of the line defined by the hidden face at height 0 on the ground of the 3D scene, as the object enclosed by the blob can not pass through the object.

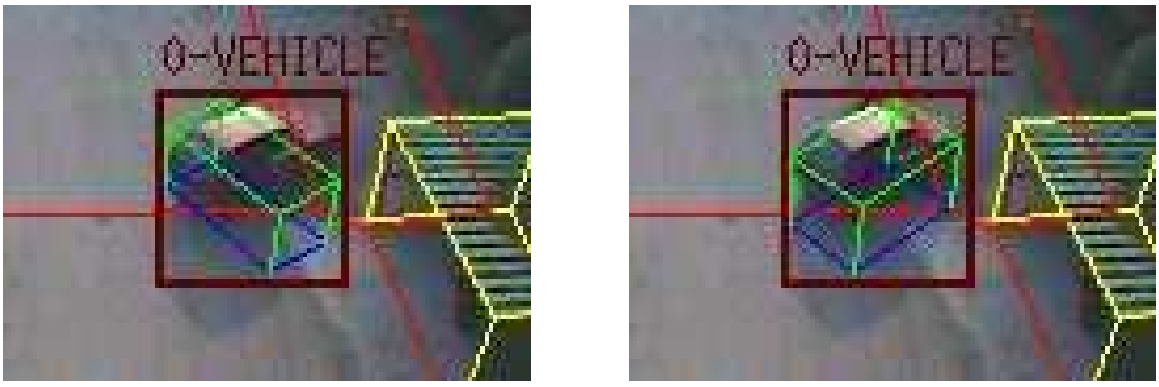
Then, for each object class, the calculation of occluded parallelepipeds is performed by taking several starting points for extended blob bounds in the occlusion direction which represent the most likely configurations for a given expected object class. Configurations which pass the allowed limit of growth are immediately discarded and the remaining blob

bound configurations are optimised locally with respect to the probability measure PM, defined in Equation (4.12), using the same algorithm presented in Figure 4.6. 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.

4.2.2 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 4.7, where two solutions are very likely geometrically given the model, but the most likely from the expected model has the wrong orientation.

A good way for discriminating between ambiguous situations is to return to moving



(a) (b)

Figure 4.7: Geometrically ambiguous solutions for the problem of associating a parallelepiped to a blob. Figure (a), shows an ambiguity between vehicle model instances, where the one with incorrect orientation has been chosen. In Figure (b), the correct solution to the problem.

pixel level. As the problem of finding the parallelepiped associated to a blob is partially solved by optimising the fitness to the expected object models, a simple solution is to store the most likely found parallelepiped configurations and to select the instance which better fits 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 (4.12)) will be able to discriminate which is the most likely object type, it exists also the possibility that overlapping object models give good PM values for different classes. 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 in order to represent the different hypotheses for a same blob. 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.

4.2.3 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 a object changing postures by the later tracking task.

4.2.4 Implementing For High Processing Time Performance

In order to obtain a high processing time performance, different mechanisms are utilised. These mechanisms search to reduce the computational load of the approach by preprocessing the information at different levels of the classification process:

1. When a blob is received as input by the classification algorithm, the size of the blob is utilised to discard object classes which can not be represented by the blob. For each object model of the expected objects in the scene, its information regarding the minimal and maximal attribute values for the model is utilised to generate the maximum and minimum size parallelepipeds for the model. These parallelepipeds are tested at different angles to generate the blobs bounding them.

If all the generated blobs are bigger or smaller than the analysed blob, the model is immediately discarded, as no solution can be later found by the classification algorithm.

2. As described in Equation (4.10), the solution of the parallelepiped association problem depends on the perspective matrix M, the blob b, the orientation angle α, and the parallelepiped height h. In order to optimise the calculation of the values of parallelepiped width w, length l, and base points P_i, with $i \in \{0, 1, 2, 3\}$, different values can be preprocessed according to the available information:

   (a) Before the execution of the classification task, the mathematical expressions only depending on the matrix M can be calculated.

   (b) Then, when the blob b to be analysed is available, the mathematical expressions only depending on the blob 2D bounds $(X_{left}, Y_{bottom}, X_{right}, Y_{top})$ are calculated.

   (c) Next, when a value for the orientation α has been fixed among the interval of analysis for the orientation, the mathematical expressions now only depending on α are calculated, arriving to the linear expressions of the Equation (4.9).

   (d) Finally, values for w, l, and P_i, with $i \in \{0, 1, 2, 3\}$, can be inexpensively calculated by evaluating the linear expressions for the different valid values for h.

   This way, a cascade of constant values calculation is performed in order to avoid extensive recalculation at each level of the classification algorithm.

3. Another mechanism for improving the processing time performance of the approach is the utilisation of the analysed object model information to narrow the interval of valid values for h. For this purpose, the minimal q_{min} and maximal q_{max} values for the attribute $q \in \{w, l, h\}$ are utilised, in the following way:

   (a) First, the limits of w interval $[w_{min}, w_{max}]$ are utilised with Equation (4.9) to obtain an interval of valid attribute h values defined as $h_w = [(w_{min} - N_w)/M_w, (w_{max} - N_w)/M_w]$.

   (b) Second, h_w interval is intersected with the model limits for h interval $h_h = [h_{min}, h_{max}]$, obtaining the interval $h_\wedge$.

   (c) Third, the limits of l interval $[l_{min}, l_{max}]$ are utilised with Equation (4.9) to obtain an interval of valid attribute h values defined as $h_l = [(l_{min} - N_l)/M_l, (l_{max} - N_l)/M_l]$.

   (d) Finally, interval $h_\wedge$ is intersected with the interval h_l, obtaining the final interval of valid values for h.

4. Also, as described in previous Section 4.2.1, the utilisation of starting points for searching the best solutions for a static occlusion situation, improves the processing time performance, by guiding the algorithm through the most likely parallelepiped configurations.

4.3 Testing Robustness and Processing Time Performance

In order to evaluate the processing time performance and robustness of the classification approach, a test has been performed on synthetic data. For this purpose, 27000 parallelepiped model instances have been generated, with different 3D dimensions, orientation, 3D position, and object type. They have been generated in two environments: the first one corresponds to a parking lot scene, referred as **Borel** sequence (**B**, in short), and the second corresponds to an apartment scene, referred as the **Gerhome** sequence (**G**, in short).

The test consists in utilising each of the synthetic parallelepipeds to obtain the blob bounding the parallelepiped. This blob is then utilised as input of the proposed classification approach and an associated parallelepiped is obtained. Finally, both parallelepiped, the synthetic and the obtained one, are compared and error measures are calculated. This way, for performing this test, the synthetic parallelepipeds were determined in the following way:

- Three expected object models are utilised: two models correspond to the postures of a person (**Person-Standing** (**P-S**, in short) and **Person-Crouching** (**P-C**)), and one model represents a **Vehicle** (**V**, in short). Models **P-S** and **P-C** are calculated in both sequences **B** and **G**, while model **V** is just calculated in sequence **B**.

- For each of these three models, **P-S**, **P-C**, and **V**, combinations of three values for each dimensional attribute w, l, and h, are considered. The three values correspond to the set $\{\max(\mu_q - \sigma_q; q_{min}), \mu_q, \min(\mu_q + \sigma_q; q_{max})\}$, with $q \in \{w, l, h\}$, chosen to represent situations where the likelihood of a numerical attribute value, with respect to the attribute model value, is not high. Taking all possible combinations among the three-values sets for the dimensional attributes, 27 combinations are considered. The set of three values $\{q_1, q_2, q_3\}$, with $q \in w, l, h$, considered for each object model are summarised in Table 4.1.

Model	w_{min}	w_{max}	$\sigma_{\mathbf{w}}$	w_1	$w_2 = \mu_w$	w_3	l_{min}	l_{max}	$\sigma_{\mathbf{l}}$	l_1	$l_2 = \mu_l$	l_3	h_{min}	h_{max}	$\sigma_{\mathbf{h}}$	h_1	$h_2 = \mu_h$	h_3
P-S	30	100	20	30	40	60	20	70	30	20	25	55	120	220	60	120	170	220
P-C	40	100	20	40	50	70	40	80	30	40	60	80	90	140	60	90	110	140
V	125	190	50	125	156	190	200	480	100	271	371	471	100	160	35	100	134	160

Table 4.1: Values considered for each object dimension $q \in \{w, l, h\}$, as $\{q_1, q_2, q_3\} = \{\max(\mu_q - \sigma_q; q_{min}), \mu_q, \min(\mu_q + \sigma_q; q_{max})\}$.

- Then, for each of the 27 combinations of object model dimensions, four values for orientation α are considered: $\{0.0, \pi/6, \pi/3, \pi/2\}$.

- Finally, 50 different parallelepiped 3D positions are considered for each value of α. Among these parallelepiped positions, situations representing static border occlusion

are considered. Just border occlusion is considered because, as described in previous Section 4.2.1, the treatment of the static occlusion situation is independent from the type of occlusion, and because the border occlusion can be detected just with 2D information.

This way, 5400 parallelepipeds are processed for each of the five utilised Sequence-Type pairs, giving a total of 27000 analysed parallelepipeds. Examples of the processed parallelepipeds are depicted in Figure 4.8. Figures 4.8(a) and 4.8(b), shows the detected parallelepipeds from the sequence **B** for the vehicle class, while Figures 4.8(c), 4.8(d), 4.8(e), and 4.8(f) show the parallelepipeds from the sequence **G** for two postures of the person class. Figures 4.8(a), 4.8(c), and 4.8(e) correspond to occlusion situations, while Figures 4.8(b), 4.8(d), and 4.8(f) represent non-occlusion situations.

For robustness evaluation, three error measures have been calculated:

- **Mean Dimensional Error ϵ_d:** This measure corresponds to the mean value of the dimensional errors, as presented in Equation (4.13).

$$\epsilon_d = \frac{\epsilon_w + \epsilon_l + \epsilon_h}{3.0}, \text{ with: } \epsilon_w = |w_s - w_c|,\ \epsilon_l = |l_s - l_c|, \text{ and } \epsilon_h = |h_s - h_c|, \quad (4.13)$$

  where dimensions with s subscripts stand for the synthetic parallelepiped, while attributes with c subscripts stand for the calculated parallelepiped.

- **Mean Position Error ϵ_p:** This measure corresponds to the mean value of the euclidean distance between the 3D position of the synthetic and calculated parallelepipeds.

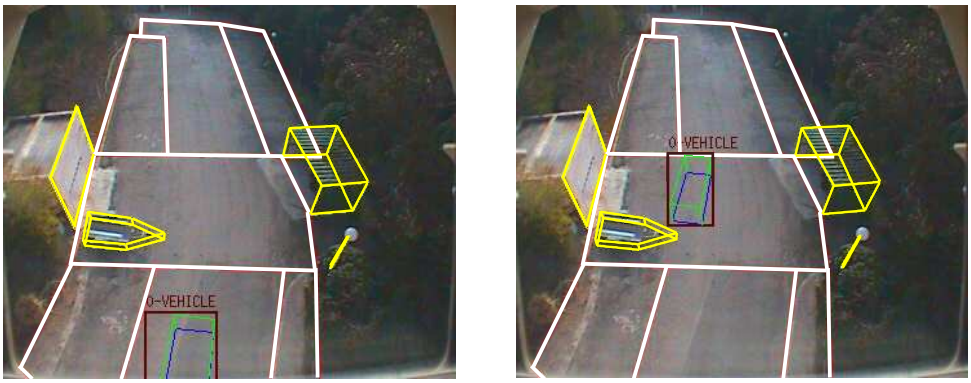
- **Mean Alpha Error ϵ_α:** Difference between the orientation angle α of the synthetic and calculated parallelepipeds.

For processing time performance evaluation the blob rate measure, representing the number of blobs that the classifier can process per second ([**blobs/sec**]), and the blob speed measure, representing the mean time spent in the classification of a blob ([**secs/blobs**]), have been calculated.

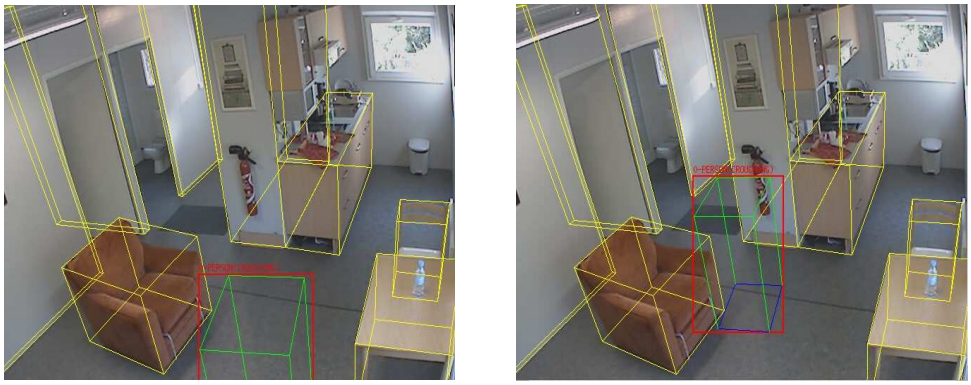
4.3.1 Results

The tests were performed on a computer with processor Intel Xeon CPU 3.00 GHz, with 2 Giga Bytes of memory. The obtained results in terms of the three error measures are summarised in Table 4.2. Note that the error measures are normally higher for static occlusion situations as a hidden part of the blob forces the model to fit its dimensional values to the model, approaching to mean dimension values.

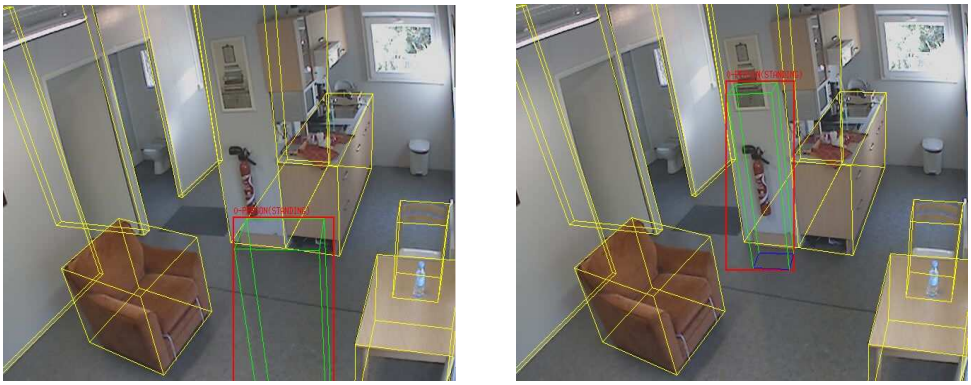
In general terms, mean error values show that the associated parallelepiped presents



(a) (b)



(c) (d)



(e) (f)

Figure 4.8: Examples of calculated parallelepiped for the test with synthetic data. Blob bounds are coloured according to the object type (red for person, and brown for vehicle). Parallelepiped base is in blue, while projections in height are in green. Static context objects are coloured in yellow and context zones are coloured in white.

Test	Non-Occluded Blobs			Occluded Blobs		
	$\epsilon_{\mathbf{d}}$ ($\sigma_{\epsilon_{\mathbf{d}}}$)[cm]	$\epsilon_{\mathbf{p}}$ ($\sigma_{\epsilon_{\mathbf{p}}}$)[cm]	ϵ_{α} ($\sigma_{\epsilon_{\alpha}}$)[deg]	$\epsilon_{\mathbf{d}}$ ($\sigma_{\epsilon_{\mathbf{d}}}$)[cm]	$\epsilon_{\mathbf{p}}$ ($\sigma_{\epsilon_{\mathbf{p}}}$)[cm]	ϵ_{α} ($\sigma_{\epsilon_{\alpha}}$)[deg]
B: P-S	9.25 (5.86)	20.33 (14.77)	38.17 (25.32)	17.37 (8.56)	15.53 (6.88)	45.11 (26.56)
B: P-C	9.43 (5.29)	19.18 (13.83)	41.44 (27.01)	16.08 (8.6)	16.27 (7.13)	41.92 (26.64)
B: V	22.49 (15.26)	22.75 (15.18)	19.7 (19.58)	34 (20.35)	61.36 (37.94)	31.97 (24.04)
G: P-C	9.87 (6.2)	7.34 (6.43)	45.97 (29.18)	14.64 (8.02)	28.79 (21.41)	46.19 (27.98)
G: P-S	8.81 (5.88)	6.92 (6.2)	37.88 (28.78)	13.84 (6.85)	40.13 (25.82)	43.7 (29.16)
Mean	**12.13** (**7.79**)	**16.16** (**11.81**)	**36.12** (**25.69**)	**18.74** (**10.25**)	**34.68** (**21.91**)	**42.02** (**27.16**)

Table 4.2: Three analysed errors for each analysed object type and sequence. The error ϵ_d corresponds to the mean error in parallelepiped dimensions estimations, the error ϵ_d corresponds to the 3D parallelepiped position error, and the error ϵ_α corresponds to the error in orientation of the parallelepiped. Results are separated in occluded and not occluded object situations. The standard deviation of each analysed error is displayed between parentheses.

low error when the blob is completely visible, and also the errors present a low variability. Nevertheless, the effect of partial occlusion can be noticed by the added error in attribute estimation. This increment in the error occurs because the classification algorithm always tries to fit the most likely parallelepiped according to the models of expected objects present in the scene. As an occlusion situation adds another degree of freedom allowing the growth of a 2D dimension, the algorithm is less geometrically constrained to find a solution nearer to the model mean, in despite of real situations where the instance is not near the mean values

Figures 4.9, 4.10, and 4.11 present graphically the mean and standard deviation in error measures for error measures ϵ_d, ϵ_p, and ϵ_α, respectively.

From Figures 4.9, and 4.10 the influence of the variability of an object model with respect to the increment of the dimensional and position error can be observed. As seen in Table 4.1, the vehicle model presents the higher variability in dimensions and Figures 4.9, and 4.10 show, at the same time, a higher error mean and standard deviation for both ϵ_d and ϵ_p errors can be noticed for the vehicle model. The posture models for a person are quite similar in variability and this similarity is also reflected in the graphics.

The similar behaviour between errors ϵ_d and ϵ_p was expected, as they are tightly related because ϵ_d measures the mean error for the 3D dimensions w, l, and h, while ϵ_p measures the error of the 3D position of the parallelepiped, which is calculated based on the dimensions w and l.

Figure 4.11 shows the behaviour of the error in the orientation angle α. The orientation error for person postures has maintained its behaviour for both sequences, showing the independence of α with respect to the proximity and position to the camera. The orientation error for the vehicle model is lower than the other errors, which could be due to the pixel analysis mechanism described in Section 4.2.2. As the vehicle model

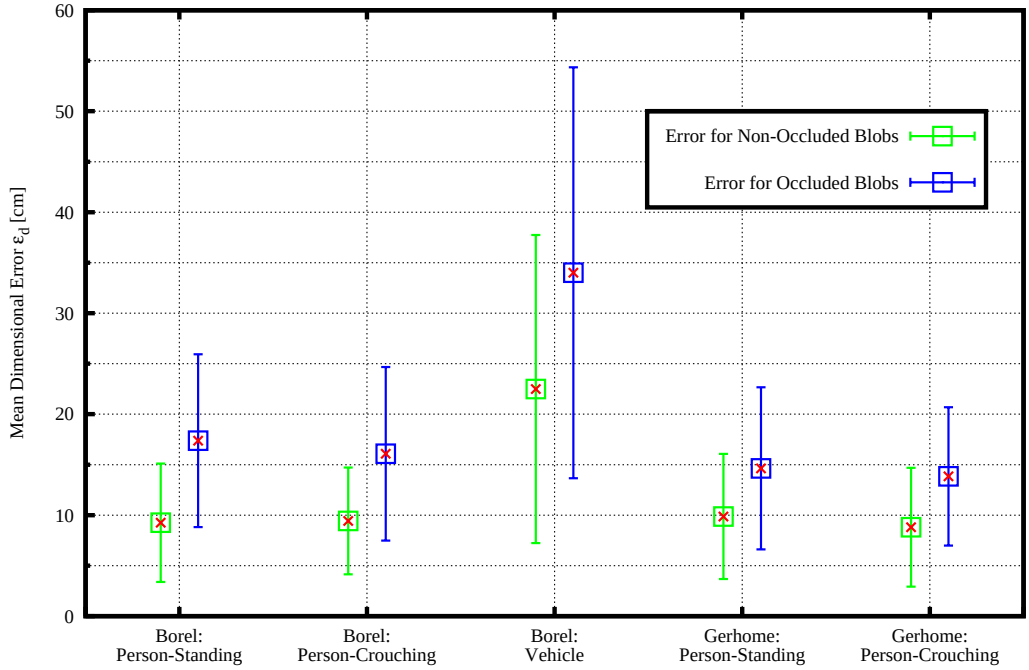


Figure 4.9: Dimensional Error ϵ_d for each analysed object type and sequence. The red cross shows the mean error, while the blue and green lines represent the standard deviation on the error. Green colour is utilised for non-occluded blob solutions, while blue colour is used for occlusion situations.

can be better appreciated from top than a person, the pixel analysis mechanism was able to better discriminate between the available solutions to find a more correct one to be associated to the blob.

Table 4.3 presents the results for the analysis of computer performance information for the proposed classification algorithm. Results show a good blob rate for non-occluded

Test	**Non-Occluded Blobs**			**Occluded Blobs**		
	Blob rate$[\frac{\text{blobs}}{\text{sec}}]$	**Blob speed**$[\frac{\text{sec}}{\text{blobs}}]$	**Frames**	**Blob rate**$[\frac{\text{blobs}}{\text{sec}}]$	**Blob speed**$[\frac{\text{sec}}{\text{blobs}}]$	**Frames**
B: P-S	106.24	0.009413	4603	9.99	0.100073	797
B: P-C	86.03	0.011624	4372	8.78	0.113873	1028
B: V	185.43	0.005393	4180	122.45	0.008167	1220
G: P-C	33.39	0.029953	3289	13.39	0.074672	2111
G: P-S	51.64	0.019366	3509	21.72	0.046033	1891
Total			**19953**			**7047**
Mean	**70.47**	**0.014191**		**15.61**	**0.064064**	

Table 4.3: Processing time performance for non-occluded and occluded blobs.

blobs, which indicates that the classification algorithm could perform with an adequate processing time performance for real world applications. In presence of occlusion, the

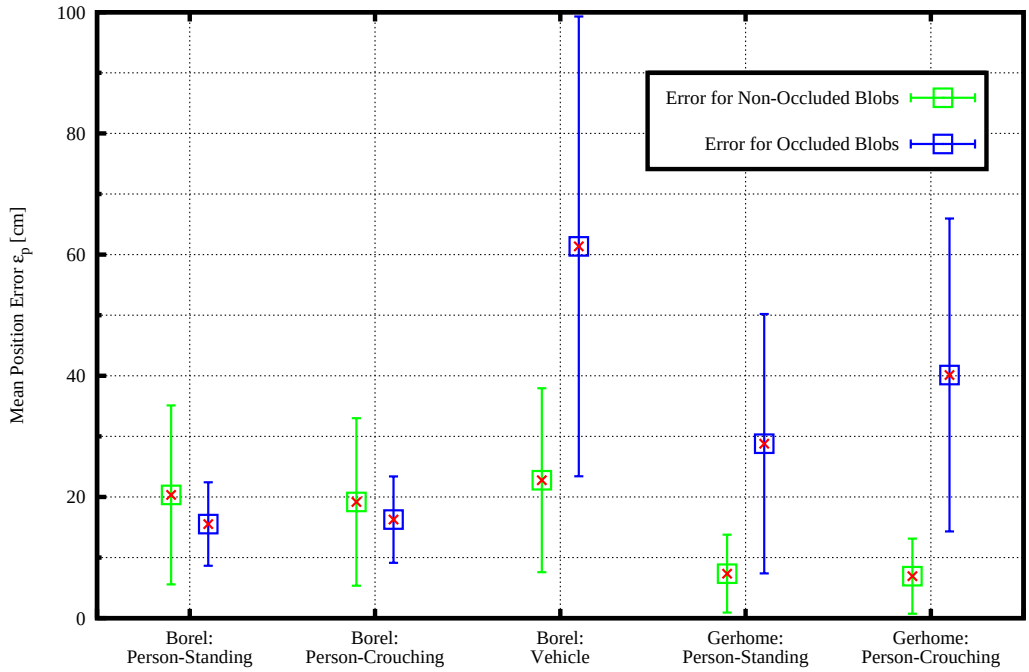


Figure 4.10: Position Error ϵ_p for each analysed object type and sequence. The red cross shows the mean error, while the blue and green lines represent the standard deviation on the error. Green colour is utilised for non-occluded blob solutions, while blue colour is used for occlusion situations.

classification algorithm still could have a high time performance for a scene of low complexity, but it is not possible to ensure a high processing time performance of the classification approach. This means that other mechanisms have to be envisaged to improve the processing time performance.

4.3.2 Experiment Conclusion

The experiment has shown that the obtained classification error is not excessive and its variability is also not high. The presented results on synthetic data show a robust behaviour of the classification approach. The computation of the orientation α and the 3D dimensions and position of the parallelepiped are independent from the relative position to the camera.

From the performance results, the method has shown its capability of obtaining an adequate processing time performance for situations of moderated complexity, but the results in performance for static occlusion situations indicate the necessity of other alternative or complementary ways for coping with the static occlusion problem.

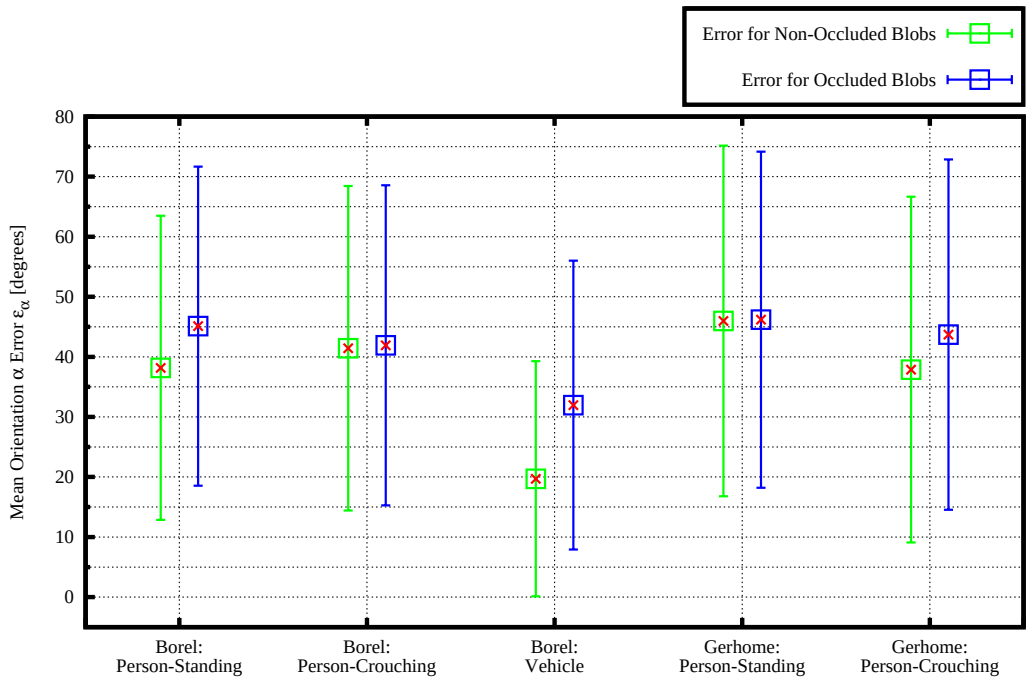


Figure 4.11: Orientation α Error ϵ_p for each analysed object type and sequence. The red cross shows the mean error, while the blue and green lines represent the standard deviation on the error. Green colour is utilised for non-occluded blob solutions, while blue colour is used for occlusion situations.

4.4 Discussion

The proposed classification method has shown interesting characteristics to be highlighted:

- Adequate processing time performance for scenarios of moderated complexity.
- Classification results independent from the camera view and orientation of the object, in case of synthetic data.
- Capability of coping with static occlusion situations. Nevertheless, the parallelepiped attribute estimation will be negatively affected by the degree of occlusion.
- Methods for disambiguation between several geometrically plausible alternatives.
- Representation capability for a large variety of objects, even those with different postures.

Visual reliability measures have been presented but not used by the classification method. These measures are intended to be used by the tracking approach to guide the temporal estimation of object features through the most reliable information.

The proposed object classification method presents the following limitations:

- The first one is related to the representation capability of the model. Even if this generic model is good for describing a large variety of objects, the result from the classification algorithm is a coarse description of the object. In order to address the interpretation of more complex situations, more detailed and class-specific object models could be utilised when needed. This problem, even if very interesting, is not in the scope of this thesis.

- A second limitation which belongs to the scope of this thesis, is the limited processing time performance. Even if the algorithm is quick enough to cope with several situations with a high processing time performance, it seems that the classification approach will have performance problems with scenarios of higher complexity. The main problem causing this limited performance is the lack of knowledge to guide the classification process to quickly find the optimal solution. In this sense, the tracking approach to be presented in next Section 5 can be of great help on indicating which are the parallelepiped attribute values more coherent with the currently tracked object attributes.

- A third limitation arising from the obtained results of the test presented in Section 4.3, is the imprecision in the estimation of the object orientation angle α. The results show that, for situations without occlusion, the mean orientation error for the person class is near 40°, while for the vehicle class the mean orientation error is near 20°. While 20° of mean error can be considered as acceptable, 45° seems very high. This error can be explained because of the camera view of the evaluated videos, where the model instances for the vehicle class are better discriminated than for the person class from this camera view, as more parallelepiped configurations for the person class at different orientation angles have a high value of the probability measure PM (Equation (4.12)).

- As a fourth limitation, the quality of the classification algorithm depends on the quality of the motion segmentation results. Therefore, more work still needs to be done in order to measure the impact of segmentation errors (e.g. shadows, reflections, poorly contrasted objects) on the classification results. In this work, the considered reliability concept measures mostly the visual ambiguity related to geometrical object attributes. Other reliability concepts can be taken into account in order to measure the occurrence of segmentation errors.

- Finally, a fourth limitation of geometrical nature can be identified. The resolution of the parallelepiped calculation problem presented in Section 4.1.1 has been formulated for focal point positions higher than the objects evolving in the scene. An object higher than the focal point height will lead to an erroneous calculation of the possible parallelepipeds associated to the object. This situation can not be considered as an error, but as case that has not been taken yet into account. The

solution of this limitation implies the resolution of a new system of equations for covering these situations.

As a summary, the proposed 3D shape object representation presents the following contributions:

1. A representation independent from the camera view and the orientation of the object with respect to the 3D referential of the scene.

2. A simple generic object representation model which allows users to easily define new mobile objects that could be present in the scene.

3. A model which instances can be obtained with an adequate processing time performance, with better precision than generic 2D primitive shape representations, providing 3D object features which are more interesting for event analysis tasks.

4. Reliability measures proposed to calculate the visibility of the obtained 3D object features, accounting for occlusion situations and camera view.

This classification approach is controlled by the new multi-object tracking approach proposed for the video understanding framework, which is described in next Chapter 5. This description includes the utilised data framework of hypotheses, the tracking algorithm and methods for hypothesis generation.

Chapter 5

Multi-target Tracking using Reliability Measures

In order to obtain coherent and reliable information about the objects evolving in a video scene, a new multi-object tracking approach has been proposed. This tracking method is a component of the video understanding framework proposed in this thesis work and presented in Chapter 3, as depicted in Figure 5.1. The object tracking approach takes as input the blobs which are the result from the previous image segmentation task to provide as output the most reliable and coherent list of tracked objects, described by a set of attributes with associated reliability measures. These reliability measures describe the visual quality of the analysed data and the temporal coherence of the obtained mobile object attribute values.

This tracking method maintains a list of likely configuration hypotheses for the mobile objects present in the scene. These hypotheses are validated or rejected according to the new visual evidence arriving to the tracker, checking the coherence and reliability of the estimated information for each tracked object. The most likely tracking hypotheses for a mobile are efficiently estimated in order to manage the complexity of the problem with a processing time performance adequate for real world applications. This approach combines blob 2D information, together with 3D information obtained from the 3D classifier, to generate a set of mobile object configuration hypotheses.

The hypotheses are grouped according to their visual proximity relations in the scene in order to separate the tracking procedure into different tracking sub-problems. A hypothesis is eliminated if it becomes unlikely in time, compared with other related hypotheses.

Each mobile object is represented as a set of statistics of features inferred from visual evidences of their presence in the scene. The tracking approach takes advantage of the 3D parallelepiped model presented in Chapter 4 to track the objects present in the scene using the most reliable available 2D and 3D information about the object. At the same time, the tracker guides the 3D classifier in two ways:

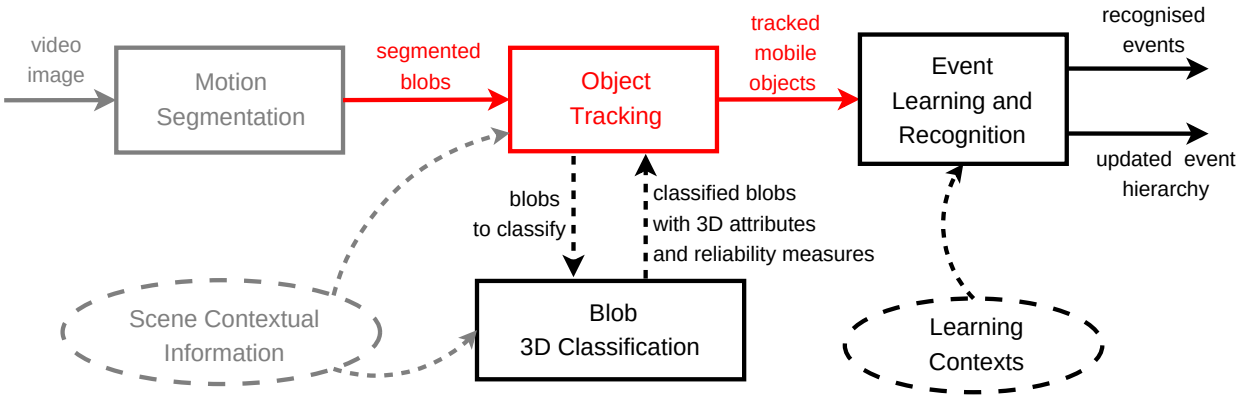


Figure 5.1: Proposed object tracking approach as a component of the video understanding framework. Black elements correspond to the contributions of this thesis work. Gray elements correspond to elements used by the proposed framework, but not forming part of the contributions of this work. Red elements correspond to the elements analysed in this chapter, related with the object tracker.

- By performing the 3D classification process when a minimal amount of visual support on the existence of the object has been collected. The 2D spatial and size coherence is evaluated in the first frames in which an object has been detected, in order to perform the 3D classification process over blobs associated to object hypotheses which are likely to be really occurring in the real situation. This way, the computer performance can be enhanced by processing less blobs to obtain the 3D information.

- By guiding the 3D classifier in the search of the optimal parallelepiped model. Using the estimated 3D dimensional and position attributes of the tracked objects, the tracking approach guides the classification process by defining the starting point for the object attributes and the allowed variability for each of these attributes.

A new object dynamics model is proposed, which utilises the visual reliability measures calculated for the parallelepiped model to weight the contribution of the new attribute information to the estimated attribute calculation, with respect to the reliability of this new information. This way, reliable information is enforced in the dynamics model, contributing to the robustness of the approach by handling noisy data. Also, a cooling function is utilised in order to diminish the contribution of old information, and highlight the contribution of the newest information. The functions utilised to update the dynamics model information are defined incrementally, in order to improve the calculation time.

The proposed tracking approach is able to cope with several issues common to multi-object tracking techniques. The problems of partial object segmentation or over-segmentation are solved by the proposed tracking approach by maintaining the temporal coherence of each tracked object, evaluating if the possible hypotheses for the objects in the current

frame are coherent with respect to the expected attribute values of the dynamics model, and then suppressing incoherent hypotheses. The static occlusion problem resolution proposed by the 3D classifier is reinforced by the tracking approach, guiding the classifier in the search of the 3D attributes and the real size of the 2D blob, based on the temporal coherence of the expected tracked object 2D and 3D attributes.

This chapter is organised as follows. First, the terminology utilised for describing the tracking approach is defined in Section 5.1. Second, in Section 5.2, the proposed representation of tracking information is presented, explaining its different levels of abstraction and how the dynamics model of the mobile object attributes is updated in time. Third, Section 5.3 presents the object tracking method explaining the general framework of the algorithm, and the processes involved in the tracking approach. These tracking processes are dedicated to the separation of the tracking problem in sub-problems, the generation and elimination of mobile object hypotheses, and the treatment of visual interpretation domain issues. Fourth, in Section 5.4, an illustration of the object tracking approach is presented. Finally, in Section 5.5, the most important aspects of the tracking approach are discussed.

5.1 Multi-object Tracking Terminology

In the context of the proposed object tracking approach, several concepts must be defined:

Definition 5.1 *A **mobile object** (or simply a mobile) is a potential physical object present in the scene, observed during a time interval. It is described by a set of attribute statistics calculated with the accumulation of the information provided by the visual evidences of the presence of the object in the scene.*

Definition 5.2 *A **mobile track** (or simply a track) is a potential mobile at frame k. This concept represents a possible new spatial configuration of a tracked object in the scene, inferred from the information of the mobile at frame $k-1$ and the new visual evidence at time t.*

Definition 5.3 *A **blob buffer** is the set of visual evidences of a mobile object for the k latest frames. The information at each frame k is represented by the 2D blob information, together with the 3D information provided by the classification task described in previous Section 4. The size of the blob buffer is a pre-defined value.*

Definition 5.4 *A **hypothesis** is a potential configuration of a set of mobiles. A hypothesis groups visually related mobile objects, and in this sense it corresponds to a possible interpretation of a partial world.*

Definition 5.5 *A **hypothesis set** is a set of mutually exclusive hypotheses, representing the set of different possible interpretations for a partial world. In this sense, a hypothesis set is a complete partial world.*

Definition 5.6 *A **hypothesis set list** is a list of hypothesis sets, representing the list of partial worlds. In this sense, a hypothesis set list represents the complete world.*

Definition 5.7 *An **involved blob set** is a set of blobs representing the valid correspondences between the visual evidence (blobs) at current frame k and a mobile object. This set can be also associated to a hypothesis as the union of the involved blob sets for the mobiles represented in the hypothesis, and to a hypothesis set as the union of the involved blob sets for these hypotheses.*

Next Section 5.2 presents the representation of the information utilised by the proposed object tracking approach.

5.2 Tracking Hypotheses Representation

The representation of the tracking information corresponds to a *hypothesis set list* as seen in Figure 5.2. Each *related hypothesis set* in the list is composed by a set of hypotheses which are exclusive between them. These hypotheses represent different alternatives for mobile 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.

This representation scheme is similar to the one utilised by the MHT approaches (Section

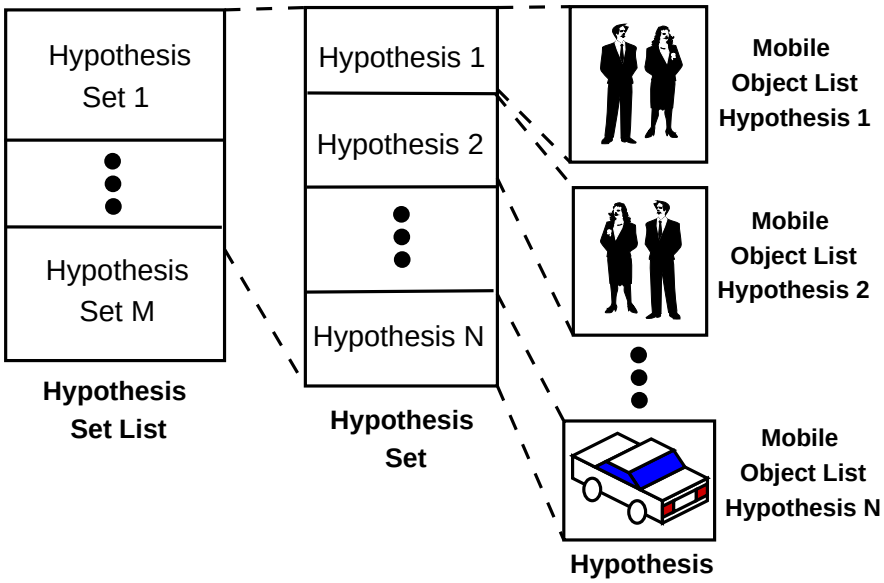


Figure 5.2: Representation scheme utilised by the new tracking approach. The representation consists in a list of hypothesis sets. Each hypothesis set consists of a set of hypotheses temporally or visually related. Each hypothesis corresponds to a set of mobile objects representing a possible object configuration in the scene.

2.2), as it explicitly considers the separation of the tracking problem into sub-problems

according to the spatial proximity of objects evolving in the scene, in order to diminish the complexity of the problem.

The difference of the utilised representation with existing work in tracking, lies fundamentally in the **dynamics model**. The most innovative aspect of the dynamics model is the explicit inclusion of reliability measures in the object attribute updating functions in order to control the influence of new incoming information according to its reliability, as described in Section 5.2.2.

5.2.1 Hypothesis Level

A hypothesis corresponds to a set of mobile objects, related to a group of visually related blobs in a certain frame or to different tracks for a set of mobiles. Each hypothesis has associated a likelihood measure, as seen in Equation (5.1).

$$P_H = \sum_{i \in \Omega(H)} p_i \cdot T_i, \tag{5.1}$$

where $\Omega(H)$ corresponds to the set of mobiles represented in hypothesis H, p_i to the likelihood measure for a mobile i (defined in Equation (5.15)), 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. This reliability measure is defined in equation (5.2).

$$T_i = \frac{F_i}{\sum_{j \in \Omega(H)} F_j}, \tag{5.2}$$

where F_i corresponds to the number of frames where the mobile object i has been observed. The temporal reliability is a weight for the global hypothesis likelihood, which is computed according to the life-span of each object, to take into account the number of evidences found for each object.

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.

5.2.2 Dynamics Model

The dynamics model is the process for computing and updating the attributes of the mobile objects. Each mobile object contained in a hypothesis is represented as a set of statistics inferred from visual evidences of their presence in the scene. These visual evidences are stored in a short-term history buffer of blobs representing these evidences, called *blob buffer*. The attributes considered for the calculation of the mobile statistics, belong to the set $A = \{X, Y, W, H, x_p, y_p, w, l, h, \alpha\}$. (X, Y) is the centroid position of the

blob, W and H are the 2D blob width and height in image plane coordinates, respectively. (x_p, y_p) is the centroid position of the calculated 3D parallelepiped base, w, l, and h correspond to the 3D width, length, and height of the calculated parallelepiped in 3D scene coordinates.

The statistics associated to an attribute $a \in A$ 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_0 as the time-stamp of the current frame and t_k the time-stamp of the k-th previous frame, the obtained statistics for each mobile are now described. The mean value $\bar{a}$ for attribute a is defined as the weighted mean between the expected and estimated values of the attribute:

$$\bar{a}(t_0) = \frac{a_{exp}(t_0) \cdot R_{a_{exp}}(t_0) + a_{est}(t_0) \cdot R_{a_{est}}(t_0)}{R_{a_{exp}}(t_0) + R_{a_{est}}(t_0)}, \tag{5.3}$$

where the estimated value a_{est} represents the value of a extracted from the observed visual evidence associated to the mobile (Equation (5.7)), and the expected value a_{exp} for attribute a corresponds to the expected value for current time t_0, given the estimated values for a and the velocity of a at the previous time t_1, and is defined as

$$a_{exp}(t_0) = \bar{a}(t_1) + V_a(t_0) \cdot (t_0 - t_1). \tag{5.4}$$

V_a corresponds to the estimated velocity of a (equation (5.11)). $R_{a_{exp}}$ and $R_{a_{est}}$ are the reliability measures for the expected and estimated values of a, respectively. $R_{a_{exp}}$ is determined as the mean of the global reliabilities R_a and R_{V_a} of a and V_a, respectively, at the previous time t_1.

Global reliability R_a is calculated as the mean between $R_{a_{exp}}$ and $R_{a_{est}}$ at t_0. The reliability measure $R_{a_{est}}$ is calculated as the mean between the visual reliability RD_a (Equation (5.9)) and coherency reliability RC_a (Equation (5.5)) values. $R_{a_{est}}$ is weighted by R_{valid}, which is a reliability measure corresponding to the number of *valid* blobs in the blob buffer of the mobile over the size of the buffer.

For a 2D attribute, a *valid* blob corresponds to a blob not marked as lost, while for a 3D attribute, a *valid* blob corresponds to a blob which has been classified and has then valid 3D information. Lost blobs represent the fact of not finding any blob as visual evidence for the mobile. Not classified blobs correspond to blobs where the 3D classification method was not able to find a coherent 3D solution with respect to the current mobile attributes 3D information.

The coherence reliability measure RC_a accounts for the coherence of attribute a values throughout time. It is defined as

$$RC_a(t_0) = 1.0 - min\left(1.0, \frac{\sigma_a(t_0)}{a_{max} - a_{min}}\right), \tag{5.5}$$

with

$$\sigma_a(t_0) = \sqrt{\frac{e^{-\lambda\cdot(t_0-t_1)}\cdot S_a(t_1)}{S_a(t_0)}\cdot\left(\sigma_a(t_1)^2 + \frac{RD_{a_0}\cdot(a_0 - \bar{a}(t_1))^2}{S_a(t_0)}\right)}, \tag{5.6}$$

which corresponds to the standard deviation of the attribute a. The values a_{max} and a_{min} in (5.5) correspond to pre-defined minimal and maximal values for a, respectively.
The estimated value a_{est} represents the value of a extracted from the observed visual evidence associated to the mobile, and is defined as

$$a_{est}(t_0) = \frac{a_0\cdot RD_{a_0} + e^{-\lambda\cdot(t_0-t_1)}\cdot a_{est}(t_1)\cdot S_a(t_1)}{S_a(t_0)}, \tag{5.7}$$

with

$$S_a(t_0) = RD_{a_0} + e^{-\lambda\cdot(t_0-t_1)}\cdot S_a(t_1), \tag{5.8}$$

where a_k is the value and RD_{a_k} is the visual reliability of the attribute a, extracted from the visual evidence observed at frame k. The visual reliability of an attribute RD_{a_k} changes according to the attribute. In the case of 3D dimensional attributes w, l, and h, these visual reliability measures are obtained with the Equation (4.11). 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.

The visual reliability measure RD_a represents the mean of the reliability measures RD_{a_k}, weighted by the forgetting factor. Similarly to the Equation (5.7) for a_{est}, the visual reliability measure RD_a is incrementally defined as

$$RD_a(t_0) = \frac{S_a(t_0)}{sumCooling(t_0)}, \tag{5.9}$$

with

$$sumCooling(t_0) = sumCooling(t_1) + e^{-\lambda\cdot(t_0-t_1)}. \tag{5.10}$$

All RD_{a_k} values, regardless the concerned attribute a, are weighted by a *visual support factor* ϕ accounting for the quality of visual evidence obtained in the analysed frame for the mobile. This factor allows to differentiate between normally coherent situations and special cases where the visual evidence represent a *lost*, *sub-segmented* or *over-segmented* mobile.

Normal situations correspond to mobiles which attributes are validated with visual evidence, and $\phi = 1$. A mobile is considered *lost* when no visual evidence can be associated to the estimated state for the analysed frame, and $\phi = 0$. If the size of expected 2D blob constructed with the mobile 2D attributes is considerably inferior than the blob considered as the visual evidence, the mobile is considered as *sub-segmented*, and $\phi \in]0, 1[$. In the other side, if the expected 2D blob constructed with the mobile 2D attributes is considerably bigger than the blob considered as the visual evidence, the mobile is considered as *over-segmented*, and also $\phi \in]0, 1[$. In these special cases with $\phi < 1$, the expected state of the mobile is considered to keep the temporal coherence of the mobile attributes with less visual reliability, allowing to cope with segmentation errors and dynamic occlusion.

The value $e^{-\lambda \cdot (t_0 - t_1)}$, present in Equations (5.6), (5.7), and (5.8), 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)).

This way, $a_{est}(t_0)$ value in Equation (5.7) is updated by adding the value of the attribute for the current visual evidence, weighted by the visual reliability value for this attribute value, while previously obtained estimation is weighted by the forgetting factor.

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 defined as

$$V_a(t_0) = \frac{V_{a_0} \cdot RD_{V_{a_0}} + e^{-\lambda \cdot (t_0 - t_1)} \cdot V_a(t_1) \cdot S_{V_a}(t_1)}{S_{V_a}(t_0)}, \tag{5.11}$$

with

$$S_{V_a}(t_0) = RD_{V_{a_0}} + e^{-\lambda \cdot (t_0 - t_1)} \cdot S_{V_a}(t_1), \tag{5.12}$$

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 frame index in time to k. $RD_{V_{a_k}}$ corresponds to the visual reliability of the current instant velocity and is calculated as the mean between the visual reliabilities RD_{a_k} and RD_{a_j}.

The coherence reliability function RC_{V_a} for V_a is defined as

$$RC_{V_a}(t_0) = 1.0 - min\left(1.0, \frac{\sigma_{V_a}(t_0)}{V_{a_{max}} - V_{a_{min}}}\right), \tag{5.13}$$

with

$$\sigma_{V_a}(t_0) = \sqrt{\frac{e^{-\lambda\cdot(t_0-t_1)}\cdot S_{V_a}(t_1)}{S_{V_a}(t_0)}\cdot\left(\sigma_{V_a}(t_1)^2 + \frac{RD_{V_{a_0}}\cdot(V_{a_0}-V_a(t_1))^2}{S_{V_a}(t_0)}\right)}, \tag{5.14}$$

which corresponds to the standard deviation of the attribute velocity V_a. The values $V_{a_{max}}$ and $V_{a_{min}}$ in Equation (5.13) correspond to pre-defined values for the minimal and maximal values for V_a, respectively.

The global reliability R_{V_a} for velocity V_a is calculated as the mean between the visual reliability RD_{V_a} and coherency reliability RC_{V_a} (Equation (5.13)) values, where RD_{V_a} corresponds to the mean visual reliability of measured velocity values for attribute a. R_{V_a} is weighted by RV_{valid}, which is a reliability measure corresponding to the number of valid blob consecutive pairs in the blob buffer of the mobile.

Finally, the **likelihood measure p_m for a mobile** m in Equation (5.1) can be defined in many ways by combining the present attribute statistics. The chosen likelihood measure for p_m is a weighted mean of the probability measures for different group of attributes (groups $\{w,l,h\}$, $\{x,y\}$, $\{W,L\}$, and $\{X,Y\}$), weighted by a joint reliability measure for each group, throughout the video sequence, as presented in Equation (5.15).

$$p_m = \frac{CD_{2D}\cdot RD_{2D} + CD_{3D}\cdot RD_{3D} + CV_{2D}\cdot RV_{2D} + CV_{3D}\cdot RV_{3D}}{RD_{2D} + RD_{3D} + RV_{2D} + RV_{3D}} \tag{5.15}$$

with

$$CD_{3D} = \frac{(RC_w + P_w)\cdot RD_w + (RC_l + P_l)\cdot RD_l + (RC_h + P_h))\cdot RD_h}{2\cdot(RD_w + RD_l + RD_h)}, \tag{5.16}$$

$$CV_{3D} = \frac{MP_V + P_V + RC_V}{3.0}, \tag{5.17}$$

$$CD_{2D} = R_{valid_{2D}}\cdot\frac{RC_W + RC_H}{2}, \tag{5.18}$$

$$CV_{2D} = R_{valid_{2D}}\cdot\frac{RC_{V_X} + RC_{V_Y}}{2.0}, \tag{5.19}$$

where $R_{valid_{2D}}$ is the R_{valid} measure for 2D information, corresponding to the number of not *lost* blobs in the blob buffer, over the current blob buffer size. RD_{2D} is the mean between visual reliabilities RD_W and RD_H, multiplied by $R_{valid_{2D}}$ measure. RV_{2D} is the mean between RD_X and RD_Y, also multiplied by $R_{valid_{2D}}$ measure.

RD_{3D} is the mean between RD_w, RD_l, and RD_h for 3D dimensions w, l, and h, respectively, and multiplied by $R_{valid_{3D}}$ measure. $R_{valid_{3D}}$ is the R_{valid} measure for 3D information, corresponding to the number of not *classified* blobs in the blob buffer, over the current blob buffer size. RV_{3D} is the mean between RD_x and RD_y for 3D coordinates x and y, also multiplied by $R_{valid_{3D}}$ measure.

Measures CD_{2D}, CD_{3D}, CV_{2D}, and CV_{3D} are considered as measures of temporal coherence (i.e. discrepancy between estimated and measured values) of the dimensional attributes (D_{2D} and D_{3D}) and their corresponding velocities (V_{2D} and V_{3D}). The measures RD_{3D}, RV_{3D}, RD_{2D}, and RV_{2D} are visibility measure accumulation in time (with decreasing factor) of the attribute value reliability obtained from the object classification task.

P_w, P_l, and P_h in Equation (5.16) 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 (5.7). Note that parameter t_0 has been removed for simplicity. MP_V, P_V, and RC_V values present in Equation (5.17) are inferred from V_x and V_y. MP_V represents the probability of the current velocity magnitude $V = \sqrt{V_x^2 + V_y^2}$ with respect to a pre-defined velocity model for the classified object, added to the expected object model, defined in the same way as described in Section 4.1. P_V corresponds to the mean probability for the position probabilities P_{V_x} and P_{V_y}, calculated with the values of P_w and P_l, as the 3D position is inferred from the base dimensions of the parallelepiped. RC_V corresponds to the mean between RC_{V_x} and RC_{V_y}.

This way, the value p_m for a mobile object m will mostly consider the probability values for attribute groups with higher reliability, using the values that can be trusted the most.

5.3 Reliability Multi-Target Tracking

In this Section, the proposed tracking method 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 2.2.1. The main differences from these methods are induced by the object representation utilised for tracking, the dynamics model, 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. This consideration implies the conception of new methods for creation and update of object hypotheses.

The complete object tracking process is depicted in Figure 5.3. First, a *hypothesis preparation* phase 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.

Then, the blob-to-mobile correspondences are calculated according to the proximity to the currently estimated mobile attributes to the blobs serving as visual evidence for the

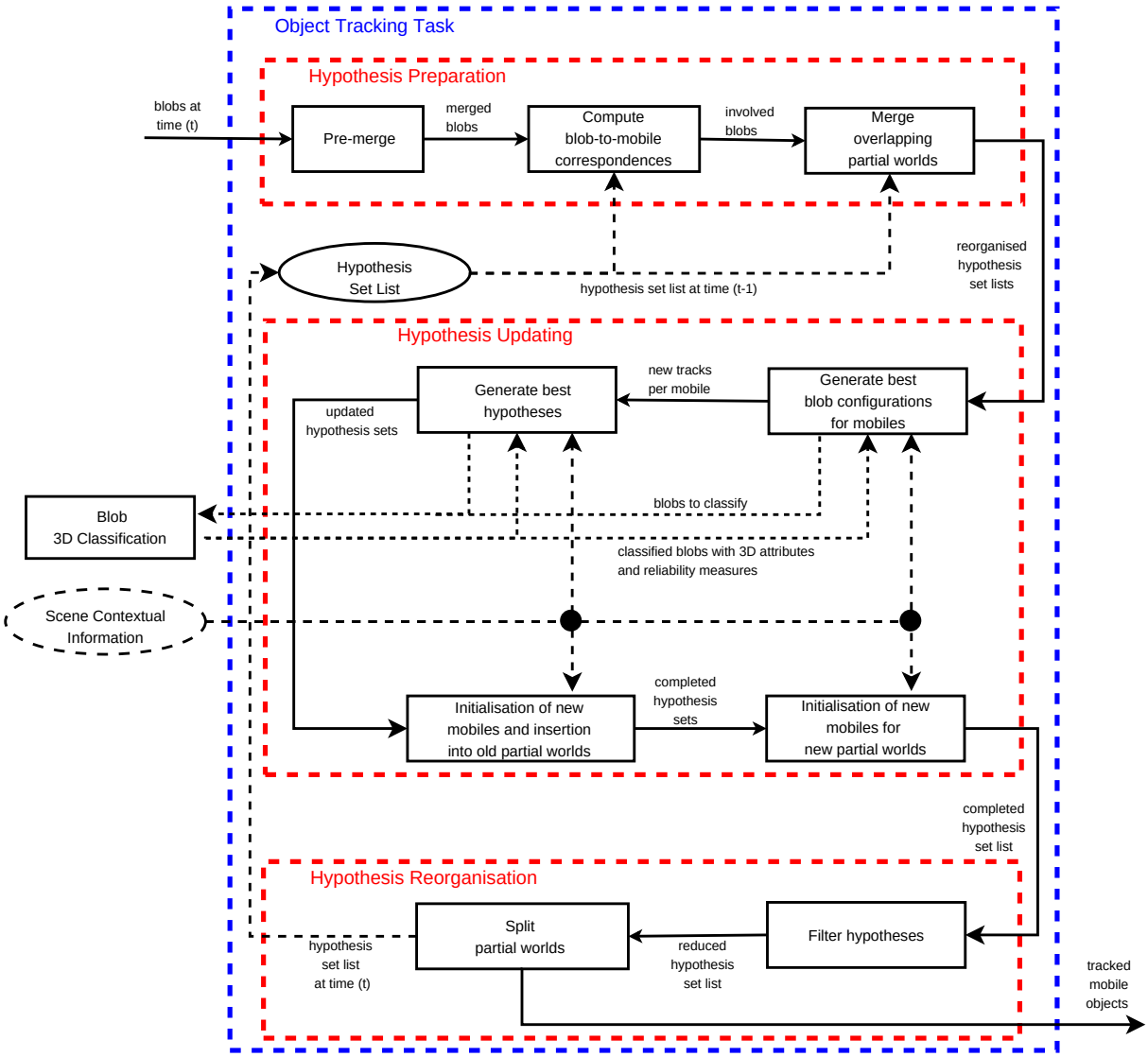


Figure 5.3: The proposed object tracking approach. The blue dashed line represents the limit of the tracking process. The red dashed lines represent the different phases of the tracking process.

current frame. This set of blob 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.

Finally, partial worlds (hypothesis sets) sharing a common set of blobs (visual evidence) are merged, to account for new object configurations produced by this shared visual evidence. The processes involved in the *hypothesis preparation* phase are described in more detail in Section 5.3.1.

Then, a *hypothesis updating* phase starts with the generation of the new possible tracks for each mobile object present in the scene. 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.

These sets of most likely tracks are combined in order to obtain the most likely hypotheses representing the current alternatives for a partial world. The process of generation of hypotheses has been also conceived to immediately generate the best set of hypotheses, instead of generating and pruning.

After, visual evidence not utilised for certain hypotheses of the hypothesis set are considered as the alternative of new objects entering to an existing partial world. Hence, 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 of given 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. This last task completes the description of the world and is the last part of the hypothesis updating phase, which is detailed in Section 5.3.2.

A last phase of *hypothesis reorganisation* is performed to filter lost mobiles, and unlikely or redundant hypotheses. In this phase the last task consists in separating partial worlds where currently the mobile objects are not related.

The tracking process internally updates the hypothesis set list with the updated hypothesis sets. The most likely hypotheses are utilised to generate the list of most likely mobile objects which corresponds to the output of the tracking process.

5.3.1 Hypothesis Preparation

If hypothesis sets already exist at the currently analysed frame, several tasks prior to updating the currently tracked mobiles must be performed in order to prepare the

hypothesis sets for the task of hypothesis updating.

For each mobile belonging to a hypothesis, the *involved blob set* is determined by using the previously obtained mobile attribute information. First, the estimated mobile position is determined from the currently most reliable velocity and position information (2D or 3D position), using the coherence reliability measures RC_a and RC_{V_a}, defined in Equations (5.5) and (5.13), respectively.

Then, the estimated dimensions for the mobile object at the current frame are also determined based on the previous dimensional attribute information, for obtaining the estimated bounding box position and dimensions for the object. This estimated bounding box is enlarged according to the possible variation of the mobile attributes, determined with the standard deviation values for the mobile attributes σ_a, and σ_{V_a}, defined in Equations (5.6) and (5.14), respectively. Finally, if intersection between a blob detected in the current frame and the estimated and enlarged bounding box, is not null, the analysed blob detected in the current frames belongs to the *involved blob set* of the analysed mobile.

In the case that none of the 2D and 3D position and velocity information is reliable, a predefined maximal velocity is considered to determine a variation value used to enlarge the estimated bounding box in all directions, as no velocity direction information is available. This is the common case of first detected visual evidences for a mobile, where the velocity can not be determined or it has a very low reliability.

The involved mobile set is also utilised for determining which blobs can be considered as visual evidence for the current object, and is then used in several processes of hypothesis updating. This way, the determination of the involved blob set for a mobile enables to immediately filter mobile tracks which are very unlikely to occur, corresponding to another mechanism for controlling the combinatorial explosion.

After determining the *involved blob set* for each mobile of a hypothesis, the involved blob set for the hypothesis is determined by performing a union of the involved blob sets of the mobiles. In the same way, the involved blob set for a hypothesis set is determined by performing a union of the involved blob sets of the hypotheses conforming the set.

When every *involved blob set* for the hypothesis sets is determined, it can be detected if different partial worlds, represented by different hypothesis sets, are visually related between them. If this is the case, these hypothesis sets must be merged in order to represent new hypotheses relating mobile objects which can share visual evidence. Two partial worlds are visually related if the intersection between their involved blob sets is not null. If this is the case, the merge process between two hypothesis sets is performed, which consists in generating the hypotheses of the new set by merging every pair of hypotheses that can be constructed from the combination of the two hypothesis sets. The merge process continues until no other intersection of the involved blob sets for two hypothesis set is not null.

When the merging process ends, the resulting hypothesis sets are ready to be updated with the new visual evidence information. This hypothesis updating phase is explained in next Section 5.3.2.

5.3.2 Hypothesis Updating

The first task of the hypothesis updating phase corresponds to the generation of the new possible tracks for each mobile object. A list of the most likely tracks is associated to each mobile object contained in a hypothesis set. This tracks are also represented as mobile objects, updated with the visual evidence extracted from the current video frame.

The track generation method applies two different methods according to the number of frames of mobile life-span. The first method is applied with a life-span of one or two frames, as for first and second frames, it is not possible to determine the coherence of the mobile velocity attributes.

This first generation method consists in considering combinations of all the blobs belonging to the *involved blobs set* of a mobile object for the generation of new tracks. Each generated visual evidence from the combinations of these blobs must be valid in the sense that the utilised blobs must be near between each other.

Each new coherent visual evidence is utilised to generate a track combining the dynamics model information of the analysed mobile with the visual evidence information. The coherence of the new obtained track is checked with the mobile object information obtained in previous frames. If the coherency test is passed, the new track is included in the list of tracks for the analysed mobile. If no coherent association has been found for the analysed mobile, a new mobile is created and tagged as *lost*. The treatment for *lost* objects is described in Section 5.3.4.

Finally, the first generation method ends by limiting the number of possible tracks for a mobile. The new mobiles are suppressed if their likelihood measure p_m, normalised by the best p_m measure, is lower than a pre-defined *MinimalRelativeMobileLikelihood* threshold. Then, the best surviving new mobile number is limited to a pre-defined *MaximumMobileTracks* threshold.

The second generation method is applied with a life-span of more than two frames, as now is possible to determine the coherence of the velocity attributes for the mobile. This generation method consists in using the set of involved blobs to first generate the new evidence associated to the mobile which best fits the estimated bounding box associated to a mobile from its current attribute values, and then generates other mobile tracks using the remaining involved blobs.

If no involved blobs have been found for the analysed mobile, a new mobile is created and tagged as *lost*. The treatment of this case is the same as described in the first mobile generation method.

If only one involved blob has been found for the currently analysed mobile, a new mobile is immediately generated by updating the analysed mobile dynamics with the information extracted from the involved blob. If the analysed mobile is in ensure mode the occurrence of the special situations is analysed, as presented in Section 5.3.4.

When the involved blob set size is higher than one blob, velocity information is available. Hence, the visual evidence can be searched in a neighbourhood of the bounding box generated using the current information of the mobile object.

Using the *involved blob set* of the mobile, the combination of blobs which better covers the current estimated bounding box calculated with the mobile information, is considered as the initial visual evidence for the mobile. Then, other blob combinations are searched using the initial combination as a starting point. Each generated visual evidence is then treated similarly as the first generation method.

The second task of the hypothesis updating phase corresponds to the hypothesis generation task. This task utilises as input the result of the mobile track generation process, and consists in generating for each hypothesis set, the new set of hypotheses with updated mobile information which maximises the hypothesis likelihood measure P_H presented in Equation (5.1). The idea is to immediately generate these best hypothesis sets, instead of generating all the possible hypotheses and then pruning the ones with lower P_H. The hypothesis generation task is independent for each hypothesis set.

The idea of this task is to utilise ordered lists of the best tracks for each mobile object being part of the analysed hypothesis, to sequentially search for the best combinations of these objects, where each of these combinations represent a different new hypothesis. Then, a new hypothesis is considered valid if there is no severe collisions between the parallelepiped bases of the mobile objects which have available and reliable 3D information. If this is the case, the hypothesis is inserted in the list of new hypotheses of the currently analysed hypothesis.

The third task of the hypothesis updating phase corresponds to the generation of new mobiles entering into existing partial worlds. In other words, a process of insertion of new mobiles is performed to associate new mobiles to visual evidence not explained by a hypothesis.

Hence, new mobiles are created for each hypothesis in a hypothesis set, from the set

of blobs which belongs to the *involved blob set* of the hypothesis set, but which does not belong to the *involved blob set* of the hypothesis. This means that the hypothesis represents its tracked mobiles without considering blobs that are used by other hypotheses in the same set, then these not involved blobs can correspond to new blobs entering the scene. All the possible merge combinations are generated and the hypotheses are created now including the information of previously tracked blobs. Then, a hypothesis can become a set of hypotheses, enlarging the hypothesis set it belongs.

The last task of the hypothesis updating phase corresponds to the generation of new mobiles entering into a new partial world. This process is similar to the third task, with the difference that no hypothesis information about existing mobiles must be replicated.

For further details about these methods for track and hypothesis generation, see Appendix B.

Next Section 5.3.2.1 describes the updating process for a mobile object when new visual evidence is found, process which is utilised by the four tasks of the hypothesis updating phase.

5.3.2.1 Mobile Initialisation and Updating

In order to track a mobile object evolving in the video scene, its attribute information must be updated with the information given by the visual evidence associated to the object in the current frame. The process of updating this information is determined by different stages according to the mobile life-span and the coherence of its attribute information.

- First, in order to ensure a minimal evidence of the mobile object existence, the visual evidence on the first frames of existence of the tentative mobile is stored in a *blob buffer*. At these first frames only the 2D information updates the dynamics model presented in Section 5.2.2. This way, the unnecessary classification of blobs that are later lost is avoided, improving the processing time performance.

  The number of frames to be processed with only the 2D information are customisable, but a reasonable value should be considered between three and the size of the blob buffer associated to the mobile. Three values are necessary for a first verification of the temporal coherency of the attribute velocity, as two pairs of blobs are needed for getting two instant velocities. The blob buffer size is taken as an upper bound, which ensures to avoid the loss of information, as blob information leaving the buffer is lost and next step uses this blob information to estimate the initial 3D information.

- Second, when the upper bound for processing only 2D information is reached, the updating process initialises the 3D information.The 3D attribute initialisation process searches, for each blob in the blob buffer (starting from the oldest one), a coherent 3D solution. When a blob is successfully classified, the process searches for

the best configuration among all the classified expected object model classes. Then, each remaining blob in the blob buffer is classified searching for solutions which are coherent with the 3D information of the first classified blob. This means that the solutions are search in a neighbourhood of the attribute values associated to the initial blob.

This guided classification has a twofold benefit: to search 3D parallelepipeds which are coherent with the currently obtained mobile object information, and to guide the 3D classification task in the search of the 3D solution, improving its processing time performance.

All information about other non-optimal coherent 3D solutions for other object classes is also stored in order to give to the mobile attribute updating process the possibility to change the 3D information in case that another object class becomes more likely than the currently selected one.

If the classification of the initial blob does not give any class label or if no coherent 3D solution is found among all classes, only the 2D information is updated and the next blob in the blob buffer sequence is considered as starting point to search for a coherent 3D solution. If no coherent 3D solution is found at all, the mobile is considered as an object of unknown class.

- Third, for the following blob visual evidence associated to the mobile, after obtaining the result from the second process previously described, the attribute updating process continues to apply the guided classification process to classes with a previously found 3D solution, while for classes without associated 3D information, a initial 3D solution is searched. This way, the exploration of the most likely class associated to the mobile continues.

- Fourth, if the number of classified blobs for the currently most coherent class arrives to a pre-defined *minimalNumberOfClassifiedBlobs* and the mobile measure p_m is higher than a pre-defined *minimalMobileLikelihoodToEnsure* threshold, the mobile passes to **ensure mode**. In this updating mode, just the currently most coherent class is evaluated, optimising the performance of the updating process by considering that the currently associated class is the correct one for the mobile object.

5.3.3 Reorganisation of Hypotheses

The final phase of the tracking process corresponds to reorganisation tasks for improving the processing time performance of the approach.

In other way to control the combinatorial explosion in the number of generated hypotheses a chain of hypothesis filters, contained in function *filterHypotheses* (presented in Section 5.3), are applied to the generated hypothesis sets:

1. **Unlikely Mobile Elimination:**

   For each hypothesis set, consider a list of the hypotheses belonging to the set, ordered by their likelihood measure P_H (Equation (5.1)). First, each hypothesis which likelihood measure P_H, normalised by the likelihood measure of the currently best hypothesis, is lower than a pre-defined threshold is discarded. The exceptions are the hypotheses which are constituted only by initial mobiles (detected for the first time) with $P_H = 0$, as these objects are completely unreliable. Then, these initial mobile hypotheses are allowed to survive until more visual evidence is available from the following frames. Finally, just a maximal value of N (pre-defined value) hypotheses of a life-span higher than one frame can survive, which correspond to the hypotheses with higher P_H. As with the previous filtering step, initial mobile hypotheses are not eliminated.

2. **Unseen Mobile Elimination:**

   Unseen or lost mobiles are eliminated depending on their history. An unseen object will be eliminated if:

   - The object has left the scene. In this case, the object information is stored for possible future analysis.
   - The object has been lost before reaching a life-span higher that the blob buffer size. This condition represents the fact that objects which existence has not been sufficiently validated are not allowed to get lost.
   - The object is of unknown class and has been lost for a period longer than the maximum between the blob buffer size and the number of frames it has not been lost. This means that a lost object of unknown type is allowed to survive according to the time it has stayed visible in the scene before.
   - The object life-span is lower than twice the blob buffer size and the number of consecutive frames where the object has been lost is higher than the blob buffer size. This condition imposes the life-span for an object to be considered as sufficiently validated. If an object has been followed for more that twice the blob buffer size, it can not be filtered as unseen.

3. **Repeated Hypothesis Elimination:**

   As mobiles can be individually suppressed from the hypotheses in a set, it is necessary to check if the elimination has caused that now equivalent hypotheses exist in the same hypothesis set. If this is the case, redundant information must be eliminated. Two hypotheses are considered as equal if all mobiles in a hypothesis can be coupled with another mobile in the other hypothesis sharing the same used visual evidence at the current frame, and where the overlapping of the 2D blobs generated from the mobile object attributes are highly similar.

Finally, if after the hypothesis elimination process there are hypothesis sets composed by just one hypothesis, the function *splitHypothesesSets* (presented in Section 5.3) separates the mobile objects composing the hypothesis into individual hypothesis sets composed by one hypothesis with just one mobile object. This can be done because the existence of one surviving hypothesis in a set ensures that the hypothesis has been assumed as the correct one, then all mobile objects contained in the hypothesis are also validated as the correct ones, and can be treated independently.

The main objective of all these presented mechanisms is to control the combinatorial explosion in the number of hypotheses. By controlling this combinatorial explosion the tracking approach naturally tends to sustain an acceptable processing time performance, which can be considered as adequate for several real world applications.

5.3.4 Managing Special Situations

During the tracking process, several special situations can be found, which add complexity to the tracking task and which must be treated in order to obtain a more robust performance of the tracker. Four situations considered fundamental for the robust performance of the tracking process have been addressed:

- *Static Occlusion:* This situation occurs when a mobile object is partially occluded by a static object present in the scene or by the image border.
- *Dynamic Occlusion:* This situation occurs when a mobile object is partially occluded by other mobile objects, producing ambiguous visual support for these objects .
- *Sub-segmented object:* This situation occurs when the segmentation task previous to the tracking process is not able to determine the complete segmented blob, giving partial visual evidence of the object. This situation often occurs due to bad contrast between the mobile object and the background representation.
- *Over-segmented object:* This situation occurs when the segmentation task previous to the tracking process is not able to determine the correct blob limits, giving a visual evidence which covers a larger zone than the object would have covered. This situation can occur due to the presence of shadows, and illumination changes, among other situations.

If the analysed mobile is in ensure mode, it means that it has shown a sufficiently reliable behaviour to test the occurrence of more complex situations, as bad segmentation or dynamic occlusion. If the quality of the new mobile track generated with the involved blob visual evidence is low, these complex situations are analysed applying a sequence of tests between the blob corresponding to the visual support and the estimated bounding box from the current attributes of the analysed mobile.

First, the *blobSupport* and *mobileSupport* measures are calculated considering the intersection between the blob serving as visual evidence and the bounding box generated

from the expected mobile attribute values. These measures correspond to coverage rates in the interval $[0.0; 1.0]$.

The *blobSupport* measure accounts for the coverage ratio of the estimated bounding box by the visual evidence blob, where the maximal value 1.0 is obtained when the bounding box obtained from the mobile attribute estimation is contained by the visual support blob.

The *mobileSupport* measure accounts for the ratio of coverage of the visual evidence by the estimated bounding box, where the maximal value 1.0 is obtained when the blob serving as visual evidence is contained by the bounding box estimated with the mobile attributes.

If both *blobSupport* and *mobileSupport* are higher than the *HighVisualSupportRate* threshold, the association blob-mobile is considered as a normal situation of good quality. The *HighVisualSupportRate* has shown the desired results at a rate of 0.95. This situation is considered as normal because a high rate for both measures implies that the visual evidence is in concordance with the estimated mobile attribute values. This situation is depicted in Figure 5.4(a).

(a) (b)

Figure 5.4: Normal situations determined after analysing *blobSupport* and *mobileSupport* measures. Red box represents the visual evidence, while green box represents the estimated bounding box generated from the mobile information. The yellow zone represents the intersection between both blobs. Figure (a) represents the normal situation where visual evidence corresponds in size and in position to the expected attributes of the mobile object. Figure (b) depicts the normal situation where visual evidence corresponds in size to the expected attributes of the mobile object, but not in position.

If the later test has not classified the situation as normal, it means that it is still possible that a special situation is occurring. A second test is then performed in order to be sure that the situation is a special case. First, the differences in width and height, between

the blob serving as visual evidence and the bounding box estimated from the mobile attributes, are calculated.

Then, the tolerance of the width and height differences considered as normal are determined from the mobile attributes W and H standard deviations, inferiorly bounded by a minimal pixel tolerance. If the width and height tolerances comply with the width and height differences, the analysed situation is also considered as normal. This is because the blob can be considered as acceptable visual evidence for the mobile, but it is the position of the blob which does not fit properly the position proposed by the mobile attributes. This situation is depicted in Figure 5.4(b).

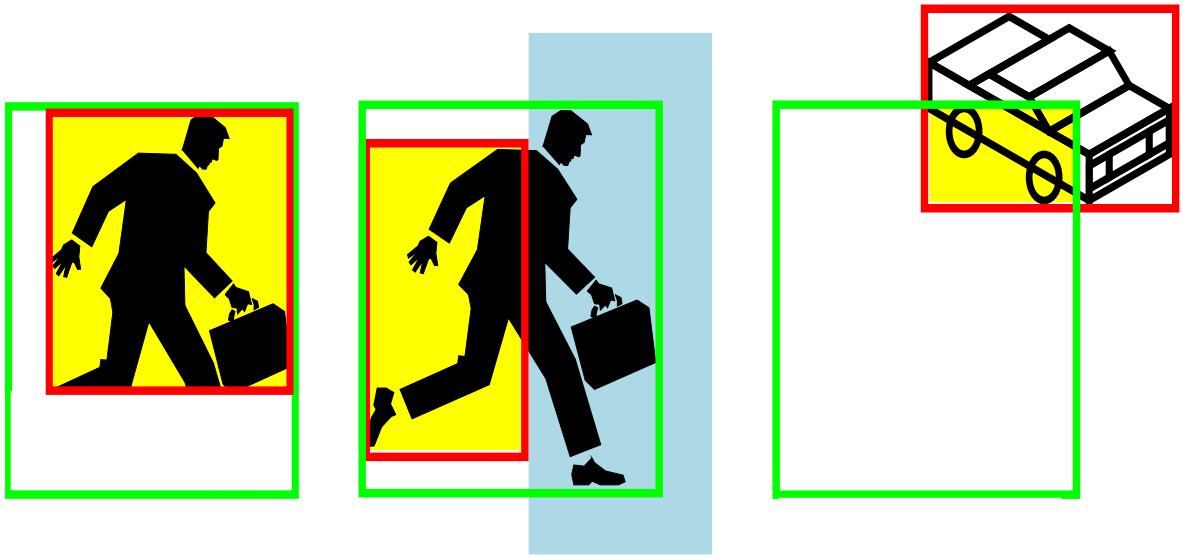
If the situation is still not considered as normal, now remaining cases can correspond to the different special situations. To determine the right situation, a sequence of tests is performed. The first test evaluates if the *mobileSupport* is higher than the *HighVisualSupportRate* threshold and if the area of the visual support blob is lower than the area of the bounding box estimated from the mobile attributes. If this the case, as the first test has failed, it implies that the *blobSupport* is lower than the *HighVisualSupportRate* threshold. This corresponds to situations where the bounding box estimation by mobile attributes is weakly covered by the visual support blob, as depicted in Figures 5.5(a) and 5.5(b).

Then, in this case this situation can represent a *sub-segmented* object, or a *static occlusion* case. To differentiate both cases it is sufficient to analyse the possibility of occlusion for the visual evidence, and evaluate if the not visible part is in a zone of possible occlusion. For further information about how to determine the occlusion zones refer to Section 4.2.1.

If the previous test fails, the second test evaluates if the *mobileSupport* is lower than the *highVisualSupportRate* threshold and if the *blobSupport* is lower that a pre-defined *lowVisualSupportRate* threshold. This situation corresponds to a *lost* object, as depicted in Figure 5.5(c). In practise, the *lowVisualSupportRate* threshold has been set to 0.05. Here, the visual support blob is not highly contained by the bounding box estimated with the mobile attributes, and the estimated bounding box is weakly covered by the visual support, meaning that it does not exist enough evidence to associate the visual evidence to the mobile object.

For updating the analysed mobile considering a *lost* visual evidence, the counter of lost blobs is incremented and the dynamics model updates the current state of attributes using previously obtained information. The reliabilities are updated considering the currently obtained information as not valid.

Figure 5.5(c) depicts this situation with an erroneous case where the visual evidence really corresponds to a vehicle evolving in a farther position. In practise, the blob could really correspond to infinite possibilities, as noise, illumination changes, shadows, and so on.



(a) (b) (c)

Figure 5.5: Special situations where the object is partially or not validated by the visual evidence. Red box represents the visual evidence, while green box represents the estimated bounding box generated from the mobile information. The yellow zone represents the intersection between both blobs, while the blue zone represents an occlusion zone. Figure (a) represents the special situation where the object is sub-segmented (legs not detected). Figure (b) depicts the special situation where the object is not completely detected because it is partially occluded. Figure (c) represents a situation where the object is lost, and the blob supposed to correspond to visual evidence does not correspond to the analysed tracked object.

If the lost mobile test fails, a last test remains which evaluates if the *blobSupport* is higher than the *HighVisualSupportRate* threshold and if the area of the visual support blob is higher than the area of the bounding box estimated from the mobile attributes. As the first normal test has failed, this means that the visual evidence is considerably bigger than the estimated bounding box from the mobile attributes, having an *over-segmented* object situation, as depicted in Figure 5.6.

This situation can correspond to two more specific cases: an *over-segmented* object due, for example, to the presence of shadows or illumination changes (Figure 5.6(a)), or dynamic occlusion situation, where more than one object share common visual evidence (Figure 5.6(b)). To determine which of these situations can be really happening, dynamic occlusion can be detected by analysing the mobile objects in a same hypothesis, which share the same visual evidence.

In the static occlusion and *sub-segmented* object situations, the analysed mobile is updated by considering the *visual support factor* ϕ (defined in Section 5.2.2) equal to the

(a) (b)

Figure 5.6: Special situations where the visual evidence is considerably bigger than the estimated bounding box from the object attributes. Red box represents the visual evidence, while green box represents the estimated bounding box generated from the mobile information. The yellow zone represents the intersection between both blobs. Figure (a) represents the special situation where the object is over-segmented by the segmentation of a shadow as part of the object. Figure (b) depicts the special situation where the object is part of a dynamic occlusion situation where the visual evidence corresponds to more than one object.

blobSupport measure, accounting for the coverage rate of the mobile object.

In the same way, in the dynamic occlusion and *over-segmented* object situations, the analysed mobile is updated by considering the *visual support factor* ϕ equal to the *mobileSupport* measure, accounting for the coverage rate of the blob visual support and weighting this way the possible error committed in the estimation of the new mobile track attributes.

For the four special situations, before updating the current attributes of the mobile for the current frame, the estimated bounding box limits are adjusted with the visual evidence blob, in order to improve the visual support factor by adjusting the estimated bounding box limits to lie within the visual evidence blob limits, and then recalculating the mobile attributes with the adjusted blob. This way, all these special situations can be treated in a consistent way.

5.4 Illustration of The Tracking Approach

The tracking approach is illustrated with a video sequence with one person performing diverse activities in a furnished apartment. This scene presents various features which allow to test the capability of the proposed tracking method: occluding furniture, strong illumination changes, shadows, and reflections, among other features.

The tests were performed with a computer with processor Intel Xeon CPU 3.00 GHz, with 2 Giga Bytes of memory. The test consists of two clips delimited by the person entering and leaving the scene in different ways. A total of 587 analysed frames (1 minute and 13 seconds) has been considered, testing the quality of trajectories, the capability of differencing between two postures and the processing time performance. Videos of test results are available at the website:

http://www-sop.inria.fr/pulsar/personnel/Marcos.Zuniga/name-of-video.avi

where *name-of-video* are *gerhome-clip1* (a short clip with one severe occlusion and posture change), *gerhome-clip2* (a zone with strong shadows and reflexions), or, *gerhome-segmentation-clip2* which corresponds to the video of the segmentation result with associated blob bounding boxes, used as input of the tracking algorithm.

5.4.1 Results

In Table 5.1 and 5.2 a summary of the obtained results is presented.

Sequence	**Length**	**Mean Time [sec/frame]**	**Frame Rate[frames/sec]**
gerhome-clip1	115	0,0063	159,7470
gerhome-clip2	472	0,0180	55,4604
Mean		**0,0157**	**63,6625**

Table 5.1: Evaluation of results obtained for both analysed video clips in terms of processing time performance.

In Table 5.1, Mean Time corresponds to the mean time per video frame (in seconds), while Frame Rate corresponds to the number of frames per second that the tracking algorithm can process. The frame rate for an isolate mobile proves that the system is able to achieve high processing time performance, but it is not enough to validate the processing time performance of the approach in more complex situations, because the interaction of several objects in the same image zone can produce a significant increment of initial hypotheses.

Sequence	Length	Good Trajectories	Trajectory Rate
gerhome-clip1	115	113	0,9826
gerhome-clip2	472	442	0,9364
Total	**587**	**555**	**0,9455**

Table 5.2: Evaluation of results obtained for both analysed video clips in terms of processing time performance.

In Table 5.2, **Good Trajectories** is the number of trajectory points which were at reasonable position (as evaluated by a human observer) with respect to the person present in the scene. **Trajectory Rate** is the rate of good trajectories with respect to the total number of analysed frames. In both video clips, the algorithm has never lost its target, even in presence of severe occlusion. The points considered out of the trajectory were visibly far from the centre of the base of the tracking target. Also, the postures standing and crouching were analysed, obtaining a success rate of $91,31\%$.

In Figures 5.7 and 5.8, some image frames from the analysed videos can be found. Left images correspond to the image segmentation input. Right images are the corresponding tracking algorithm result.

In Figure 5.7, the first frame pair from top to bottom presents a person coming from left door who crouches and hides behind the couch. This frame is challenging because previously the person was standing and, at the same time, the couch is occluding the legs of the person. The algorithm is quite successful in estimating both the 3D bounding box of the object and its position in the 3D referential of the scene. Second pair presents the solution for a sub-segmented person where his legs are almost not segmented. Instead tracking algorithm finds a good estimate of the real position of the object and its 3D bounding box.

In figure 5.8, the first frame pair from top to bottom presents a person coming from the entrance door of the apartment with reasonably good segmentation. Second pair presents the solution for a poor segmentation frame where legs of the person are not segmented at all. For this situation, the tracking algorithm finds a good estimate of the real position of the object and its 3D bounding box anyway. Third pair shows a frame where the visual evidence of the person is segmented in two pieces, but the tracking algorithm corrects this situation. Fourth image pair shows the case of an image reflection and shadows present in the scene, with a good overall response of the tracking approach.

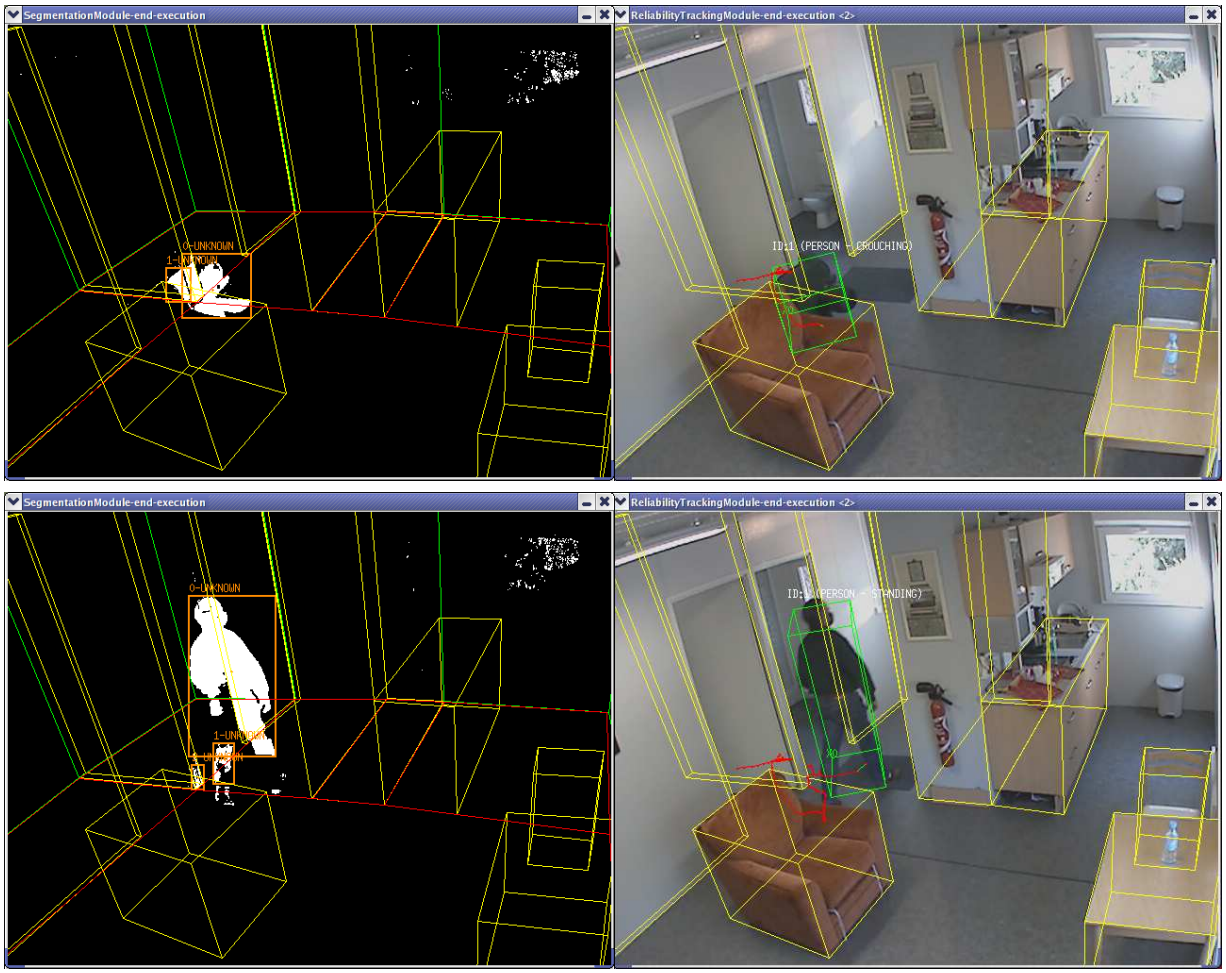


Figure 5.7: Images from the analysed video clip *gerhome-clip1*. Left images correspond to image segmentation input to tracking algorithm. Right images are the corresponding output of the proposed tracking approach. From top to bottom, first image pair correspond to frame 37 of this clip, where the person coming from left door hides behind the couch, with a successful estimation of the real position and dimensions of the crouched person. Second, frame 97 presents the solution for a poor segmentation frame.

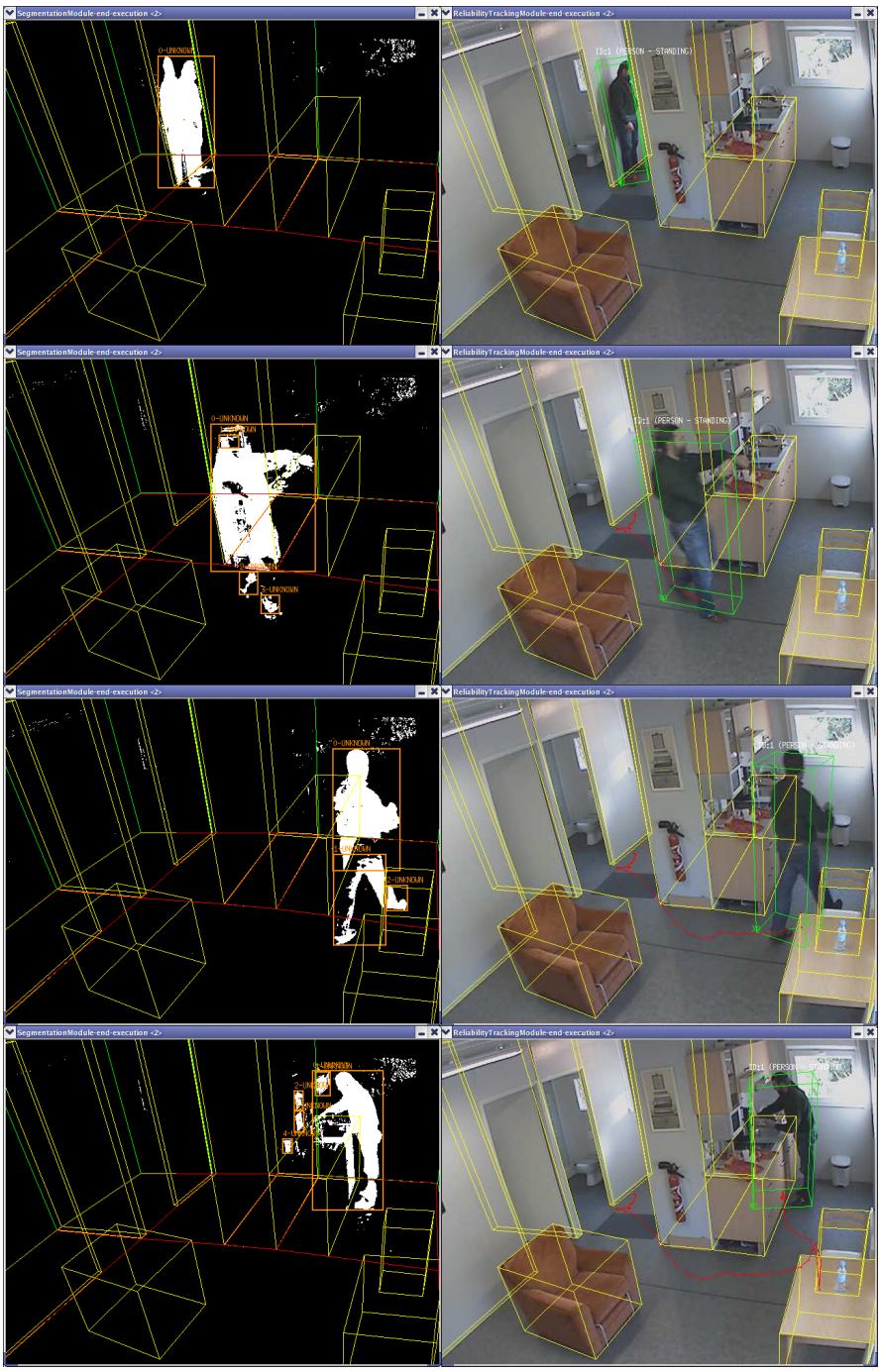


Figure 5.8: Images from the analysed video clip *gerhome-clip2*. Left images correspond to image segmentation input to tracking algorithm. Right images are the corresponding output of the proposed tracking approach. From top to bottom, first image frame 3 corresponds to the person passing the entrance door. Second frame pair 24 presents the solution for a sub-segmented person. Third frame pair 31 shows the resolution of a split visual evidence problem. Fourth frame pair 68, shows a case with shadows and reflection.

5.4.2 Experiment Conclusion

The preliminary tests performed show that the tracking approach is able to achieve a adequate processing time performance. Extensive testing is needed in order to establish the limitations and potential of the approach. The proposed tracking approach has shown its capability of tracking a target even if the segmentation is of bad quality. Following tests will be oriented to the interaction with other mobile objects. This approach can solve a large number of different static occlusion situations. Nevertheless, just simple and partial dynamic occlusions can be solved by keeping the motion coherence, as no appearance models are utilised for this approach.

Reliability measures provide a simple way of determining the quality of obtained information. These measures have helped in the robustness of the approach by allowing the proper consideration of most reliable information. They are also used in the event learning task of the proposed video understanding framework to consider the most coherent information, validated in time by the tracking approach.

5.5 Discussion

The proposed tracking approach presents several features which aims at obtaining a processing time performance which is adequate for real world applications. These features can be found through all the tracking process:

- The proposed tracking approach explicitly cooperates with the 3D classification process (described in Chapter 4), by guiding the classification process using the previously learnt mobile object attributes. This way, the tracking process is able to indicate a starting point and the bounds of search for the parallelepiped attributes to be found by the classification approach, as described in Section 5.3.2.1.

  This cooperation scheme allows a reduction in the processing time dedicated to 3D classification. As mobile information can become more reliable as more visual evidence is available, the cooperation scheme can be also considered to improve its quality in time, as more reliability implies a more accurate mobile dynamics model and less variability of mobile attributes, establishing tighter bounds to the search space.

- When the mode of a mobile object becomes the *ensure mode*, even a better processing time performance can be obtained by the 3D classification process, as the parallelepiped is estimated just for one object class. 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 (for details, see Section 5.3.2.1).

- The determination of the involved blob sets, described in Section 5.3.1, allows to control the number of possible blob associations for a mobile object and to separate

the tracking problem into sub-problems according to the proximity of the blobs representing the visual evidence.

Then, the involved blob sets determination presents a two-fold contribution to the early control of the combinatorial explosion, as less possible associations per mobile and less related mobiles per tracking sub-problem imply the immediate reduction in the number of hypotheses to generate, contributing to the improvement of the processing time performance.

- The hypothesis updating process, presented in Section 5.3.2, have been oriented to optimise the estimation of the updated hypothesis set, in order to obtain the most likely hypotheses avoiding to generate unlikely hypotheses that must be eliminated later. The generation of the mobile tracks utilises a similar principle, generating the initial solution nearest to the estimated mobile attributes, according to the available visual evidence, and then generating the other mobile track possibilities starting from this initial solution.

  This way, the generation is focused on optimising the processing time performance by immediately generating good quality solutions, instead of generating all the possible combinations and pruning the solutions with bad quality.

- Even if the hypothesis updating process is focused in generating the minimal possible number of hypotheses, the processing load for the next frame can be reduced by filtering redundant, not useful, or unlikely hypotheses, as described in Section 5.3.3.

- Finally, the split process for hypothesis sets, also presented in Section 5.3.3, represents another mechanism to improve the processing time performance as it immediately reduces the number of mobiles in a same hypothesis set, generating different hypothesis sets, which can be treated as separated tracking sub-problems.

Several of the presented ideas in the proposed tracking approach are not new and find their parallel in the literature, as [Avanzi et al. 2001]. The different *Screening* techniques [Kurien 1990], presented in Section 2.2, can be found in the algorithm in the following ways:

- The *gating* technique is similar to the method for determination of the involved blob sets. The main difference lies in the fact that the gating technique is focused in finding possible correspondences for points, which can only participate in a one-to-one association with a mobile, while the involved blob set determines the blobs which can correspond to a part of the visual evidence of the mobile or blobs which can represent the visual evidence for several mobiles, consequently allowing associations one-to-many (object segmented in parts) and many-to-one (dynamic occlusion).

- The *clustering* technique can be found in the proposed tracking algorithm in the form of the *mergeHypothesesSets* (Section 5.3.1) and *splitHypothesesSets* (Section 5.3.3) functions, which define when a set of mobiles must be considered as part of the same tracking sub-problem or not.

- The confidence level concept (or mobile *age*) proposed by the *classification* technique is similar to the utilisation of the *ensure* mode for mobiles when mobile attribute reliability is high and in the consideration of only 2D information for the first frames of life-span of a mobile, when information is highly unreliable (Section 5.3.2.1). It can be considered an improvement with respect to the *classification* technique the fact that the *ensure* mode is not just based on the *age* of the mobile, but on the reliability of its attributes.

The *Pruning* techniques also presented in Section 2.2, can also find its parallel in the proposed tracking approach. Compared with the *Lower probability* technique, the proposed tracking approach disposes of the *Unlikely Mobiles Elimination* filter (Section 5.3.3), which differs from the *Lower probability* technique in that the filter normalises the likelihood of the hypotheses with respect to the best hypothesis of the set, in order to ensure that at least one hypothesis remains and that the filtering process is independent from the average quality of the hypothesis set.

The *n-Scan Approximation* pruning technique finds its parallel in the utilisation of a blob buffer by the proposed tracking approach, for storing the visual evidence associated to a mobile in the later frames. As the n-scan approximation technique uses the information from few consecutive frames for assigning the measurements to mobiles, this blob buffer limits the search of a mobile object solution to some few frames before the current frame.

The proposed tracking method has shown that is capable of achieving an adequate 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 of solving static occlusion, sub-segmentation, and object segmented into several parts problems.

The tracking approach can also solve dynamic occlusion situations by maintaining the temporal coherence of the set of occluding objects, and by checking the validity of the new possible solutions in terms of 3D model collisions. As the tracking approach does not use object appearance information, it can only solve dynamic occlusion situations where involved temporal attribute coherency is maintained. One of the considered aspects in future work is the inclusion of object appearance models for coping with more complex dynamic occlusion situations. The dynamic occlusion problem resolution capability has still to be validated.

The tracking approach utilises the reliability measures to control the uncertainty in the obtained information, learning more robust object attributes and establishing 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. Briefing, the proposed multi-object tracking approach presents the following main contributions:

1. A new dynamics model for object tracking which computes the tracking likelihood

in an optimised way, given the available information. This dynamics model includes several measures of reliability associated to real physical notions. Moreover, the computation of these measures are accumulated throughout time by a summation of the different notions weighted by a forgetting factor. Thanks to this weighted summation, tracking reliability is naturally normalised by the a priori reliability of the physical notion. This approach contrasts with the state of the art, where most tracking approaches (MHT or particle filtering) update the tracking likelihood by the joint probability of current and past likelihood, requiring a non intuitive normalisation of the tracking likelihood.

2. Explicit interaction between the tracking and classification tasks, allowing the achievement of a higher processing time performance.

3. New methods for best object hypothesis generation in order to ensure a tracking performance adequate for real world applications.

4. A new multi-hypothesis algorithm for tracking multiple objects in noisy environments, for real world applications. The approach partially copes with static occlusion, several situations with dynamic occlusion, and poorly segmented objects (e.g. divided in several parts, or with some misdetected part). Dynamic occlusion can be addressed if tracked objects have a high motion coherence before occlusion, as this approach does not include appearance model information.

The proposed object tracking approach presents the following limitations:

1. The first limitation is related to dynamic occlusion situations. The tracking approach is able to cope with dynamic occlusion utilising the object attribute information estimated in the previous frames to estimate the current values for the object attributes. As the tracking approach only estimates the current attributes based on previous information, the behaviour of the objects during the occlusion period can not be determined, which can lead to tracking errors, such as mistaken tracks.

2. A second limitation corresponds to the incapability of the tracking approach to identify an object leaving the video scene and the re-entering in the scene as the same object. This is due to the geometrical nature of the information utilised for tracking and due to the no utilisation of appearance models of the tracked objects.

3. Third, the quality of the tracking task depends on the segmentation and the classification results. Thus, situations with crowd and strong shadows can still be a challenge.

4. Fourth, the tracking algorithm requires the tuning of several parameters. Further analysis has to be performed in order to automatically tune these parameters according to the application.

5. Finally, a third limitation can be identified with respect to the processing time performance on scenes with a high number of objects evolving in the scene. Even if the hypothesis generation process of the tracking approach has been optimised a large number of objects simultaneously entering the scene can produce a high number of initial object configuration hypotheses as no object information is available when a new object enters the scene. This limitation is relative to the application, as a high processing time performance is not always a requirement for all application.

The mobile objects resulting from the tracking process are utilised as input by the last task of the proposed video understanding framework, corresponding to incremental event learning, which is described in detail in the following Chapter 6.

Chapter 6

Incremental Event Recognition and Learning

In this chapter, a new method for incremental learning of events in videos is presented. This learning method is a component of the video understanding framework presented in Chapter 3, as depicted in Figure 6.1. This event learning method takes as input the mobile objects which are the result of the previous object tracking task presented in Chapter 5.

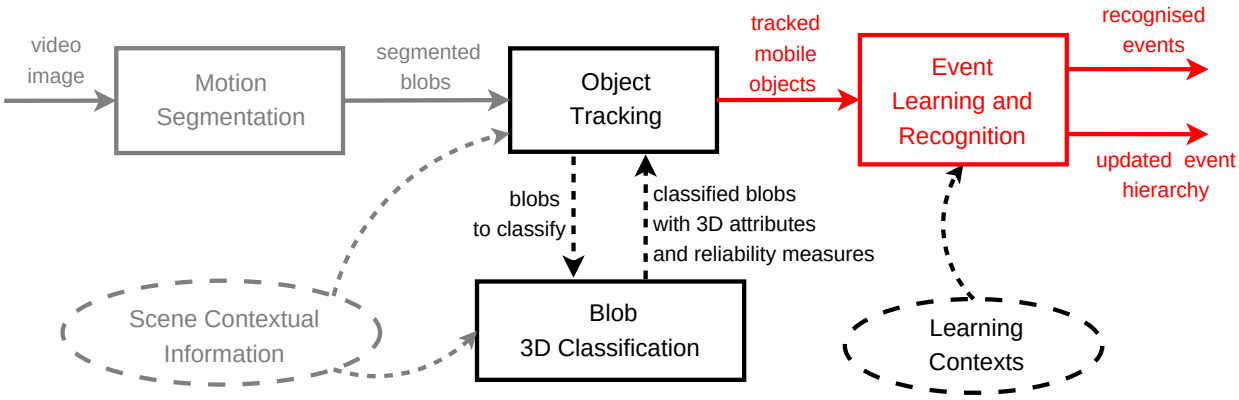


Figure 6.1: Proposed event learning approach as a component of the video understanding framework. Black elements correspond to the contributions of this thesis work. Gray elements correspond to elements used by the proposed framework, but not forming part of the contributions of this work. Red elements correspond to the elements analysed in this chapter, related with the proposed event learning method.

The event learning method is based on models of incremental concept formation ([Gennari et al. 1990], [Carbonell 1990]). The models of incremental concept formation allow to

incrementally build a concept hierarchy, by updating the hierarchical concept structure with the arrival of each new data instance. These models also allow the recognition of a new instance, based on the inferred concepts from previously processed data. In the context of the proposed learning method, a concept corresponds to a state, and data correspond to the visual attributes of mobile objects present in the video scene.

The input data of this method correspond to object visual attribute values together with a reliability measure for each attribute, obtained from the multi-object tracking approach. These reliability measures represent the temporal coherence of the tracked object attributes, and are used to perform a proper selection of the relevant information for the learning approach.

The new incremental learning algorithm proposes an extension of the models of incremental concept formation, by expanding the representation of concepts to the first-order temporal relations (i.e. Markov hypothesis) between these concepts. Thus, in the context of the proposed learning approach, concepts (represented as nodes in the hierarchy) become the states induced by the tracked objects present in the scene, while the first-order temporal relations, representing the state transitions, become the learnt events. Therefore, the learning approach is able to incrementally generate a hierarchical representation of the states and events occurring in the scene. Information about the frequency of occurrence of these states and events is also calculated, which allows to determine if the current state and event of an object is normal or abnormal in terms of frequency. The utilised hierarchical representation presents concepts describing more general states in the top of the hierarchy, while the sibling state concepts in the hierarchy represent specifications of their parent.

For guiding the learning process, it is necessary to pre-define the *learning contexts*. A learning context corresponds to a description of the scope of the events of interest for the user. It is defined as a set of object attributes, where these attributes are numerical or symbolic. For the numerical attributes, it is necessary to associate a discrimination value, which represents the granularity of interest for this attribute. As the attributes defined in the learning context can be numerical, normalisation values have to be associated to these attributes, for corresponding to a meaningful variation of the attributes. A normalisation value associated to an attribute is known as the *acuity* of the attribute.

Several learning contexts can be simultaneously processed by the proposed approach, generating for each of them a different resulting hierarchy of states and events. Then, for each learning context, the event learning method extracts the appropriate available information according to the currently tracked objects in the scene. Then, state instances are created for each tracked object. These instances are classified through the hierarchy of states and the information of the instance is used to update the state hierarchy. Each state concept in the hierarchy is described by its frequency of occurrence, and by descriptions of the attribute values it represents.

Each tracked object can participate to more than one learning process at the same time, if this object is allowed according to the associated learning context. The state and event hierarchies are learnt combining the information provided by all the allowed mobile objects being tracked.

For the symbolic attributes of a state, all their possible values are listed and a frequency of occurrence value is associated, according to the number of instances which are considered for the attribute value. Numerical attributes are represented by the mean and standard deviation of the attribute values for the collected instances in the state concept.

Then, when an instance is classified, the associated state concept description is updated with the attribute information of the instance, considering the reliability measures associated to the attributes for weighting the contribution of this new information to the model of the attribute.

The learning algorithm keeps track of the current state of each mobile object. When an object changes of state, the event information is updated or created if it is the first occurrence of this event. Each event concept contains mean and variance information about the time of permanence of the mobile object in the previous state. This information can be very useful to understand the behaviour of objects evolving in the scene.

Hence, the result of the learning process corresponds to a learnt hierarchy of states and events for each pre-defined learning context, and the currently recognised state and event for each object evolving in the scene. As the utilised event learning approach is incremental, the process of learning and recognition occurs simultaneously.

This chapter is organised as follows. First, in Section 6.1, the event learning contexts are formally presented. Then, the structure representing the learnt states and events is described. Second, Section 6.2 presents MILES algorithm, a new incremental event learning approach. This section presents the utilised data representation, the utilisation of reliability measures for guiding the learning process, the operators for updating, expanding, and contracting the event learning structure, and a detailed description of MILES learning algorithm. Third, Section 6.2.4 presents an illustration of the proposed incremental event learning algorithm. For this purpose, ten hand-crafted trajectories of eight frames each have been analysed, in order to explain the mechanics of the learning process and to understand how the real world situations are represented in the approach. Finally, in Section 6.3, remarks about the learning approach are discussed.

6.1 Description of the Learning Data

The information utilised by the proposed event learning approach corresponds to the mobile objects tracked by the previous object tracking task (presented in Chapter 5) and to the *event learning contexts* pre-defined by the user. Each learning context guides

the extraction of the appropriate features from the mobile object attributes, in order, to prepare the proper input for each learning process.

Each learning process constructs a hierarchy of states, based on the information received at each video frame. When a mobile object passes from one state to another in a given hierarchy of states, this change of states corresponds to an event. The event representation is linked to the states triggering this event, and information about the time spent by the starting state before passing to the next state is stored.

Then, in order, to feed the learning processes with the proper information, the tracked object information must be also extended to represent the information required by each concerned learning process and to store the current state and event information for the mobile object.

Next two Sections are dedicated to explain in detail the representation utilised for the information in the proposed event learning approach. First, Section 6.1.1 focuses on a detailed description of the hierarchy of states and events. Second, Section 6.1.2 focuses in formalising the definition event learning contexts and processes, together with the necessary extension to the representation of learning information for a mobile.

6.1.1 Hierarchical Events Tree

The proposed event learning approach utilises a hierarchy tree for representing the states, in the same way as proposed by [Fisher 1987], and discussed in Section 2.4.3. This representation is extended to also consider the occurrence of events as the transition between these states. More formally:

Definition 6.1 *A **hierarchy of states and events** H is defined as set of states organised hierarchically, with a set of events representing the transitions between these states. The states are hierarchically organised by generality, with the states higher in the hierarchy being more general, while the children of each state represents a specification of its parent. There is no limit to the number of children of a state. Pairs of learnt state concepts are linked by the event representation, which represents the unidirectional fact of passing from one state concept to another.*

An example of a hierarchy of states and events is presented in Figure 6.2. In the example, the state S_1 is a more general state concept than states $S_{1.1}$ and $S_{1.2}$, and so on. Each pair of states ($S_{1.1}$; $S_{1.2}$) and ($S_{3.2}$; $S_{3.3}$), is linked by two events, representing the occurrence of events in both directions.

A state S is represented in the hierarchy in the following way:

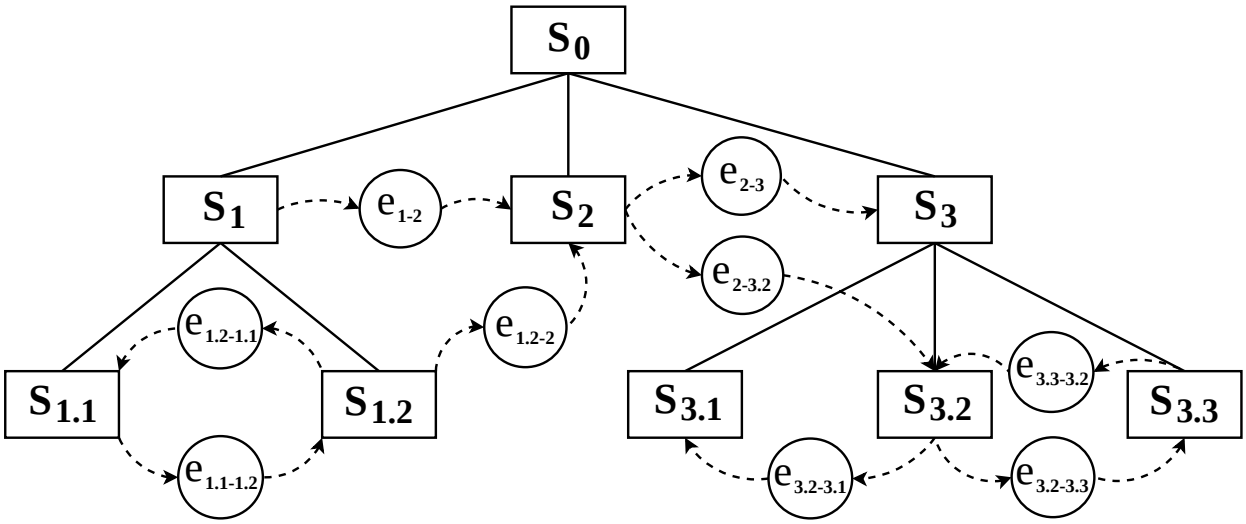


Figure 6.2: Example of a hierarchical event structure resulting from the proposed event learning approach. Rectangles represent states s, while circles represent events e. An event represents the unidirectional transition between two states.

State S {
Probability of Occurrence: $\mathcal{P}(S)$
Number of Represented Instances: $N(S)$
Number of Event Occurrences: $N_E(S)$
Attributes:
Numerical $n_1 \sim \mathcal{N}(\mu_{n_1}\ ;\ \sigma_{n_1})$
$\vdots$ $\vdots$
Numerical $n_M \sim \mathcal{N}(\mu_{n_M}\ ;\ \sigma_{n_M})$
Symbolic s_1 : {
$V_{s_1}^{(1)} \leftarrow \mathcal{P}(s_1 = V_{s_1}^{(1)}|S)$
$\vdots$
$V_{s_1}^{(L_1)} \leftarrow \mathcal{P}(s_1 = V_{s_1}^{(L_1)}|S)$ }
$\vdots$ $\vdots$
Symbolic s_P : {
$V_{s_P}^{(1)} \leftarrow \mathcal{P}(s_P = V_{s_P}^{(1)}|S)$
$\vdots$
$V_{s_P}^{(L_P)} \leftarrow \mathcal{P}(s_P = V_{s_P}^{(L_P)}|S)$ }
}

The probability of occurrence $\mathcal{P}(S)$ for a state S corresponds to the number of occurrences for the state in the video sequence, over the number of occurrences for its parent state concept. The number of represented instances $N(S)$ represents the number of times that

an object instance has been classified as the state S. The number of event occurrences $N_E(S)$ represents the number of times that state S passed to another state, generating an event.

Each numerical attribute n is considered to follow a Gaussian distribution $n \sim \mathcal{N}(\mu_n; \sigma_n)$, with μ_n corresponding to the mean value for the attribute and σ_n to its standard deviation. Each symbolic attribute is represented by every defined value for the attribute, and a conditional probability $\mathcal{P}(V_s^{(i)}|S)$ associated to each value, with $V_s^{(i)}$ being the i-th possible value for the attribute s.

An event E is represented in following way:

Event E {
Number of Occurrences: $N(E)$
Probability of Occurrence: $\mathcal{P}(E)$
Starting State: S_a
Arriving State: S_b
Starting State Time $T_{S_a} \sim \mathcal{N}(\mu_{T_{S_a}}\ ;\ \sigma_{T_{S_a}})$
}

The number of occurrences $N(E)$ corresponds to the number of occurrences for the event E in the video sequence. The probability of occurrence $\mathcal{P}(E)$ for an event E, then corresponds to the number of occurrences $N(E)$, over the number of event occurrences generated from its starting state concept $N_E(S_a)$. The event represents the change from state S_a to S_b, in that order, as the inverse order implies the occurrence of another event (see Figure 6.2). It is also estimated the time spent in state S_a before passing to state S_b, defined as $T_{S_a} \sim \mathcal{N}(\mu_{T_{S_a}}; \sigma_{T_{S_a}})$, modelled as a Gaussian distribution with $\mu_{T_{S_a}}$ corresponding to the mean value of the time T_{S_a} and $\sigma_{T_{S_a}}$ to its standard deviation.

As an example of the consideration of all these defined elements forming the described hierarchy of states and events, Figure 6.3 shows the state and event hierarchy obtained considering the learning context *Position_Posture*, previously depicted in Figure 3.13. For simplicity, just postures of interest for the example are listed in the state representation.

The hypothetic case consists in a person staying in *Standing* posture during 69 frames (state $S_{1.1}$), then the person passes to *Crouching* posture (event $E_{1.1 \to 1.2}$) for the next 43 frames (state $S_{1.2}$), and next the person returns to the *Standing* posture (event $E_{1.2 \to 1.1}$) for the following 145 frames (state $S_{1.1}$). All these posture changes have occurred approximately at the same position in the plane xy (state S_1). Finally, the *Standing* person walks to another position (events $E_{1 \to 2}$ and $E_{1.1 \to 2}$) by 122 frames (state S_2).

The structure of Figure 6.3 allows to appreciate how the state concepts are more specific

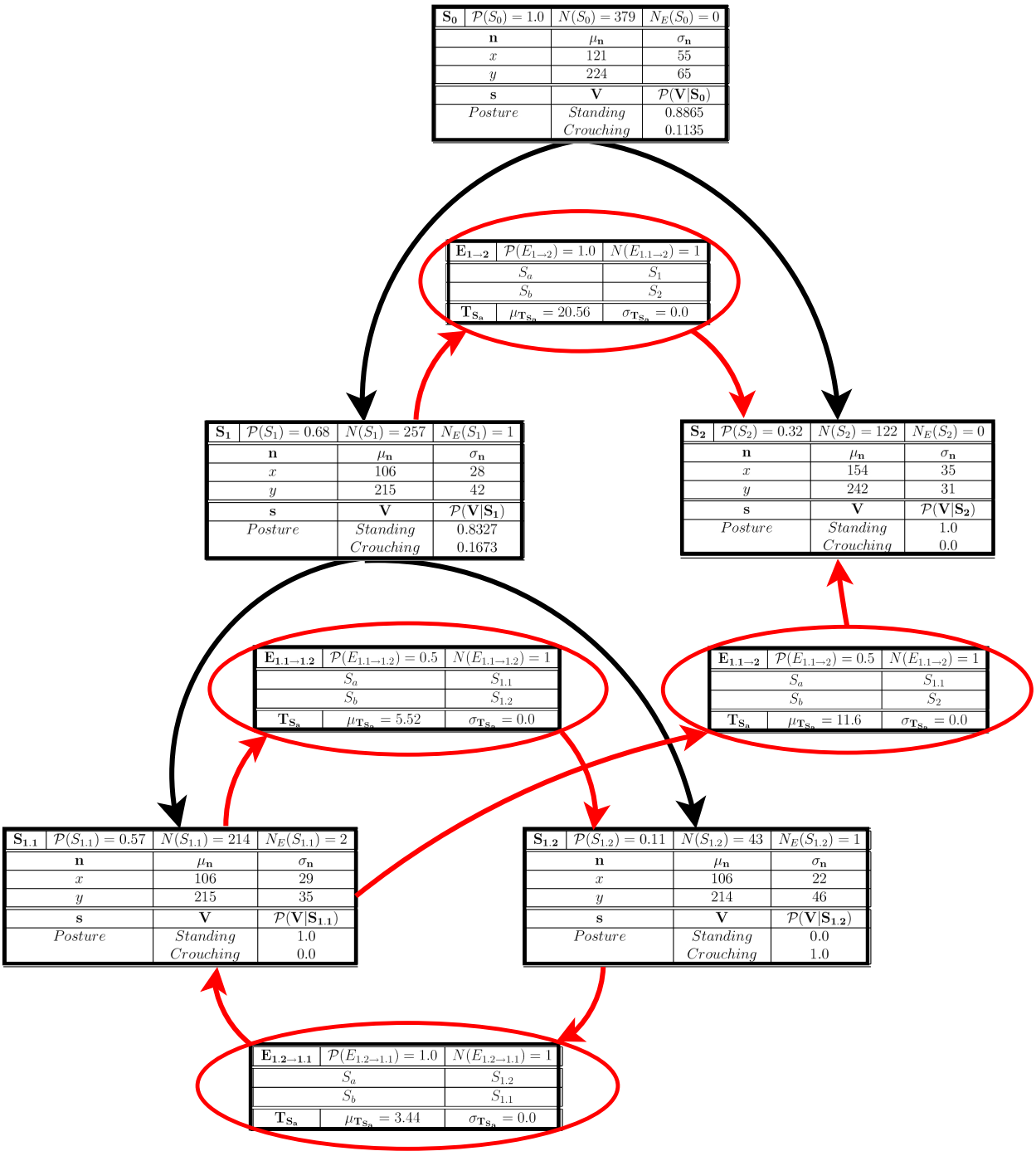


Figure 6.3: Extended example of the hierarchical state and event structure utilised in the proposed event learning approach. The structure represents learnt states and events considering a *Position_Posture* learning context (see Figure 3.13). Black rectangles represent states, while red ovals represent events.

while descending the hierarchical tree. For instance, state S_1 shows posture probabilities higher than zero for both analysed postures, while states $S_{1.1}$ and $S_{1.2}$ specify only one posture, with probability equal to one. Also, state S_0 shows a higher generality degree for the position coordinates x and y, with a higher standard deviation for both dimensions, compared to states S_1 and S_2.

Note that only three events happen in the hypothetical case, so information about state time in each event just represents the occurrence of one event, by a standard deviation equal to zero. Also notice that state S_1 ignores the events happening between its siblings, so event $E_{1\rightarrow 2}$ accounting for the event change to state S_2 as a higher level of abstraction.

Figure 6.4 depicts the states represented in Figure 6.3 in the plane xy. This graphical

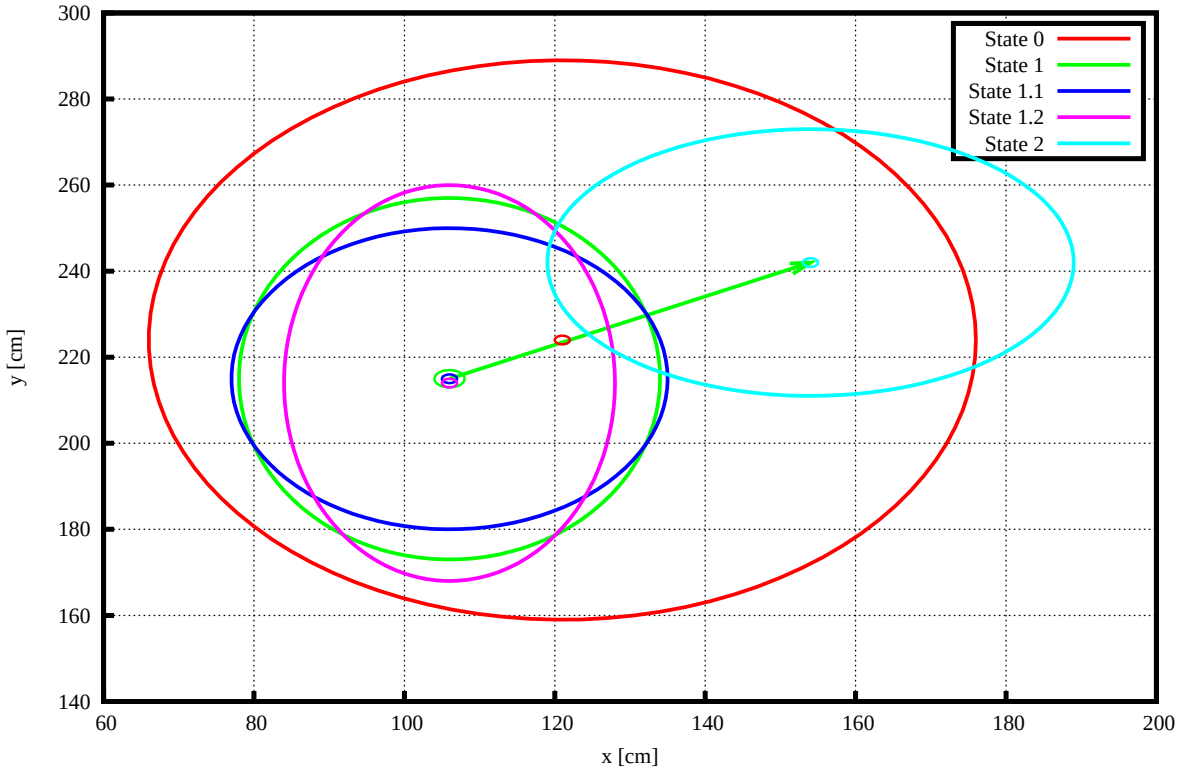


Figure 6.4: Graphical representation of the position information in the xy plane, for the extended example of the hierarchical state and event structure presented in Figure 6.3. The little ovals represent the mean value for the position of each state, while the large ovals represent the standard deviations of the position dimensions. The green arrow represents the Event $E_{1\rightarrow 2}$.

representation shows the separation between state concepts S_1 and S_2 in terms of position, and how well states S_0 and S_1 generalise their children. It is also interesting to notice the high similarity between states $S_{1.1}$ and $S_{1.2}$, as their difference lies in the posture

dimension.

6.1.2 Event Learning Contexts

In order to guide the event learning approach through the extraction of the interesting events according to the application, the user can define several *event learning contexts*.

Definition 6.2 ***Event Learning Context:*** *An event learning context corresponds to a description of the learning scope for a given event learning task. It is defined as a set of mobile object attributes to be learnt for a set of object classes. The set of mobile object attributes can correspond to a mixture of numerical and symbolic attributes. The definition of an* event learning context *LC follows the structure presented below:*

Learning Context LC {
Involved Objects: $Any \mid Any3D \mid \{O_1, O_2, \ldots, O_N\}$
Attributes:
Numerical n_1 : A_{n_1}
Numerical n_2 : A_{n_2}
$\vdots$
Numerical n_M : A_{n_M}
Symbolic s_1 : $\{V_{s_1}^{(1)}, V_{s_1}^{(2)}, \ldots, , V_{s_1}^{(L_1)}\}$
Symbolic s_2 : $\{V_{s_2}^{(1)}, V_{s_2}^{(2)}, \ldots, , V_{s_2}^{(L_2)}\}$
$\vdots$
Symbolic s_P : $\{V_{s_P}^{(1)}, V_{s_P}^{(2)}, \ldots, , V_{s_P}^{(L_P)}\}$
}

The **Involved Objects** statement defines the object classes to be analysed in the learning context LC. This definition can be Any2D if every object type is considered, even the unknown type (no 3D information available), Any3D if every object type different from unknown is considered, or a list of object classes including the objects of interest for this learning context. The Any2D option is used only when all the considered mobile object attributes are independent from the 3D referential of the scene. An example of definition of a $Trajectory$ learning context is defined in Figure 6.5, where the learnt objects can be *any* of the available classes, except the unknown class.

The **Attributes** statement defines the mobile object attributes to be considered in the learning process. For each numerical attribute n_i, with $i \in \{1, \ldots, M\}$, it is necessary to associate a normalisation value A_{n_i}, which represents the lower bound for the numerical attribute change to be considered as meaningful. In other words, the difference between the mean value for a numerical attribute n and the value of the attribute for a new instance will be considered as significant and noticeable when this difference is higher than the acuity A_n.

The normalisation value A_{n_i} corresponds to the concept of *acuity*, utilised by [Gennari

Learning Context *Trajectory* {
Involved Objects: Any3D
Attributes:
Numerical x : 2 [m]
Numerical y : 2 [m]
Numerical Vx : 10 [km/h]
Numerical Vy : 10 [km/h]
}

Figure 6.5: Definition of a trajectory learning context in a parking lot environment. For this context, the user can be interested in learning the events associated to the object position (x, y), together with the velocity (V_x, V_y) for any type of object in a parking lot environment (e.g. persons and vehicles).

et al. 1989], [Gennari et al. 1990], described in Section 2.4.5 as a system parameter that specifies the minimum value for attributes σ in the CLASSIT algorithm for incremental concept formation. In psychophysics, the *acuity* corresponds to the notion of a *just noticeable difference*, the lower limit on the human perception ability. This concept is used for the same purpose in the proposed event learning approach, but the main difference with its utilisation in CLASSIT is that the *acuity* was used as a single parameter, while A_{n_i} acuity values are defined for each attribute to be learnt for a given context. This improvement allows to represent the different normalisation scales and units associated to different attributes, as also representing the interest of users for different applications. For instance, a trajectory position attribute x could have an acuity of 50 *centimetres* for an application with a camera in an office environment, while for the same attribute, the acuity could be *two metres* for a parking lot application with a camera far from the objects, where the user is not interested in little details on position change.

For the symbolic attributes, it is necessary to list the values of interest associated to each of these attributes. As enunciated in Definition 6.2, both numerical and symbolic attributes can simultaneously be part of the same event learning context. This situation is represented with an example in Figure 6.6 for a position-posture context. This context mixes numerical position attribute information, with symbolic posture attribute information.

Each *event learning context* defines an autonomous *event learning process*, giving to the approach the sufficient flexibility to learn events of different nature utilising the same mobile objects extracted from the video sequence. In consequence, each event learning process is defined by its associated event learning context and the event hierarchy the learning process learns. Formally:

Learning Context *Position_Posture* {
Involved Objects: Person
Attributes:
Numerical x : 50 [cm]
Numerical y : 50 [cm]
Symbolic Posture : { Standing,
Crouching,
Sitting,
Lying }
}

Figure 6.6: Definition of a Position-Posture learning context for people in an office environment.

Learning Process LP_1 {
Hierarchy of States and Events: H_1
Learning Context: LC_1
}

It is then necessary to define how a mobile object will be able to feed different event learning processes simultaneously. For each mobile object, it is necessary to obtain a contextualisation of its attributes according to each learning context LC in which the object can be involved, together with information about the current state and latest event of the mobile object in a hierarchy H. This results in a structure as described below:

ContextualisedObject CO {
Learning Context: LC_1
Attributes:
$LC_1.v_1 = LC_1.V_1 \leftarrow LC_1.R_1$
$\vdots$
$LC_1.v_M = LC_1.V_M \leftarrow LC_1.R_M$
Hierarchy: H_{LC_1}
Level: L_1
Previous State: $H_{LC_1}.S_a^{(L_1)}$ | Unknown
Current State: $H_{LC_1}.S_b^{(L_1)}$ | Unknown
Time in Current State: $T_{H_{LC_1}.S_b^{(L_1)}}$
Last Event: $H_{LC_1}.E_{a\to b}^{(L_1)}$ | Unknown
$\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$
Level: L_Q
Previous State: $H_{LC_1}.S_a^{(L_Q)}$ | Unknown
Current State: $H_{LC_1}.S_b^{(L_Q)}$ | Unknown
Time in Current State: $T_{H_{LC_1}.S_b^{(L_Q)}}$

Last Event: $H_{LC_1}.E_{a\to b}^{(L_Q)}$ | Unknown

$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$

Learning Context: LC_K

Attributes:

$LC_K.v_1 = LC_K.V_1 \leftarrow LC_K.R_1$

$\vdots$

$LC_K.v_M = LC_K.V_M \leftarrow LC_K.R_M$

Level: L_1

Previous State: $H_{LC_K}.S_a^{(L_1)}$ | Unknown

Current State: $H_{LC_K}.S_b^{(L_1)}$ | Unknown

Time in Current State: $T_{H_{LC_K}.S_a^{(L_1)}}$

Last Event: $H_{LC_K}.E_{a\to b}^{(L_1)}$ | Unknown

$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$

Level: L_Q

Previous State: $H_{LC_K}.S_a^{(L_Q)}$ | Unknown

Current State: $H_{LC_K}.S_b^{(L_Q)}$ | Unknown

Time in Current State: $T_{H_{LC_K}.S_a^{(L_Q)}}$

Last Event: $H_{LC_K}.E_{a\to b}^{(L_Q)}$ | Unknown

}

Hence, in order to contextualise a tracked object O, for each learning context LC_j in which the object O is involved, with $j \in \{1, \ldots, K\}$, the contextualised object CO defines the attribute-value-measure triplets $(v_i; V_i; R_i)$, with $i \in \{1, \ldots, M\}$, where R_i corresponds to the reliability measure associated to the obtained value V_i for the attribute v_i. This triplet is defined regardless if the type of the concerned attribute is numerical or symbolic. With the object already contextualised, now it is possible to feed the event learning processes properly.

Also, the contextualised object CO must store information about the current states and events, in order to detect the occurrence of a new event and to be able to generate its representation properly, for each hierarchy H_{LC} learnt for a learning context LC. This information consists of the previous state $S_a^{(L_q)}$ where the tracked object was, the current state $S_b^{(L_q)}$ where the tracked object is, the last occurred event $E_{a\to b}^{(L_q)}$ for the object, and the time $T_{S_b^{(L_q)}}$ staying in the current state, for each level L_q of the state and event hierarchy, with $q \in \{1, \ldots, Q\}$. Notice that the level $q = 0$ is not considered as it will only contain the root node for the hierarchy representing the learning context LC (see previous Section 6.1.1).

Now, with all these elements and their interactions properly described, details on the event learning process can be presented in next Section 6.2.

6.2 MILES: Method for Incremental Learning of Events and States

As described in Section 6.1.2, each *learning context* defines a different learning process for independently generating a hierarchy of states and events, as the one described in Section 6.1.1. This Section is dedicated to the detailed description of this event learning process.

The proposed event learning process is based on models of incremental concept formation ([Gennari et al. 1990], [Carbonell 1990]), which have been discussed in Section 2.4. The models of incremental concept formation allow to incrementally build a concept hierarchy based on incomplete or uncertain data, by updating the hierarchical concept structure with the arrival of each new data instance. These models also allow the classification of a new instance, based on the inferred concepts from previously processed data.

In the context of the proposed event learning process, a concept corresponds to a state and the learnt data correspond to the visual attributes of mobile objects present in the video scene. More specifically, these data correspond to the contextualised object CO, defined in Section 6.1.2.

As every incremental concept formation model, the proposed incremental event learning approach needed a name. This approach has been called *MILES*, acronym standing for **M**ethod for **I**ncremental **L**earning of **E**vents and **S**tates. MILES state hierarchy construction is mostly based on COBWEB [Fisher 1987] algorithm (see Section 2.4.3), but also considering ideas from other existing incremental concept formation approaches. From CLASSIT [Gennari et al. 1990] algorithm (see Section 2.4.5) the concepts of *acuity* and *cutoff* are considered, but in a different way, as detailed in Section 6.2.2.

As defined in Section 2.4.3, to evaluate the concept nodes, the latest incremental concept formation models use a quality measure for categories (or concepts) called *category utility*, which favours clusterings that maximise the potential for inferring information. The objective of the category utility is to measure how well the instances are represented by a given category. In this thesis, a category is a state.

For MILES, a measure similar to the category utility function from COBWEB/3 [McKusick and Thompson 1990] algorithm has been considered, which is based in Equations (2.7), (2.8), and (2.9) (Section 2.4.6). These new equations correspond to Equations (6.1), (6.2), and (6.3), and are now defined considering a state concept S_k in a learning context LC. For the set of numerical attributes, the category utility CU_k, for a given state concept S_k, is defined as:

$$CU_k(numerical) = \frac{\mathcal{P}(S_k)\sum_{i=1}^{I}\left(\frac{A_{n_i}}{\sigma_{n_i}^{(k)}} - \frac{A_{n_i}}{\sigma_{n_i}^{(p)}}\right)}{2\cdot I\cdot\sqrt{\pi}}, \tag{6.1}$$

where $\sigma_{n_i}^{(k)}$ is the standard deviation for the numerical attribute n_i, with $i \in \{1, 2, .., I\}$, in the state concept S_k, and $\sigma_{n_i}^{(p)}$ is the standard deviation for the numerical attribute n_i in the parent or root node S_p, as defined in Section 6.1.1. The value A_{n_i} corresponds to the *acuity* for the attribute n_i.

Note that the incorporation of the acuity term A_{n_i} establishes a difference with the preceding versions of numerical category utility. The idea of utilising the acuity value is to balance the contribution of numerical and symbolic attributes to the category utility, giving to the numerical attributes the possibility to have a probability of one if the standard deviation corresponds to the acuity value. This assumption is reasonable in the sense that the acuity value defines when a change in a numerical attribute is considered as not significant. The obtained attribute contribution value always belongs to the interval $[0, 1]$, as the acuity A_{n_i} is the lower bound for the standard deviation $\sigma_{n_i}^{(k)}$. Also, the incorporation of the acuity is useful to normalise the contributions of numerical attributes representing different metric units (e.g. position and velocity) and scales (e.g. a position attribute in metres and a distance attribute in centimetres).

For the set of symbolic features, the category utility CU_k, for a given state concept S_k, is defined as:

$$CU_k(symbolic) = \frac{\mathcal{P}(S_k)\sum_{l=1}^{L}\sum_{j=1}^{J_L}\left(\mathcal{P}(s_l = V_{s_l}^{(j)}|S_k)^2 - \mathcal{P}(s_l = V_{s_l}^{(j)}|S_p)^2\right)}{L}, \tag{6.2}$$

where $\mathcal{P}(s_l = V_{s_l}^{(j)}|S_k)$ is the conditional probability that the symbolic attribute s_l has a value $V_{s_i}^{(j)}$ in the state concept S_k, with $l \in \{1, 2, .., L\}$ and $j \in \{1, 2, .., J_L\}$, while $\mathcal{P}(s_l = V_{s_l}^{(j)}|S_p)$ is the conditional probability that the symbolic attribute s_i has a value $V_{s_i}^{(j)}$, in the parent or root node S_p, as defined in Section 6.1.1.

Then, for a set of mixed symbolic and numerical attributes, the overall category utility CU_k, given a state concept S_k, is the sum of the contributions of both sets of features:

$$CU_k = CU_k(symbolic) + CU_k(numerical). \tag{6.3}$$

Finally, the category utility CU for a class partition of K classes is defined as:

$$CU = \sum_{k=1}^{K} \frac{CU_k}{K} \tag{6.4}$$

For a given learning context, MILES sequentially processes all the contextualised object instances at the current frame. MILES initialises its hierarchy to a single state concept, setting the values of the state concept attributes as the values of the first processed instance. Upon encountering a second instance, MILES averages its values into those of the initial state concept and creates two children, one based on the initial state and another based on the instance.

Then, at each state concept, MILES retrieves all children and considers classifying and placing the new instance in each of these states. Based on the category utility presented in Equation (6.3), a decision is made for the incorporation of the instance. This decision can be to incorporate the instance to an existing state concept, to generate a new state from the instance, to merge the two states best fitting the instance (merge operator in Section 6.2.3), to eliminate a state concept and replace it by its children (split operator in Section 6.2.3).

When the decision is made and the state concept is created or updated, MILES verifies whether the tracked object has changed its state for this level in the hierarchy. If this is the case, an event occurs and an event entity as described in Section 6.1.1 is updated if the entity already exists, or created if not.

If the currently chosen state concept has siblings, the learning process stops if the current state concept passes a *cutoff* criteria.

Definition 6.3 ***Cutoff:*** *The* cutoff *is a criteria utilised for stopping the creation of children by a learning process. It can be defined as:*

$$cutoff = \begin{cases} true & if \quad \{\mu_{n_i}^{(S_k)} - V_{n_i} \leq A_{v_i} | \forall i \in \{1,..,I\}\} \ \wedge \ \{\mathcal{P}(V_{s_j} | {s_j}^{(S_k)}) = 1 | \forall j \in \{1,..,J\}\} \\ false & else \end{cases} , \tag{6.5}$$

where V_{n_i} is the value of the i-th numerical attribute of the processed instance, and V_{s_j} is the value of the j-th symbolic attribute of the processed instance. The value $\mu_{n_i}^{(S_k)}$ corresponds to the mean value of the numerical attribute n_i for the state S_k.

This equation means that the learning process for the instance will stop at state S_k if no meaningful difference exists between a numerical attribute value of the instance and the mean value of the attribute for the state S_k (based on the acuity *for the attribute), or if every symbolic attribute value in he instance is totally represented in the state S_k (probability equal to one for the attribute value). This means that the learning process will stop if no noticeable difference between the attribute values is found.*

This different way of considering the *cutoff* and *acuity* concepts with respect to the utilisation proposed in CLASSIT algorithm (see Section 2.4.5) constitutes one of the

contributions of the approach to the incremental concept formation models domain.

The following Sections describe different details of the learning process of MILES. Section 6.2.1 describes the incremental updating process for information contained in the state and event concepts of a hierarchy, given the arrival of a new object instance, and how the reliability measures can control the incorporation of new information according to their quality. Then, in Section 6.2.2 the learning process is described in detail. Finally, Section 6.2.3 describes how merge and split operators are applied for modifying the hierarchy of state and event concept hierarchy.

6.2.1 Reliable Information Incorporation

Upon the arrival of a new state instance represented by a contextualised object CO, the attribute information of the instance must be used to update the state and event concept information. According to the type of attribute the information updating process differs.

For the case of a numerical attribute n, the information about the mean value μ_n and the standard deviation σ_n must be updated. The proposed updating functions are incremental in order to improve the processing time performance of the approach. The incremental updating function for the mean value μ_n of a numerical attribute n is presented in Equation (6.6).

$$\mu_n(i) = \frac{V_n \cdot R_n + \mu_n(i-1) \cdot Sum_n(i-1)}{Sum_n(i)}, \tag{6.6}$$

with

$$Sum_n(i) = R_n + Sum_n(i-1), \tag{6.7}$$

where V_n is the value for the new instance for the attribute n and R_n corresponds to its reliability. Hence, the reliability R_n weights the contribution of the new attribute value V_n to the mean value for n. Sum_n function corresponds to the accumulation of reliability values R_n for the numerical attribute n.

The incremental updating function for the standard deviation σ_n of a numerical attribute n is presented in Equation (6.8).

$$\sigma_n(i) = \sqrt{\frac{Sum_n(i-1)}{Sum_n(i)} \cdot \left(\sigma_n(i-1)^2 + \frac{R_n \cdot (V_n - \mu_n(i-1))^2}{Sum_n(i)} \right)}. \tag{6.8}$$

In the case that a new state concept is generated from the attribute information of the instance, the initial values taken for Equations (6.6), (6.7), and (6.8) with $i = 0$ correspond to $\mu_n(0) = V_n$, $Sum_n(0) = R_n$, and $\sigma_n(0) = A_n$, where A_n is the *acuity* for the attribute n, as defined in Section 6.1.2.

In case that, after updating the standard deviation Equation (6.8), the value of $\sigma_n(i)$ is lower than the *acuity* A_n, $\sigma_n(i)$ becomes equal to A_n. This way, the acuity value

establishes a lower bound for the standard deviation of an attribute, avoiding the possibility of zero division in the category utility function at the Equation (6.1).

For symbolic attributes it is necessary to update the conditional probability $\mathcal{P}(s = V_s^{(j)}|S)$ of each possible value $V_s^{(j)}$ for a symbolic attribute s, given the state concept S. For this purpose, reliability measures R_s are utilised in order to weight the quality of new incoming information, as presented in Equations (6.9), (6.10), and (6.11).

$$\mathcal{P}(s = V_s^{(j)}|S)[i] = \begin{cases} \dfrac{Sum_{V_s}^{(j)}(i)}{Sum_s(i)} & if \quad V_s = V_s^{(j)} \\ \dfrac{Sum_{V_s}^{(j)}(i-1)}{Sum_s(i)} & else \end{cases} \tag{6.9}$$

with

$$Sum_{V_s}^{(j)}(i) = R_s + Sum_{V_s}^{(j)}(i-1), \tag{6.10}$$

and

$$Sum_s(i) = R_s + Sum_s(i-1), \tag{6.11}$$

where V_s is the value for the new instance for the symbolic attribute s and R_s corresponds to its reliability. $V_s^{(j)}$ is the j-th possible value for the symbolic attribute s, with $j \in \{1, \ldots, L_s\}$ (L_s is the number of possible values for s). The functions $Sum_{V_s}^{(j)}(i)$ correspond to the accumulated reliability for each s attribute value V_s, while the function $Sum_s(i)$ corresponds the overall accumulated reliability for the attribute s. This way, the probability $\mathcal{P}(s = V_s^{(j)}|S)$ corresponds to the ratio between the accumulated reliability for the attribute value $V_S^{(j)}$, over the overall accumulated reliability for the attribute s. Notice that only the accumulated reliability for the attribute value corresponding to the value of the current instance is updated.

The right choice of the reliability functions determining the reliability associated to the attributes of a contextualised object can be of great help on increasing the robustness of MILES. For the attributes utilised by the dynamics models presented in Section 5.2.2, reliability measures have been already proposed that can be directly used in update Equations (6.6) and (6.7) for numerical attributes, as can, for instance, be appreciated at the temporal coherence reliability Equations (5.5) and (5.13).

For symbolic attributes and other numerical attributes not updated by the tracking dynamics model, the reliability measures must be defined. They can be conceived in multiple forms. For instance, a combination of the already defined reliability measures, the object probability measure of Equation (5.15) (defined in Section 5.2.2) as a general attribute reliability measure, or a combination of general and specific measures could be utilised for this purpose.

6.2.2 Events Tree Generation Algorithm

In this Section, the proposed incremental event learning algorithm MILES is described in detail. A pseudo-code representation is displayed below.

```
function MILES (P, CO, O) returns RE and H
Input
    P:       Learning Processes List.
    CO:      Contextualised objects list.
    O:       Tracked objects list.
Output
    RE:      Recognised states and events.
    H:       List of updated hierarchies of states and events.
Begin
    If O is new_mobile then
           co = initialiseContextualisedObject ( O, P );
           insertContextualisedObject ( co, CO );
    Else
           co = getContextualisedObject ( CO, O );
    End If


    For Each  p  in  P  do
           h = getAssociatedHierarchy(p);
           I = getStateInstance ( p, co );
           L = updateStates( h, I );
           If  firstFrame( co )  then
                   updateCurrentStates( co, L );
           Else
                   oldL = getCurrentStates( co );
                   For Each  l  in  oldL  do
                            If  stateChanges( l, L )  then
                                    updateEvents( co, l, L );
                            End If
                   End For
                   updateCurrentStates( co, L );
           End If
    End For
    H = empty_set;
    RE = extractRecognisedStatesAndEvents(CO);
    For Each  p  in  P  do
           h = getAssociatedHierarchy(p);
           H = insertHierarchy(h, H);
    End For
    return RE and H;
```

```
End
```

MILES algorithm utilises all tracked objects for updating the hierarchies of state and event concepts. The algorithm first initialises the contextualised object *co* associated to a new object O with the function *initialiseContextualisedObject*. This function uses the learning contexts information associated to each of learning process in the list P, as described in Section 6.1.2, to determine for which of these learning processes the object O is valid to extract the proper information accordingly.

Hence, for each learning process, this function checks if the type of the tracked object O corresponds to the allowed object types for the learning context associated to the learning process. If the type is valid, the triplets $(v; V; R)$ are extracted from the object O attribute information and used to initialise the contextualised object *co*, with V corresponding to the value of attribute v, and R to its reliability.

After, the initialised contextualised object *co* is inserted to the list of contextualised objects CO using function *insertContextualisedObject*. If the tracked object O is not new, the existing contextualised object *co* is recovered with function *getContextualisedObject*.

Then, for each of the learning processes in list P, the currently learnt hierarchy of states and events h associated to the current learning process is extracted with function *getAssociatedHierarchy*. Also, the object instance I of the contextualised object *co* containing the attribute information necessary for the current learning process, is extracted with function *getStateInstance*. Then h and I are used to update the states of the hierarchy, using the function *updateStates*. This function is very important as it corresponds to the incremental concept formation model utilised for learning the hierarchy of states, and it is described in detail at Section 6.2.2.1. The *updateStates* function returns a list composed by the current state concepts at each level of the hierarchy.

Finally, the events associated to a state transition of the tracked object O are updated. If the currently processed object O is a new object, just the current states for each level in the hierarchy are updated with function *updateCurrentStates*. If the tracked object O is not new, the states stored from the previous frame for the tracked object O are extracted from its associated contextualised object *co* with function *getCurrentStates*. Then, for these previously stored state concepts for each level of the hierarchy, the occurrence of a state transition is verified with the function *stateChanges*. This verification is made by checking if the analysed state is present in the list of updated states returned by function *updateStates*.

If a change of states is detected, the function *updateEvents* updates the events information according to the detected change of state. The occurrence of a state transition updates all the events representing the combinations between the analysed state concept from the stored list, where the possible combinations are:

- All the states of a lower level in the new list, if the state at its same level in the new

list is different than the analysed state.

- The state at its same level in the new list if it is different than the analysed state.
- All the states at a higher level in the new list which does not have a *kinship relation* with the analysed state

Examples of these state combinations can be found in Figure 6.7.

Definition 6.4 *A **kinship relation** between two states S_m and S_n in the hierarchy exists if S_m is (directly or indirectly) the ascendant or one of the descendants of the state S_n in the hierarchy. This means that the one state is related to the other as parent, or son, or grand-parent, or grand-son, and so on.*

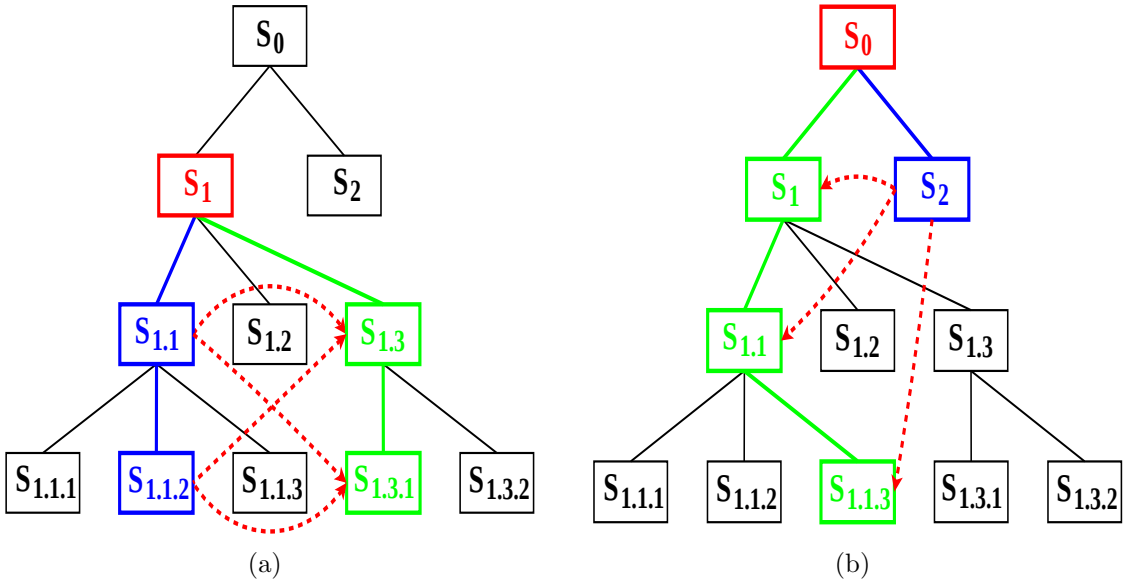


Figure 6.7: Examples of list comparisons for determining the events to update. Blue elements represent the previously stored states for a tracked object. Green elements represent the updated states obtained with the function *updateStates*. The red box represents the state concept which is common to both lists. The dashed red lines represent the events to update for two different cases. Figure (a) shows the previous state $S_{1.1}$ generating events at the same level and a lower level in the hierarchy, and the state concept $S_{1.1.2}$ generating events at the same and higher levels in the state and event concepts hierarchy. Figure (b) shows the previous state S_2 generating events at the same level and at lower levels in the hierarchy.

If an event E corresponds to a first detected event, a new event representation is created and associated to the generating state S_a and the arriving state S_b. The mean time staying at state S_a, $\mu_{T_{S_a}}$ is initialised with the accumulated time in current state $T_{S_a^{(Lq)}}$, with q corresponding to the time staying in the starting state S_a at level q. The standard deviation for the time $\sigma_{T_{S_a}}$ is initialised to 0.0.

If an event E corresponds to an existing event representation, the $\mu_{T_{S_a}}$ and $\sigma_{T_{S_a}}$ values are updated using the Equations (6.12) and (6.13), respectively.

$$\mu_{T_{S_a}}(i) = \frac{T_{S_a} + \mu_{T_{S_a}}(i-1) \cdot N_E)}{N_E + 1}, \tag{6.12}$$

where N_E is the number of times the event E has been detected.

$$\sigma_{T_{S_a}}(i) = \sqrt{\frac{N_E}{N_E + 1} \cdot \left(\sigma_{T_{S_a}}(i-1)^2 + \frac{(T_{S_a} - \mu_{T_{S_a}}(i-1))^2}{N_E + 1} \right)}. \tag{6.13}$$

Then, the updated list of current states at different levels in the hierarchy is utilised to update the current states information of the contextualised object *co*, utilising the function *updateCurrentStates*.

Finally, the list of updated hierarchies H is built with the updated hierarchies, and the recognised states and events are collected from the information contained in the contextualised object list CO. Then, the currently recognised states and events for each mobile object are returned, and also the updated hierarchies of states and events associated to each learning process.

6.2.2.1 States Updating Function

The states updating function requires special attention as it corresponds to the incremental concept learning component of the approach. This function works in a similar way compared with COBWEB algorithm [Fisher 1987]. A pseudo-code representation of this function is displayed below.

```
function updateStates ( h,  I ) returns L
Input
    h:    Current state in the event and state concept hierarchy.
    I:    Contextualised object instance.
Output
    L:    List of current states for instance I.
Begin
    If  emptyTree ( h ) then
        insertRoot ( h, I );
    Else If isTerminalState ( getRootOfHierarchy(h) ) then
        If  cutoffTestPassed ( getRootOfHierarchy(h), I ) then
            createNewTerminals ( h, I );
        End If
        incorporateState ( getRootOfHierarchy(h), I );
    Else
        If lastOperation != Split then
            incorporateState ( getRootOfHierarchy(h), I );
```

```
        End If
        P = highestScoreState ( h );
        W = categoryUtilityScore ( P );
        Q = newStateConcept ( h, I );
        X = categoryUtilityScore ( Q );
        If numberOfChildren( getRootOfHierarchy(h) ) > 2 then
            R = secondScoreState ( getRootOfHierarchy(h) );
            Y = mergeCategoryUtilityScore ( P, R );
        Else
            Y = 0.0;
        End If
        If numberOfChildren( P ) > 0 then
            Z = splittingScore ( P );
        End If
        If W is bestScore then
            updateStates ( getSubTree(h, P), I );
        Else If X is bestScore then
            insertChild ( Q, h );
        Else If Y is bestScore then
            O = mergeStates ( P, R, h );
            updateStates ( getSubTree(h, O), I );
        Else If Z is bestScore then
            splitStates ( P, h );
            updateStates ( h, I );
        End If
    End If
    insertCurrentState ( getRootOfHierarchy(h), L );
End
```

For the arrival of the first object instance to the states hierarchy (function *emptyTree*), the state updating process first initialises the hierarchy to a single state (function *insertRoot*), setting the values of the state concept attributes as the values of the first processed instance, as described in Section 6.2.1.

Then, for the case that the currently considered state $getRootOfHierarchy(h)$ (where $getRootOfHierarchy$ returns the root state of the analysed tree) in the hierarchy corresponds to a terminal state (function *isTerminalState*), which does not have any children, a *cutoff* test if performed by function *cutoffTestPassed*. This test consists in checking if the new object instance is sufficiently different to the state $getRootOfHierarchy(h)$. The test will be passed if the difference between every numerical attribute n, between the instance I and the state concept $getRootOfHierarchy(h)$, is lower than the *acuity* A_n associated to attribute n, and if each symbolic attribute s has the same value V_s in I and $getRootOfHierarchy(h)$. This way of defining the *cutoff* criterion differs from the one used by COBWEB, which considers a fixed threshold.

If the *cutoff* test is passed, the function *createNewTerminals* generates two children for current state concept $getRootOfHierarchy(h)$, one initialised with the instance information and the other as a copy of $getRootOfHierarchy(h)$. Then, passing or not passing the *cutoff* test, the information of I is incorporated to the current state concept $getRootOfHierarchy(h)$ by function *incorporateState*, which utilises the updating functions described in Section 6.2.1. Then, the process of updating the hierarchy using the object instance I stops when a terminal state $getRootOfHierarchy(h)$ is considered.

If the current state concept $getRootOfHierarchy(h)$ is not a terminal state, meaning that it has children, first the object instance I is immediately incorporated to $getRootOfHierarchy(h)$ with function *incorporateState* (if the last operation in the hierarchy was not a *split*). Then, different possibilities of evolution in the hierarchy for the object instance I must be evaluated among all the children of the current state concept $getRootOfHierarchy(h)$, choosing the alternative with best category utility score (obtained with function *categoryUtilityScore*). This category utility score has been previously defined in Equations (6.1), (6.2), (6.3), and (6.4), at Section 6.2, and defines a measure for evaluating the quality of a given class partition. The different alternatives for the evolution of I in the hierarchy are:

- The incorporation of the instance I to an existing state concept P gives the best category utility score W (function *highestScoreState*). In this case, the function *updateStates* is recursively called, using the state $getSubTree(h, P)$ as current state, where function $getSubTree$ returns the subtree of h considering P as root.

- The generation of a new state concept Q from instance I gives the best category utility score X (function *newStateConcept*). In this case, the function *insertChild* inserts the new state Q as child of the current state concept $getRootOfHierarchy(h)$, and the updating process with instance I stops.

- If the number of children of the current state $getRootOfHierarchy(h)$ is higher than two, a state merge process can be evaluated. The second best state R is determined (function *secondScoreState*), and the category utility score Y of considering a merge between best state P and state R is obtained (function *mergeCategoryUtilityScore*). If the category utility obtained from the merge process gives the best score, the hierarchy is modified by the merge process performed by the function *mergeStates*, and the function *updateStates* is recursively called, using the state $getSubTree(h, O)$, resulting from the merge process, as the current state. This merge process is detailed in Section 6.2.3.

- If the best score state P has children, a state split process can be evaluated. The category utility score Z of considering a split operation of the best state P is obtained (function *splittingScore*). If the category utility obtained from the split process gives the best score, the hierarchy is modified by the split process performed by

the function *splitStates*, and the function *updateStates* is recursively called, using again the state $getRootOfHierarchy(h)$ as the current state. This is why the incorporation of I is not performed if a split operation have been performed before, as the incorporation of I has been already made to $getRootOfHierarchy(h)$ at the previous step. This split process is also detailed in Section 6.2.3.

Finally, each current state $getRootOfHierarchy(h)$ is stored in the list of current state concepts L, by the function *insertCurrentState*.

6.2.3 Operators for the State and Event Concepts Hierarchy

As described in previous Section 6.2.2.1, three operations can modify the structure of the state and event concepts hierarchy. The first one is the creation of a new state concept from an object instance, which just consist in adding a new state concept initialised with attribute values of the object instance. The other two operations are more complex as they perform more drastic modifications to the hierarchy. They correspond to the *merge* and *split* operator which are detailed in the following Sections.

6.2.3.1 Merge Operator

The merge operator consists in merging two state concepts S_p and S_q into one state S_M, while S_p and S_q become the children of S_M, and the parent of S_p and S_q becomes the parent S_M, as depicted in Figure 6.8.

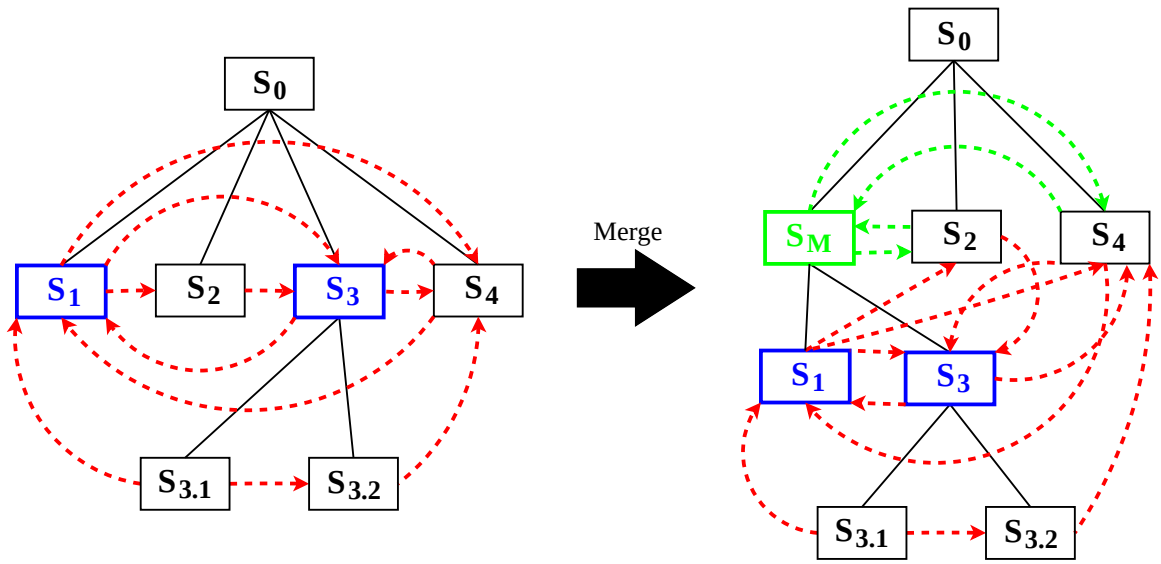


Figure 6.8: Merging states and events in MILES algorithm. Blue boxes represent the states to be merged, and the green box represents the resulting merged state. Red dashed lines represent events, while the green dashed lines are the new events appearing from the merging process.

In order to generate the state S_M several considerations must be made:

- The number $N(S_M)$ of instances represented in S_M, will correspond to the summation of the number of instances represented by S_p and S_q. The probability $\mathcal{P}(S_M)$ for the new state S_M will then be the number of instances $N(S_M)$, over the number of represented instances by the parent of S_M.

- The number $N_E(S_M)$ of event occurrences starting from S_M will correspond to the summation of the number of event occurrences $N(E)$ of all event E having as a starting state S_a the state S_p, or S_q, and as an ending state S_b a state not having a *kinship relation* with S_M (see Definition 6.4).

- Each numerical attribute n_M for S_M can be updated using the Equations (6.14), and (6.15) for mean and standard deviation of n_M, respectively.

$$\mu_{n_M} = \frac{Sum_{n_p} \cdot \mu_{n_p} + Sum_{n_q} \cdot \mu_{n_q}}{Sum_{n_p} + Sum_{n_q}}, \tag{6.14}$$

$$\sigma_{n_M} = \sqrt{\frac{Sum_{n_p} \cdot ((\mu_{n_M} - \mu_{n_p})^2 + \sigma_{n_p}^2) + Sum_{n_q} \cdot ((\mu_{n_M} - \mu_{n_q})^2 + \sigma_{n_q}^2)}{Sum_{n_p} + Sum_{n_q}}}, \tag{6.15}$$

  where Sum_{n_p} and Sum_{n_q} correspond to the reliability values accumulation of attribute n for merging states S_p and S_q, respectively, as previously defined in Equation (6.7). Then, the values for μ_{n_M} and $\sigma_{n_M}^2$ correspond to the mean between μ_n and σ_n^2 for states S_p and S_q, weighted by the reliability values accumulation Sum_n for numerical attribute n. The value of σ_{n_M} is also adjusted for considering the drift between the new mean μ_{n_M}, and the mean values μ_{n_p} and μ_{n_q}.

- Each symbolic attribute s_M for S_M can be updated using the Equation (6.16), for the conditional probability $\mathcal{P}(s_M)^{(j)}$, for the j-th value of the symbolic attribute s_M.

$$\mathcal{P}(s_M = V_{s_M}^{(j)} | S_M)[i] = \frac{Sum_{V_{s_p}}^{(j)} + Sum_{V_{s_q}}^{(j)}}{Sum_{s_p} + Sum_{s_q}}, \tag{6.16}$$

  where $Sum_{V_{s_p}}^{(j)}$ and $Sum_{V_{s_q}}^{(j)}$ correspond to the reliability values accumulation of the j-th value for symbolic attribute s for merging states S_p and S_q, respectively, as previously defined in Equation (6.10). In the same way, Sum_{s_p} and Sum_{s_q} correspond to the overall reliability values accumulation for symbolic attribute s for merging states S_p and S_q, respectively, as previously defined in Equation (6.11). Then, conditional probability Equation (6.16) corresponds to the total accumulated reliability for value $V_{s_M}^{(j)}$ of the symbolic attribute s_M, over the overall total accumulated reliability for the symbolic attribute s_M.

The last task for the merging operator is to represent the events incoming and leaving states S_p and S_q, corresponding to the green dashed lines in Figure 6.8, by generating new events which generalise the transitions as the events incoming and leaving the state S_M. For the incoming events to these states the event merge process is described as follows:

- If a state S_n is the starting state for an event $E_{n \to x}$ arriving to only one state S_x of the merging states S_p and S_q (as event $E_{S_2 \to S_3}$ between states S_2 and S_3 in Figure 6.8), a new event $E_{n \to M}$ between states S_n and S_M must be generated with the same information as event $E_{n \to x}$, except for the arriving state that becomes the state S_M.

- If a state S_n is the starting state for the events $E_{n \to p}$ and $E_{n \to q}$ arriving to both states S_p and S_q (as events $E_{S_4 \to S_1}$ and $E_{S_4 \to S_3}$ in Figure 6.8), a new event $E_{n \to M}$ between states S_n and S_M must be generated as follows:

  - The number of occurrences $N(E_{n \to M})$ will be the sum between the event occurrences $N(E_{n \to p})$ and $N(E_{n \to q})$.
  - The probability of occurrence $\mathcal{P}(E_{n \to M})$ will be the number of occurrences $N(E_{n \to M})$, over the number of event occurrences $N_E(S_n)$ for the starting state S_n.
  - The starting and ending states will be the states S_n and S_M, respectively.
  - The mean value $\mu_{T_{S_n}}^{(E_{n \to M})}$ and the standard deviation $\sigma_{T_{S_n}}^{(E_{n \to M})}$ of the time T_{S_n} staying in the starting state S_n for the new event $E_{n \to M}$ are determined using Equations (6.17), and (6.18), respectively.

$$\mu_{T_{S_n}^{(E_{n \to M})}} = \frac{\mathcal{P}(E_{n \to p}) \cdot \mu_{T_{S_n}^{(E_{n \to p})}} + \mathcal{P}(E_{n \to q}) \cdot \mu_{T_{S_n}^{(E_{n \to q})}}}{\mathcal{P}(E_{n \to p}) + \mathcal{P}(E_{n \to q})}, \tag{6.17}$$

$$\sigma_{T_{S_n}^{(E_{n \to M})}} = \sqrt{\frac{\mathcal{P}(E_{n \to p}) \cdot \sigma^2_{T_{S_n}^{(E_{n \to p})}} + \mathcal{P}(E_{n \to q}) \cdot \sigma_{T_{S_n}^{(E_{n \to q})}}}{\mathcal{P}(E_{n \to p}) + \mathcal{P}(E_{n \to q})}}, \tag{6.18}$$

Merging events leaving the states S_p and S_q is the hardest task for the merging operator, as the staying time at the new starting state S_m must be represented based on the information provided by the events starting from its children states S_p and S_q. The problem is that the time of the starting event S_M can not be absolutely certain, because the children states S_p and S_q can perform several state transitions between them, before performing a state transition to a state which does not have a *kinship relation* with S_M (see Definition 6.4). This problem is also depicted in Figure 6.8, where events exist in both directions for merging states S_1 and S_3, generating a loop between the states.

Taking this problem into consideration and considering an arriving state S_n, for the events leaving the states S_p and S_q, the event merge process is described as follows:

- The number of occurrences $N(E_{M \to n})$ will be the sum between the event occurrences $N(E_{p \to n})$ and $N(E_{q \to n})$.

- The probability of occurrence $\mathcal{P}(E_{M \to n})$ will be the number of occurrences $N(E_{M \to n})$, over the number of event occurrences $N_E(S_M)$ for the starting state S_M.

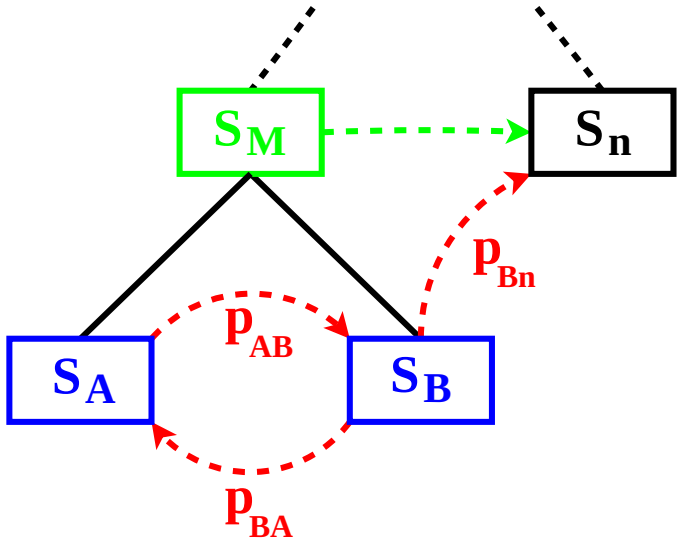


Figure 6.9: Simplified scheme of the problem of estimation of the parameters for the time of permanence in the starting state of an event $T_{S_M}^{(E_{M\to n})}$, for an event occurring between the merge result state S_M and a state S_n not having a *kinship relation* with S_M (see Definition 6.4). Red dashed lines represent events, while the green dashed line corresponds to the new event appearing from the merging process. Notice that a loop of events is occurring between the children states S_A and S_B of the state S_M.

- The starting and ending states will be the states S_M and S_n, respectively.

- As previously described, the mean value $\mu_{T_{S_M}}^{(E_{M\to n})}$ and the standard deviation $\sigma_{T_{S_M}}^{(E_{M\to n})}$ of the time T_{S_M} staying in the starting state S_M for the new event $E_{M\to n}$ must consider the different possibilities of time spent in transitions between the children states S_p and S_q before leaving to the state S_n. These inner transitions can be even infinite, when the children states form a loop, as depicted in Figure 6.9.

  Consider that S_B is a child state of S_M with non-zero transition probability $\mathcal{P}(E_{B\to n})$, and state S_A is the other child state of S_M, as in Figure 6.9. For simplicity, also consider the probabilities $\mathcal{P}_{AB} = \mathcal{P}(E_{A\to B})$, $\mathcal{P}_{BA} = \mathcal{P}(E_{B\to A})$, and $\mathcal{P}_{Bn} = \mathcal{P}(E_{B\to n})$, the mean values of staying state time $\mu_{AB} = \mu_{T_{S_A}^{(E_{A\to B})}}$, $\mu_{BA} = \mu_{T_{S_B}^{(E_{B\to A})}}$, and $\mu_{Bn} = \mu_{T_{S_B}^{(E_{B\to n})}}$, and the standard deviations of staying state time $\sigma_{AB} = \sigma_{T_{S_A}^{(E_{A\to B})}}$, $\sigma_{BA} = \sigma_{T_{S_B}^{(E_{B\to A})}}$, and $\sigma_{Bn} = \sigma_{T_{S_B}^{(E_{B\to n})}}$.

  In order to solve this problem, only an approximation to $\mu_{T_{S_M}}^{(E_{M\to n})}$ and $\sigma_{T_{S_M}}^{(E_{M\to n})}$ can be obtained. Hence, these approximations are defined at Equations (6.19), and (6.25), for the approximations $\mu_{\tau_{S_M}^{(E_{M\to n})}}$ and $\sigma_{\tau_{S_M}^{(E_{M\to n})}}$ of the mean value and the

standard deviation, respectively.

$$\mu_{\tau_{S_M}^{(E_{M\to n})}}(S_A, S_B) = \frac{\Lambda_A + \Lambda_B}{\Delta_A + \Delta_B}, \tag{6.19}$$

with

$$\Lambda_A = \mathcal{P}(S_A)\cdot \sum_{i\in\Omega} \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}\cdot (i\cdot \mu_{BA} + (i+1)\cdot \mu_{AB} + \mu_{Bn}), \tag{6.20}$$

$$\Lambda_B = \mathcal{P}(S_B)\cdot \sum_{i\in\Psi} \mathcal{P}_{AB}^{i}\cdot \mathcal{P}_{BA}^{i}\cdot (i\cdot (\mu_{BA} + \mu_{AB}) + \mu_{Bn}), \tag{6.21}$$

$$\Delta_B = \mathcal{P}(S_B)\cdot \sum_{i\in\Psi} \mathcal{P}_{AB}^{i}\cdot \mathcal{P}_{BA}^{i}, \tag{6.22}$$

and

$$\Delta_A = \mathcal{P}(S_A)\cdot \sum_{i\in\Omega} \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}, \tag{6.23}$$

where set $\Psi = \{i \in \mathbb{N} \mid \mathcal{P}_{AB}^{i}\cdot \mathcal{P}_{BA}^{i} \geq \mathcal{P}_{min}\}$ and set $\Omega = \{i \in \mathbb{N} \mid \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i} \geq \mathcal{P}_{min}\}$, with $\mathcal{P}_{min}$ is a pre-defined minimal conditional probability threshold.

The function Λ_A represents the accumulated mean time of different sequences of state transitions between states S_A and S_B, starting from state S_A, until the final transition to the state S_n. Each sequence of state transitions is weighted by the conditional probability $\mathcal{P}(S_A)\cdot \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}$, which represents the probability of starting from state S_A, next to perform i loops between states S_A and S_B, and finally arriving to S_B to perform the transition to the state S_n. The set Ω limits the inclusion of accumulated mean time values to a minimal pre-defined value P_{min} for the afore mentioned conditional probability.

In the same way, the function Λ_B represents the accumulated mean time of different sequences of state transitions between states S_A and S_B, but this time starting from state S_B, until the final transition to the state S_n. Similarly, each sequence of state transitions is weighted by the conditional probability $\mathcal{P}(S_B)\cdot \mathcal{P}_{AB}^{i}\cdot \mathcal{P}_{BA}^{i}$, which represents the probability of starting from state S_B, and next to perform i loops between states S_A and S_B, to finally perform the transition to the state S_n. The set Ψ limits the inclusion of accumulated mean time values to the same minimal P_{min} for this conditional probability.

Functions Δ_A and Δ_B are used in the mean time Equation (6.19) to accumulate the considered conditional probabilities starting from S_A and S_B respectively. These functions are utilised for normalising the weighted sums Λ_A and Λ_B.

Then, in order to obtain the estimation of the mean value $\mu_{T_{S_M}}^{(E_{M\to n})}$ of the time T_{S_M}, the Equation (6.24) can be utilised.

$$\mu_{T_{S_M}}^{(E_{M\to n})} = \frac{\mathcal{P}(E_{B\to n})\cdot \mu_{\tau_{S_M}^{(E_{M\to n})}}(S_A, S_B) + \mathcal{P}(E_{A\to n})\cdot \mu_{\tau_{S_M}^{(E_{M\to n})}}(S_B, S_A)}{\mathcal{P}(E_{B\to n}) + \mathcal{P}(E_{A\to n})}. \tag{6.24}$$

This equation calculates the mean value $\mu_{T_{S_M}}^{(E_{M\to n})}$ as the weighted mean of the estimators $\mu_{\tau_{S_M}^{(E_{M\to n})}}$, considering rather than the child S_A is the state which generates the outgoing event to the state S_n and the child S_B is considered as the other child, or vice-versa. The weights correspond to the probabilities of transition from the children states S_A and S_B, to the state S_n.

Notice that if the event between the states S_A and S_n does not exist ($\mathcal{P}(E_{A\to n}) = 0$), the Equation (6.24) simplifies to $\mu_{T_{S_M}}^{(E_{M\to n})} = \mu_{\tau_{S_M}^{(E_{M\to n})}}(S_A, S_B)$. In the same way, if the event between the states S_B and S_n does not exist ($\mathcal{P}(E_{B\to n}) = 0$), the Equation then simplifies to $\mu_{T_{S_M}}^{(E_{M\to n})} = \mu_{\tau_{S_M}^{(E_{M\to n})}}(S_B, S_A)$.

For the standard deviation the idea is similar, as defined in Equation 6.25.

$$\sigma_{\tau_{S_M}^{(E_{M\to n})}}(S_A, S_B) = \sqrt{\frac{\Gamma_A + \Gamma_B}{\Delta_A + \Delta_B}}, \tag{6.25}$$

with

$$\Gamma_A = \mathcal{P}(S_A)\cdot \sum_{i\in\Omega} \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}\cdot \frac{i^2\cdot \sigma_{BA}^2 + (i+1)^2\cdot \sigma_{AB}^2 + \sigma_{Bn}^2}{2\cdot (i\cdot (i+1)+1)} \tag{6.26}$$

and

$$\Gamma_B = \mathcal{P}(S_B)\cdot \sum_{i\in\Psi} \mathcal{P}_{AB}^{i}\cdot \mathcal{P}_{BA}^{i}\cdot \frac{i^2\cdot (\sigma_{BA}^2 + \sigma_{AB}^2) + \sigma_{Bn}^2}{2\cdot i^2 + 1} \tag{6.27}$$

Similar to the function Λ_A, the function Γ_A represents the weighted standard deviation summation of the S_M staying time for different sequences of state transitions between states S_A and S_B, starting from state S_A, until the final transition to the state S_n. Each sequence of state transitions is weighted by the conditional probability $\mathcal{P}(S_A)\cdot \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}$ as in function Λ_A, and limited by the set Ω in the same way.

Similar now to the function Λ_B, the function Γ_B represents the weighted standard deviation summation for different sequences of state transitions between states S_A and S_B, now starting from state S_B, until the final transition to the state S_n. Each sequence of state transitions is weighted by the conditional probability $\mathcal{P}(S_A)\cdot \mathcal{P}_{AB}^{i+1}\cdot \mathcal{P}_{BA}^{i}$ as in function Λ_B, and limited by the set Ψ in the same way.

As with the mean time function in the Equation (6.19), functions Δ_A and Δ_B are used by the standard deviation function of time in Equation (6.25) to accumulate the considered conditional probabilities starting from S_A and S_B, respectively, and utilised for normalising the weighted sums Γ_A and Γ_B.

Then, in the same way as Equation (6.24), in order to obtain the estimation of the

standard deviation $\sigma_{T_{S_M}}^{(E_{M\to n})}$ of the time T_{S_M}, the Equation (6.28) can be utilised.

$$\sigma_{T_{S_M}}^{(E_{M\to n})} = \sqrt{\frac{\mathcal{P}(E_{B\to n})\cdot \sigma^2_{\tau_{S_M}^{(E_{M\to n})}}(S_A,S_B) + \mathcal{P}(E_{A\to n})\cdot \sigma^2_{\tau_{S_M}^{(E_{M\to n})}}(S_B,S_A)}{\mathcal{P}(E_{B\to n}) + \mathcal{P}(E_{A\to n})}}. \quad (6.28)$$

This equation calculates the standard deviation $\sigma_{T_{S_M}}^{(E_{M\to n})}$ as the weighted mean of the estimators $\sigma_{\tau_{S_M}^{(E_{M\to n})}}$, in the same way as Equation (6.24).

Functions Λ_A, Λ_B, Γ_A, Γ_B, Δ_A, and Δ_B where built to represent the hardest situation where states S_A and S_B form an event loop, as depicted in Figure 6.9. These functions can be extremely simplified in more simple cases where the event loop is broken. If there is no event defined from the state S_B to the S_A, the functions simplify to:

$$\begin{aligned}
\Lambda_A &= \mathcal{P}(S_A)\cdot \mathcal{P}_{AB}\cdot(\mu_{AB}+\mu_{Bn}),\\
\Lambda_B &= \mathcal{P}(S_B)\cdot \mu_{Bn},\\
\Gamma_A &= \mathcal{P}(S_A)\cdot \mathcal{P}_{AB}\cdot(\sigma^2_{AB}+\sigma^2_{Bn}),\\
\Gamma_B &= \mathcal{P}(S_B)\cdot \sigma^2_{Bn},\\
\Delta_A &= \mathcal{P}(S_A)\cdot \mathcal{P}_{AB},\\
\Delta_B &= \mathcal{P}(S_B)
\end{aligned} \quad (6.29)$$

If there is no event defined from the state S_A to the S_B, the functions then simplify to:

$$\begin{aligned}
\Lambda_A &= 0,\\
\Lambda_B &= \mathcal{P}(S_B)\cdot \mu_{Bn},\\
\Gamma_A &= 0,\\
\Gamma_B &= \mathcal{P}(S_B)\cdot \sigma^2_{Bn},\\
\Delta_A &= 0,\\
\Delta_B &= \mathcal{P}(S_B)
\end{aligned} \quad (6.30)$$

Then, considering these simplifications, the Equations (6.24) and (6.28) reduce to $\mu_{\tau_{S_M}^{(E_{M\to n})}} = \mu_{Bn}$ and $\sigma_{\tau_{S_M}^{(E_{M\to n})}} = \sigma_{Bn}$, respectively.

The approximated solution proposed for the events starting at state S_M is based on two assumptions. The first assumption is to consider that the time of permanence in the

state S_M, $T_{S_M} \sim \mathcal{N}(\mu_{T_{S_M}}^{(E_{M\to n})}, \sigma_{T_{S_M}}^{(E_{M\to n})})$ follows a Gaussian distribution. This assumption had been already considered for the definition of an event, in Section 6.1.1.

The second assumption is that the considered time variables T_S to combine are independent. This is a verifiable assumption, as two variables T_{S_c} and T_{S_d} in the hierarchy of states concepts are dependent only if the states S_c and S_d have a *kinship relation* between them (see Definition 6.4), but a state transition never happens between states with a *kinship relation*.

These assumptions allow to calculate the mean and variance for T_{S_M} as a linear combination of the mean and variance of other states or sequences of states, as the properties of the weighted sum of independent variables following a Gaussian distribution allow.

6.2.3.2 Split Operator

The split operator consists in replacing a state S with its children, as depicted in Figure 6.10. This process implies to suppress the state concept S together with all the events in which the state is involved. Then, the children of the state S must be included as children of the parent state of S.

The split process corresponds to the inverse process of the merge operator. However,

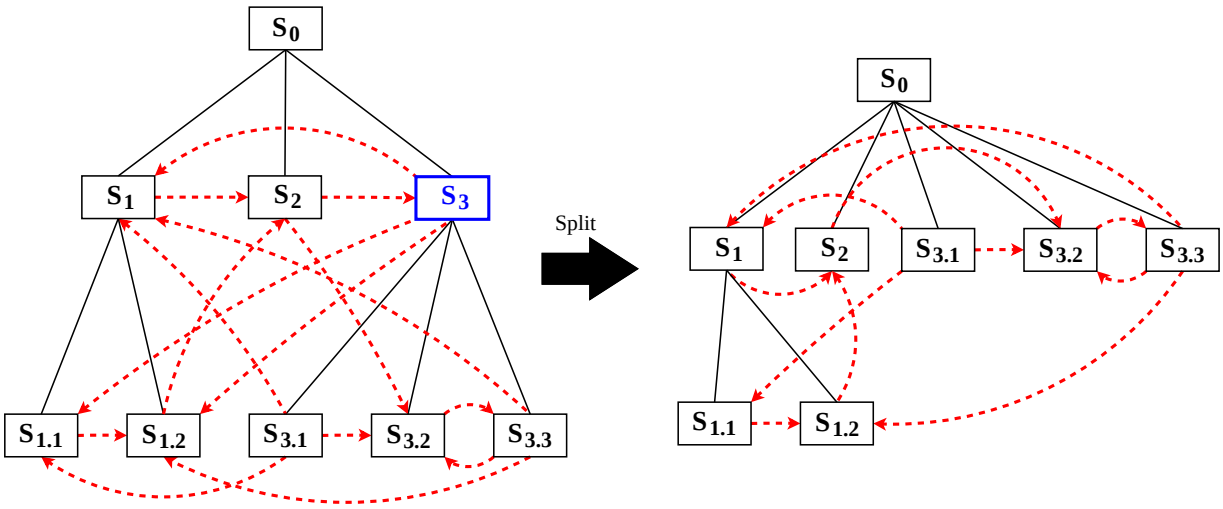


Figure 6.10: Split operator in MILES algorithm. The blue box represents the state to be split. Red dashed lines represent events. Notice that the split operator suppresses the state S_3 and its arriving and leaving events, and ascends the children of S_3 in the hierarchy.

the process involved in the split process is much more simple than the process for the merge operator. The reason for this difference in complexity is that the merge operator

has to create a state, events, and estimate parameters, while the split operator has just to destroy the proper elements. It is clear that it is always easier to destroy than to build.

6.2.4 Illustration of the Incremental Event Learning Algorithm

In order to better understand the learning process of the proposed algorithm for incremental event learning, an illustration example is presented in this section. The example consists in ten persons evolving in a metro scene, starting at different positions and time instants. A top view of the scene is depicted in Figure 6.11. The evolution of the persons in the scene is represented by ten hand-crafted trajectories (T0 - T9) of eight coordinate points (x,y) in the ground plane of the scene.

The scene consists of three Access/Exit zones (referenced in the Figure 6.11 as **A**, **C** and **D**), and a zone with a ticket vending machine, represented as a red box in Figure 6.11. The ten persons evolve in the scene over 13 time instants, as depicted in Table 6.1.

The idea is to utilise the (x,y) person positions presented in Table 6.1 as input of

	Time Instant												
ID	1	2	3	4	5	6	7	8	9	10	11	12	13
T0	(104,922)	(180,794)	(213,712)	(260,614)	(305,477)	(348,360)	(385,238)	(397,105)					
T1	(77,916)	(146,782)	(181,707)	(226,604)	(275,470)	(322,358)	(354,231)	(363,99)					
T2	(407,74)	(552,173)	(705,298)	(702,293)	(703,295)	(649,411)	(691,594)	(880,681)					
T3	(412,83)	(520,138)	(608,199)	(680,296)	(689,290)	(702,293)	(730,480)	(872,659)					
T4		(396,98)	(365,258)	(327,377)	(289,488)	(244,608)	(202,721)	(192,792)	(98,912)				
T5		(389,84)	(442,154)	(516,273)	(553,388)	(601,472)	(648,590)	(703,635)	(881,676)				
T6				(872,698)	(699,651)	(593,608)	(553,490)	(501,407)	(459,302)	(438,174)	(382,103)		
T7			(102,918)	(193,790)	(216,707)	(272,613)	(313,475)	(352,351)	(391,241)	(401,115)			
T8						(415,101)	(553,183)	(702,298)	(704,293)	(705,295)	(690,350)	(691,523)	(875,691)
T9					(870,701)	(702,654)	(594,607)	(561,492)	(515,404)	(465,297)	(436,169)	(387,104)	

Table 6.1: Ground-plane position (x,y) in the scene of the persons evolving in the scene of the illustration example. Positions are in centimetres. Blank spaces denote the absence of the person in the scene at the corresponding time instant.

the proposed event learning approach. Then, the evolution of the hierarchy of states and events in time can be analysed to understand the event learning process, and the relations between the obtained states and events and the trajectories of the persons can be studied to understand how the hierarchical representation represents the situations occurring in this scene.

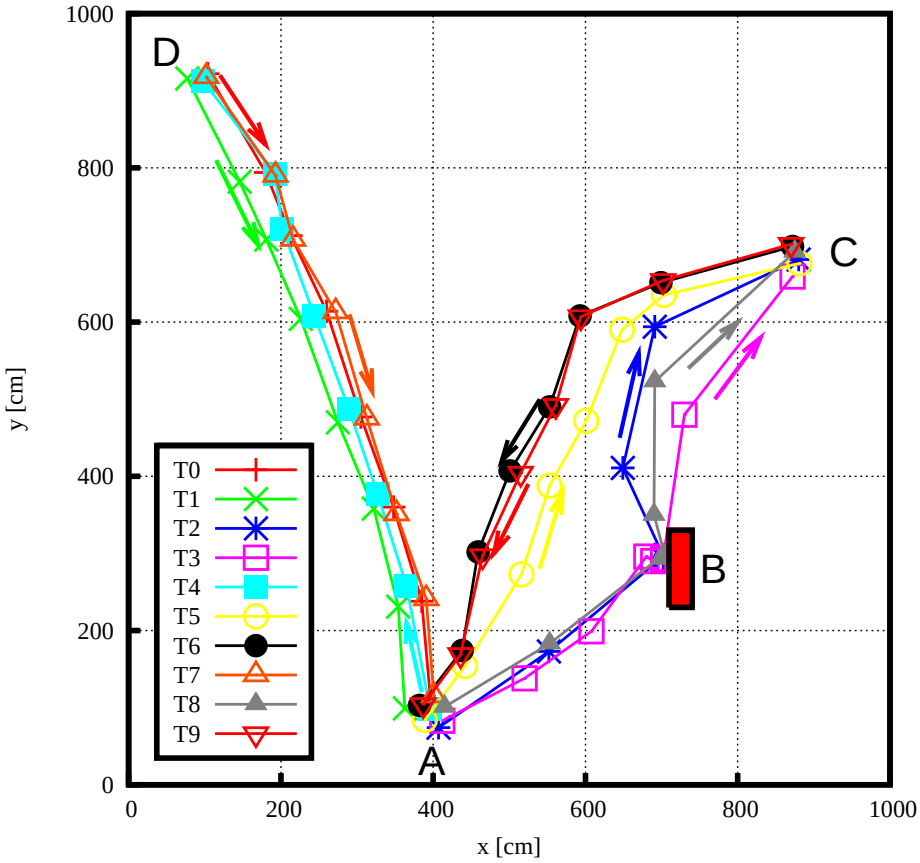


Figure 6.11: Top view of the metro scene illustration example. The plane (x,y) corresponds to the coordinates of the ground plane of the scene. The ten hand-crafted trajectories (T0 - T9) are displayed. The zones A, C, and D correspond to Access/Exit zones, while zone B corresponds to a ticket vending machine zone (where the vending machine is represented as a red box).

More formally, the learning context utilised by the event learning approach is described below:

Learning Context *Position* {
Involved Objects: Person
Attributes:
Numerical x : 200 [cm]
Numerical y : 200 [cm]
}

Note that the acuity value for the position attributes x and y has been fixed as 200 centimetres. This high value is intentionally high to control the size of the resulting hierarchy and allow its analysis. Next section describes how the learning process

constructs the hierarchy of states and events from the ground-plane position of the persons in the scene.

6.2.4.1 Incremental Event Learning Process

In order to understand the evolution of the hierarchy of states and events upon the arrival of new instances, the learning process is analysed at different time instants, explaining how the instances have influenced the creation, update, or deletion of the states and events of the hierarchy.

- **Learning up to Time instant 1:**

  At this instant two persons (represented by T0 and T1) arrive from the access zone **D** and two other persons (represented by T2 and T3) arrive from the access zone **A**.

  This situation is represented by two different states of the hierarchy, as depicted in Figure 6.12, because the person positions entering at the two different zones were similar enough to be represented in the same state concept. The positions of persons T0 and T1 are then represented by the State 1, while the positions of persons T2 and T3 are represented by the State 2.

  Figure 6.13(a) shows a top view of the scene where these the two new states

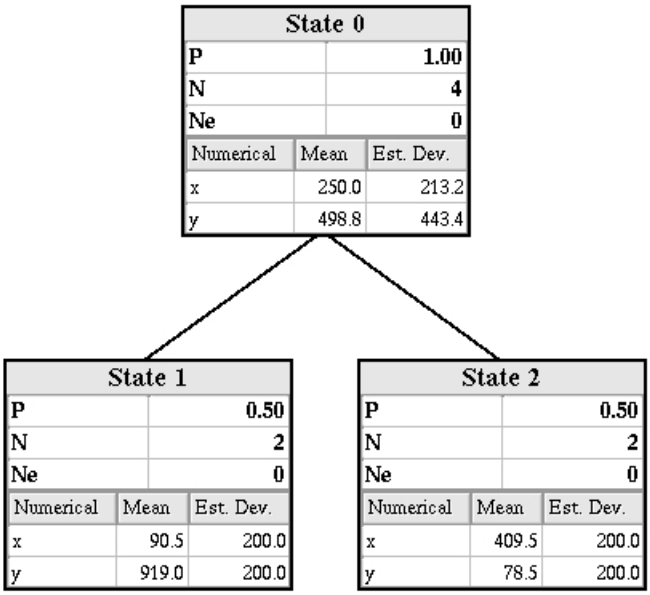
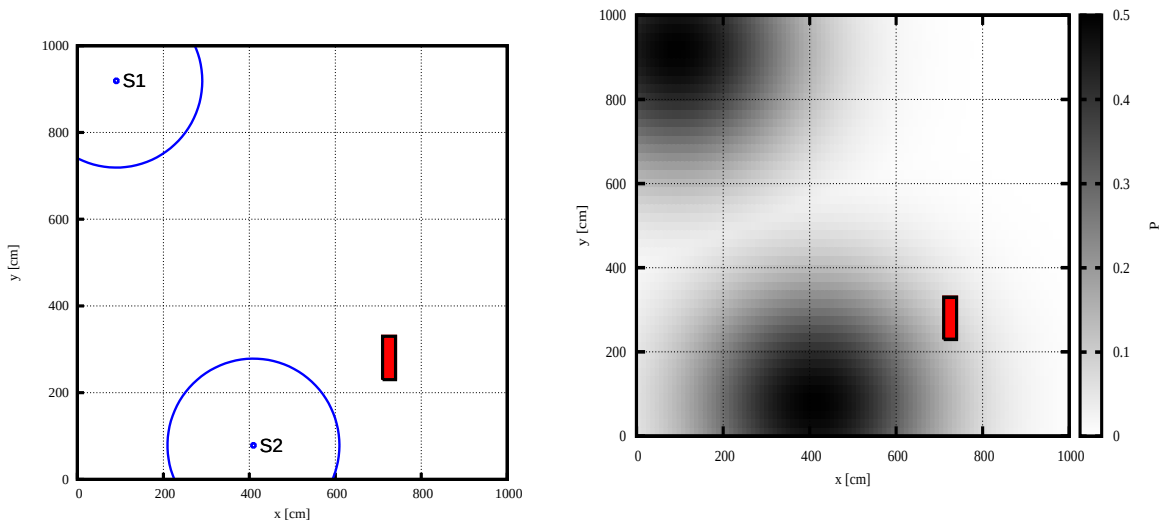


Figure 6.12: Hierarchy of states and events obtained for instant 1. No events have occurred yet.

  are represented. Figure 6.13(b) depicts the maximal marginal probability for each point in the scene, given the current two states of the hierarchy.



(a) (b)

Figure 6.13: Graphical representation of the states and events hierarchy associated to the position learning context, at instant 1. Figure (a) shows the position of the terminal states in a top view scene. The oval surrounding the mean position of the state concept represents the standard deviation of this position. The blue colour represents a state in the first level of the hierarchy. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

- **Learning up to Time instant 3:**

  The evolution of the hierarchy of states and events until this instant is depicted in Figure 6.14.

  At previous instant 2, two new persons (T4 and T5) have arrived from access zone **A**. This situation has reinforced the probability of the State 2.

  At the current instant 3, person T4 starts walking in the direction of the exit zone **D**, while person T5 goes in the direction of exit zone **C**. The position of persons T4 and T5 is not different enough yet to generate a new state. Then the probability of the State 2 is still reinforced.

  The two persons represented by T0 and T1 walk in the direction of the exit zone **A**, but their position is similar enough to the position represented in the State 1, reinforcing its probability. Also, another person (T7) arrives from the access zone **D**, reinforcing the probability of the State 1 even more.

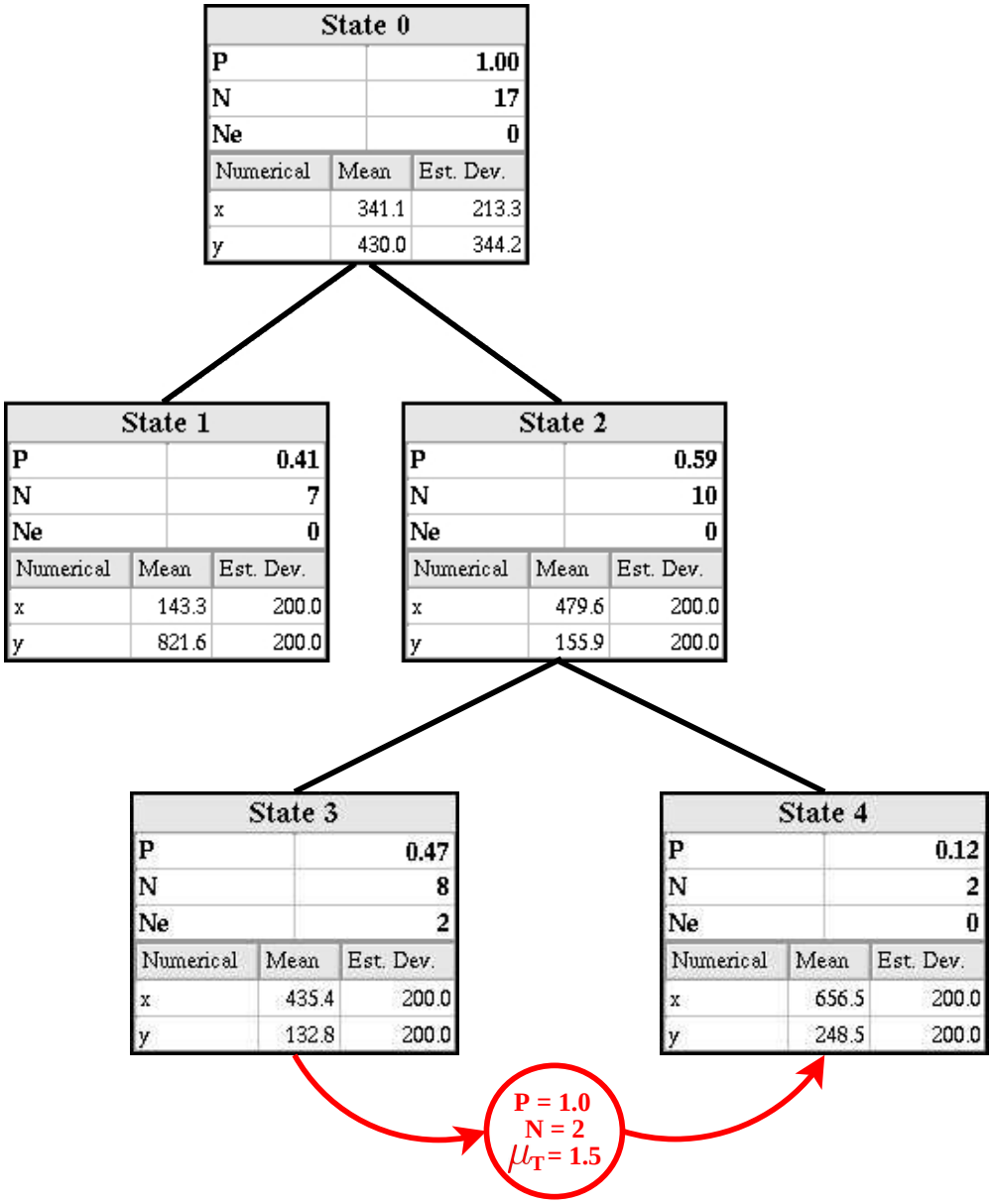
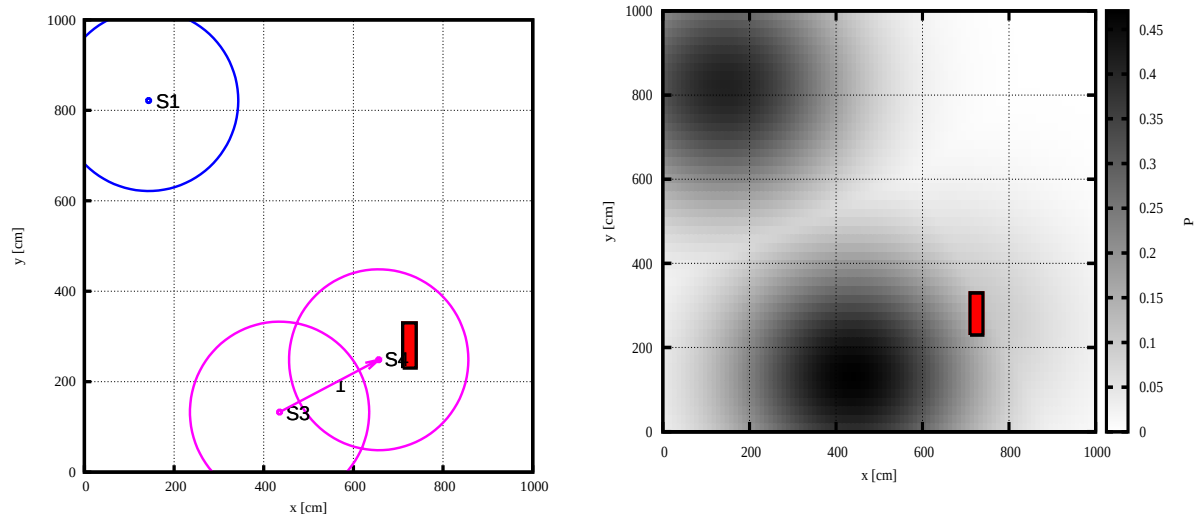


Figure 6.14: Hierarchy of states and events obtained up to instant 3. Events are coloured in red.

The persons T2 and T3 walk to the ticket vending machine **B**. Now, the position of these persons is different enough to the position represented by the State 2, to induce the creation of two children states of State 2, one state (State 3) representing the position near the access zone **A**, and the other representing the new created State 4 near the ticket vending machine zone **B**. The new positions of persons T2 and T3 have also induced a change of state, represented by the first event in the hierarchy between States 3 and 4. This event is depicted in Figure 6.14, and graphically represented by an arrow between States 3 and 4, in Figure6.15(a).

Note in Figure 6.15(b) that the new created state does not have a strong probability, compared with the other states of the hierarchy.



(a) (b)

Figure 6.15: Graphical representation of the states and events hierarchy associated to the position learning context, up to instant 3. Figure (a) shows the position of the terminal states and the events occurring between these states (represented as arrows with a transition probability) in a top view of the scene. The blue colour represents a state in the first level of the hierarchy, while the magenta colour a state on the second level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

- **Learning up to Time instant 4:**

  The evolution of the hierarchy of states and events until the instant 4 is depicted in Figure 6.16. From now and for simplicity, the attention in the analysis is focused on the person positions generating new states or events.

  At this instant, persons T0 and T1 have advanced enough from access zone **D** to induce the creation of two children from State 1 (States 5 and 6), inducing also an event between these new states.

  In the same way, person T4 becomes far enough from the position represented by State 3, to induce the creation of States 7 and 8, and an event between them. The new position of the person T5 at the current instant reinforces the probability of occurrence of the event between States 7 and 8.

  As State 7 represents the information of State 3 until the time instant before the introduction of the new States 7 and 8, the State 7 also contains the event transitions information inherited from State 3, and now considering the new event induced by person T4 and reinforced by person T5, the outgoing events from state 7 share an equal probability of occurrence of 0.5, as depicted in Figure 6.17(a).

  Also at this time instant, the person T6 arrives to the scene from the access zone **C**, inducing the creation of two new States 9 and 10, children of State 4.

  Note in Figure 6.16 that a state transition induces the creation and update of the states at all levels where there is no a *kinship* relation (see Section 6.2.2) between them, as is the case for the events between States 7 and 9, and States 7 and 4.

  Note also in Figure 6.17(b) that the probability near the ticket vending machine **B** is gaining strength as persons T2 and T3 stay near the vending machine.

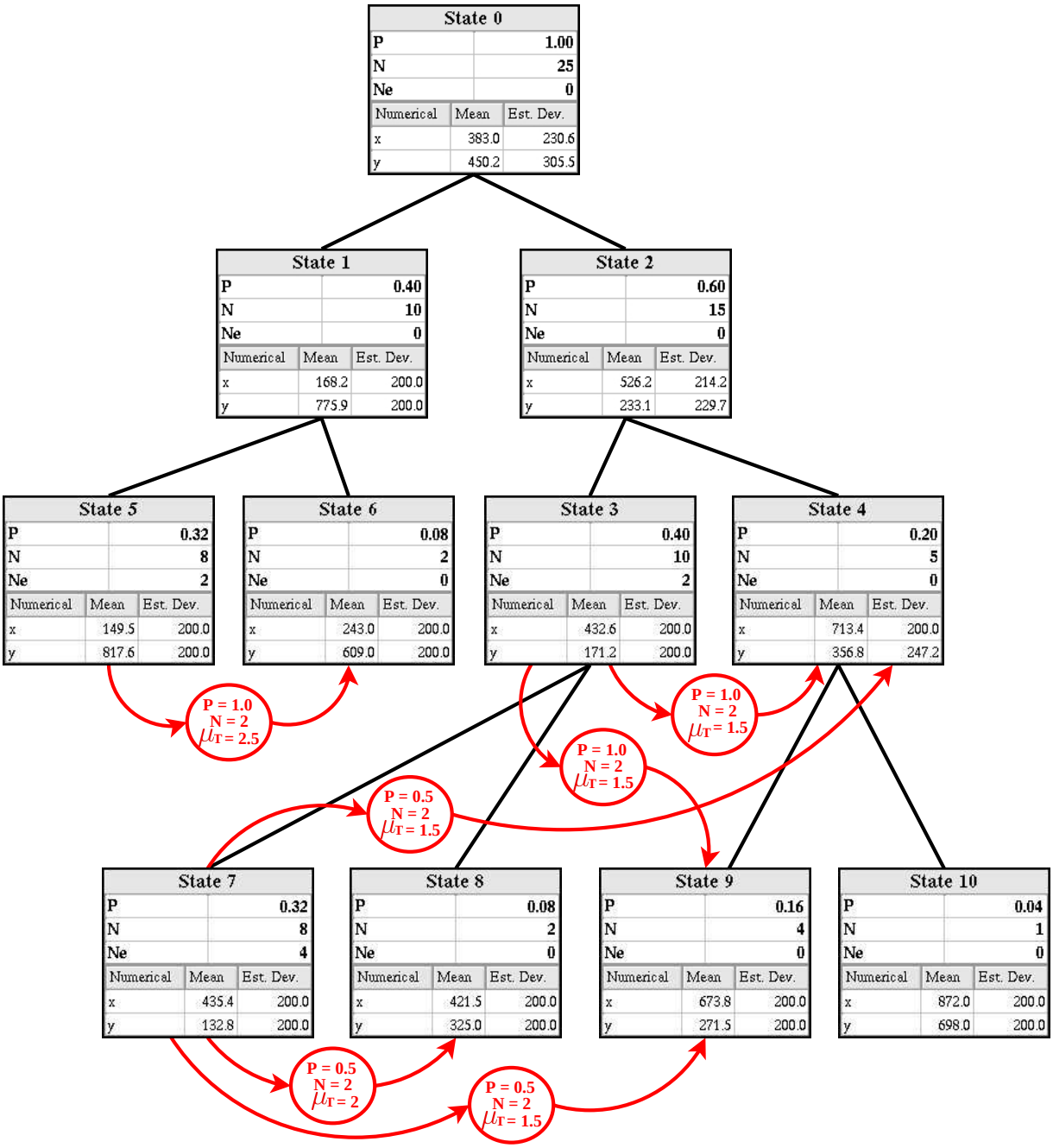
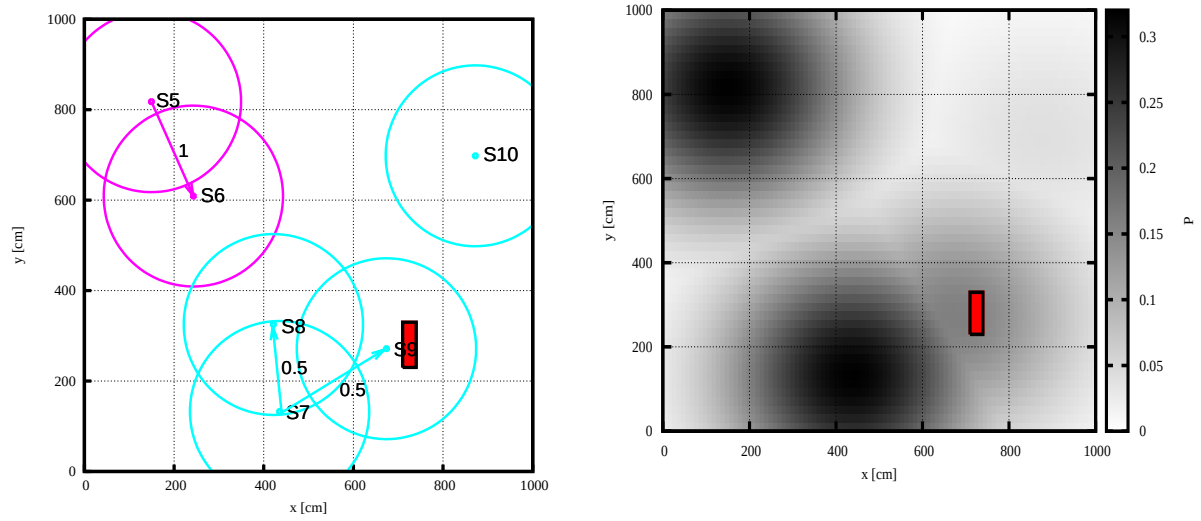


Figure 6.16: Hierarchy of states and events obtained up to instant 4. Events are coloured in red.



(a) (b)

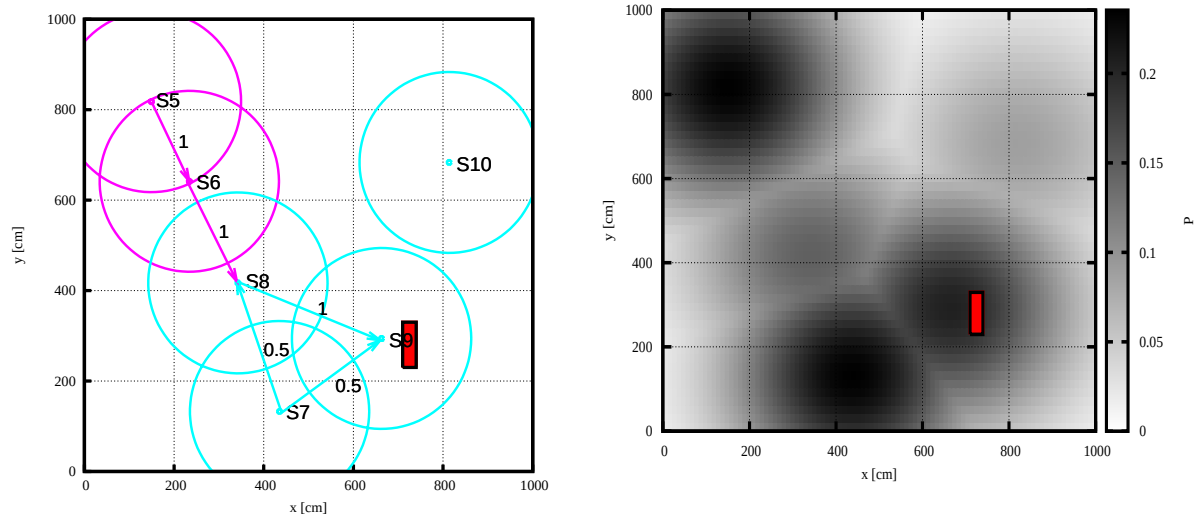
Figure 6.17: Graphical representation of the states and events hierarchy associated to the position learning context, at instant 4. Figure (a) shows the position of the terminal states and the events occurring between these states in a top view of the scene. The magenta colour represents a state in the second level of the hierarchy, while the cyan colour a state on the third level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

- **Learning up to Time instant 5:**

  At this time instant, the new position of person T4 produces an adjustment of the position of State 8, while the new position of person T5 induces the creation of a new event between States 8 and 9, as depicted in Figure 6.18(a). Person T5 walks in the direction of exit zone **C**, then the transition between States 8 and 9 seems imprecise, but this is one of the costs of considering a coarse value for the acuity of position attributes x and y.

  Also, the person T9 arrives to the scene from the access zone **C**, reinforcing the probability of State 10.

  Note in Figure 6.18(b) that the permanence of persons T2 and T3 at the vending machine zone **B** has reinforced the probability of the State S9 near this zone. Also note that the reposition of State 8, induced by person T4, has also reinforced the probability of occurrence of the State 8.



(a) (b)

Figure 6.18: Graphical representation of the states and events hierarchy associated to the position learning context, at instant 5. Figure (a) shows the position of the terminal states and the events occurring between these states in a top view of the scene. The magenta colour represents a state in the second level of the hierarchy, while the cyan colour a state on the third level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

- **Learning up to Time instant 6:**

  The evolution of the hierarchy of states and events until the instant 6 is depicted in Figure 6.19. This figure shows the level of complexity that can be managed with this representation. At this time instant several events have been induced and reinforced.

  The new position of person T6 has induced the creation of two children States 11 and 12, from State 10, and has also induced an event between these new states.

  At this time instant, the last person T8 enters to the scene from access zone **A**. Figure 6.20(a) shows the new events induced by the position of persons T4 (between States 8 and 6), and T6 (between States 11 and 12).

  Figure 6.20(b) shows the reinforcement of the probability of State 9 by persons T2, T3 and T5.

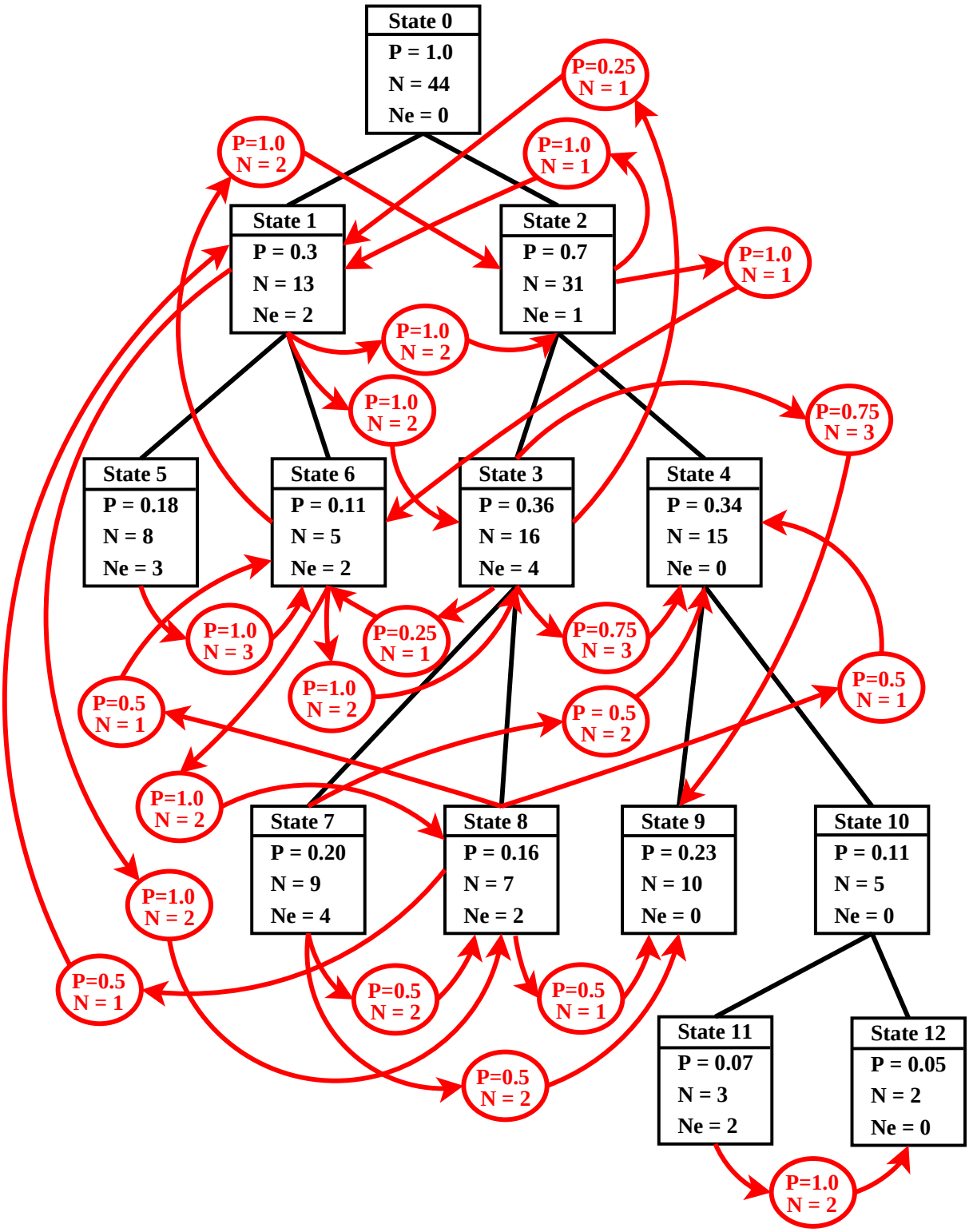
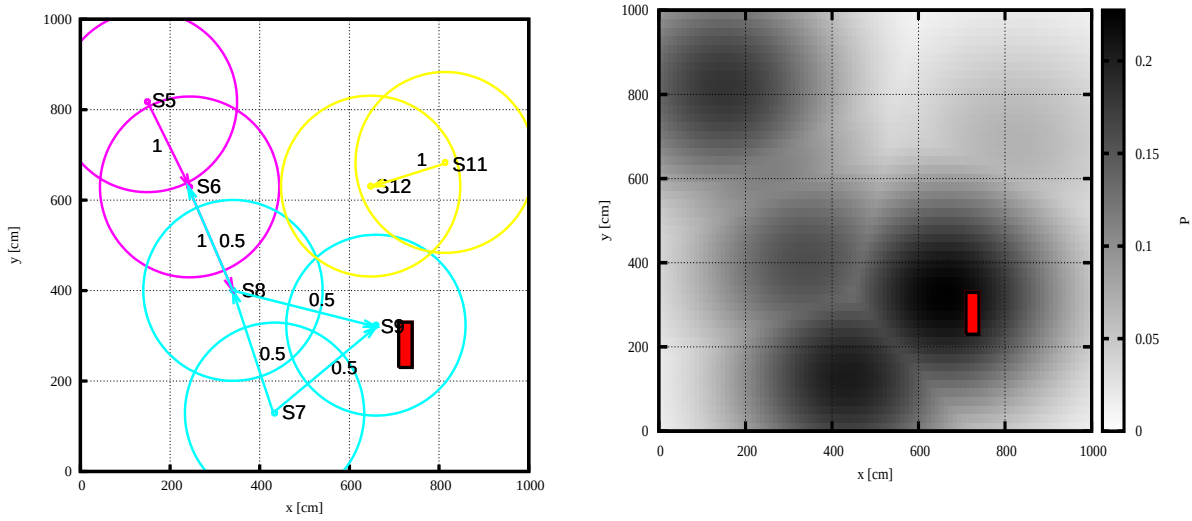


Figure 6.19: Hierarchy of states and events obtained up to instant 6. State graphical representation have been reduced for simplicity. Events are coloured in red.



(a) (b)

Figure 6.20: Graphical representation of the states and events hierarchy associated to the position learning context, at instant 6. Figure (a) shows the position of the terminal states and the events occurring between these states in a top view of the scene. The magenta colour represents a state in the second level of the hierarchy, the cyan colour a state on the third level, and the yellow colour a state on the fourth level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

- **Learning up to Time instant 7:**

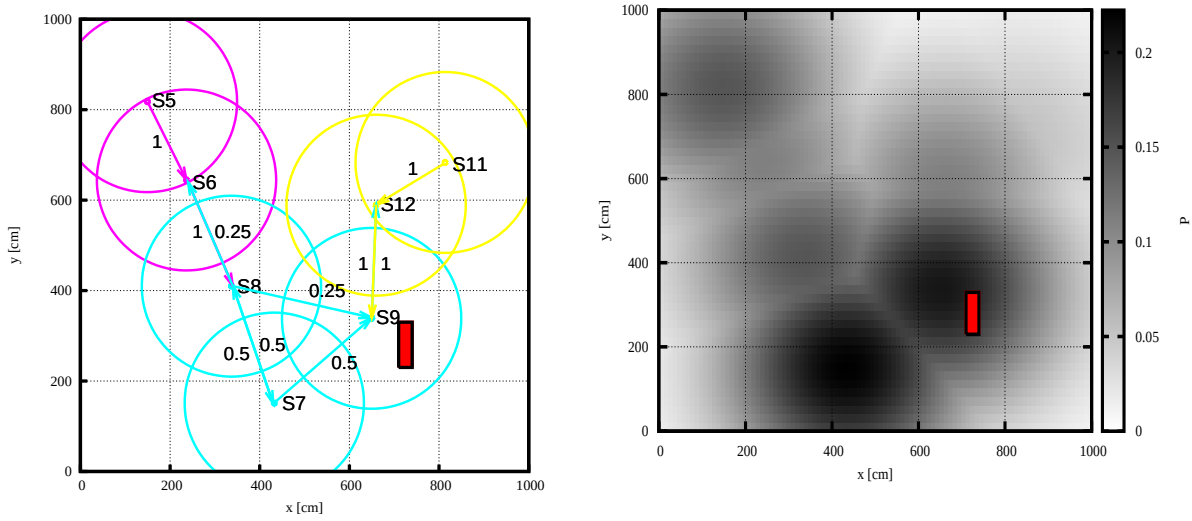
  At this time instant, the hierarchy of states and concepts has arrived to a stable number of states.

  The new position of person T6 induces a new event between States 12 and 9. At the same time, the position of person T2 induces a new event between States 9 and 12 (in that order), as depicted in Figure 6.21(a).

  Figure 6.21(b) shows that even the probability map has arrived to a quite stable state, where only slight differences can be observed.

  From this time instant and until the end of the illustration example, the hierarchy tree structure is very stable, only showing some new events and updates in the probability of the states.

- **Learning up to Final time instant 13:**



(a) (b)

Figure 6.21: Graphical representation of the states and events hierarchy associated to the position learning context, at instant 7. Figure (a) shows the position of the terminal states and the events occurring between these states in a top view of the scene. The magenta colour represents a state in the second level of the hierarchy, the cyan colour a state on the third level, and the yellow colour a state on the fourth level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

The final result for the hierarchy of states and events of this illustration example at the instant 13 is depicted in Figure 6.22. This figure shows that the hierarchy has arrived to a stable state since time instant 6.

In Figure 6.23 only slight differences can be observed, with some few new events and slight modifications in the probability map.

6.2.4.2 Summary

This illustration has served to show the incremental nature of the proposed event learning approach.

The hierarchy of states and events has shown a consistent behaviour on representing the frequency of states and events induced by the persons of the illustration example. The representation has converged to a stable number of states at time instant 6. Figure 6.24 shows the evolution of the number of states and events over the complete learning process.

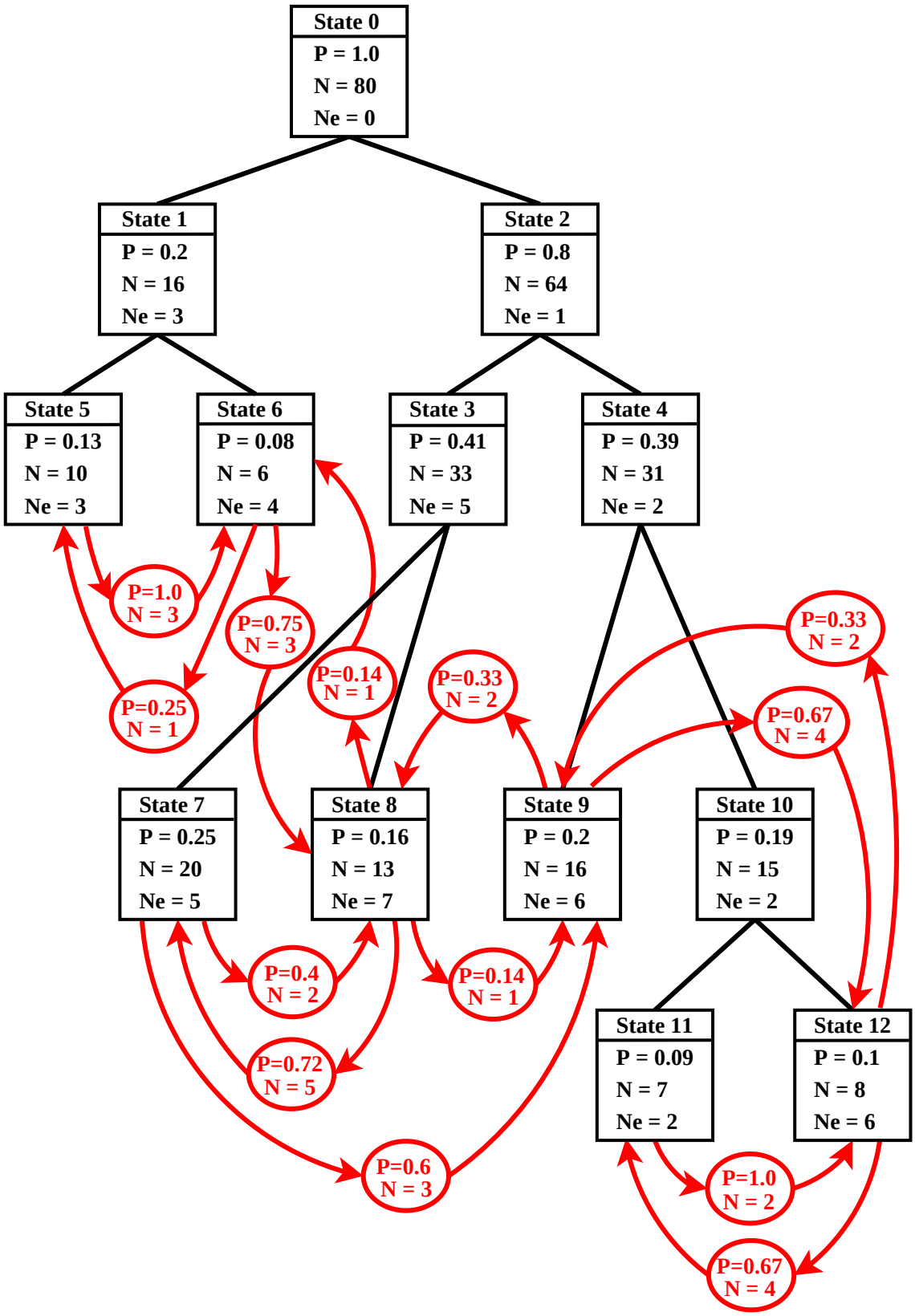
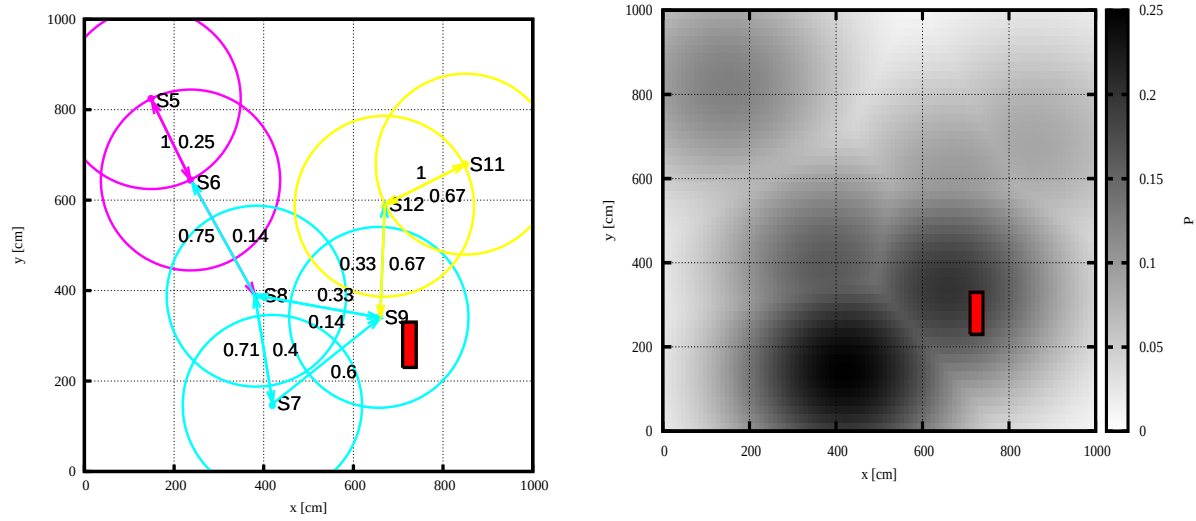


Figure 6.22: Final hierarchy of states and events obtained up to instant 13. For simplicity, only events between terminal states are displayed.



(a) (b)

Figure 6.23: Final graphical representation of the states and events hierarchy associated to the position learning context, at instant 13. Figure (a) shows the position of the terminal states and the events occurring between these states in a top view of the scene. The magenta colour represents a state in the second level of the hierarchy, the cyan colour a state on the third level, and the yellow colour a state on the fourth level. Figure (b) depicts the maximal marginal probability of a state of the hierarchy for a given position. A darker colour represents a higher probability.

Note that the number of events grows far quicker than the number of states, as a single state transition can induce the creation of events in several different levels in the hierarchy. Nevertheless, both the number states and events show a converging behaviour on the number of states and events.

6.3 Discussion

The proposed learning approach has been conceived to be able to learn state and event concepts in a general way. Depending on the availability of tracked object attributes, the possible combinations for learning contexts is enormous. The attributes already proposed by the object tracking approach presented in Chapter 5.3 give a sufficient information to flexibly explore a large variety of scenarios. Anyway, users can always define more object attributes, by either combining existing attributes or by creating new ones from new object descriptors.

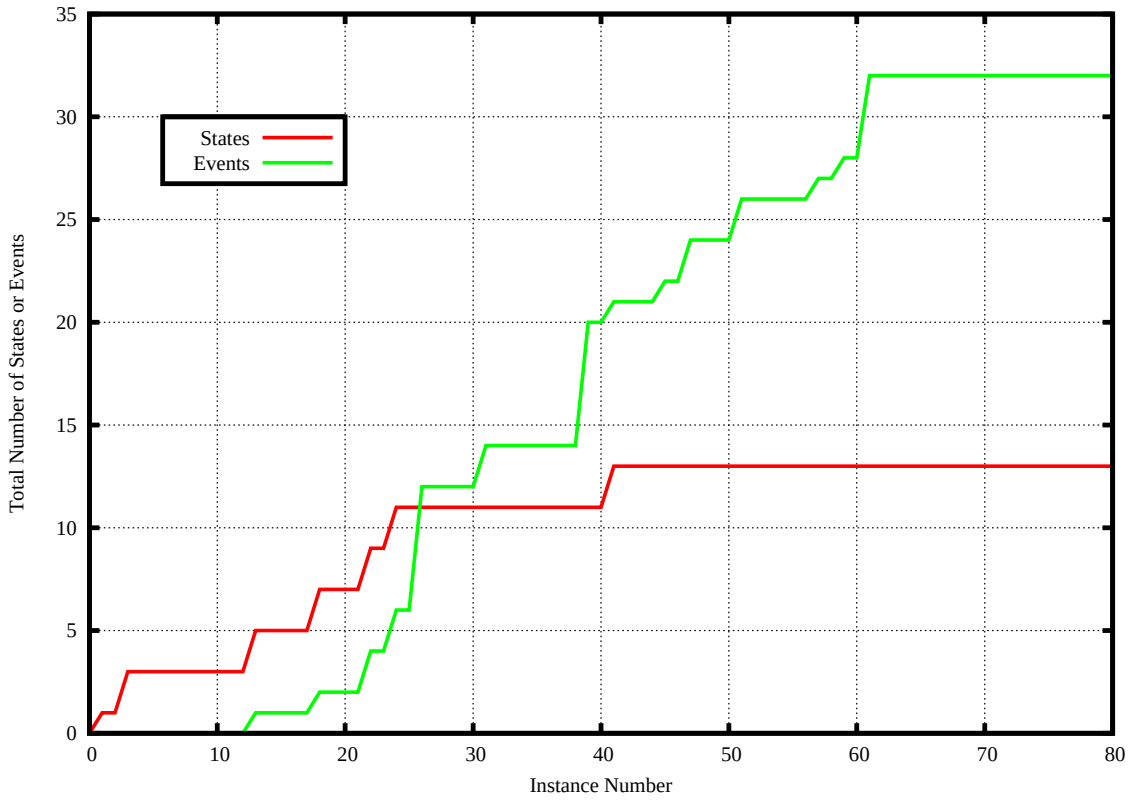


Figure 6.24: Evolution of the number of states and events over the complete learning process in the illustration example.

The incremental nature of the proposed event learning algorithm MILES, allows to obtain a learning performance that can be utilised in on-line learning for real world applications. The main contributions of MILES with respect to its predecessors are the following:

- The main contribution of MILES is the utilisation of incremental concept learning models to learn the states as a hierarchy of concepts and to extend the incremental concept learning hierarchy to learn the events as first order temporal relations between the learnt states.
- Another contribution is the way of utilising the concepts of *cutoff* and *acuity*. Before, these concepts were treated as general parameters for an incremental concept learning algorithm. Now, the acuity is utilised as a way of representing the interest on an attribute for a given learning context. The cutoff is now defined as a function of the acuity values and of the symbolic attribute differences for the analysed learning context.
- Also, the extension to event learning has implied the redefinition of the existing merge and split hierarchy operators.
- Another important contribution is the consideration of reliability measures associated to the input data, which are utilised to guide the learning process through

the most reliable information.

- Finally, the definition of multiple learning contexts allows MILES to simultaneously learn several hierarchies of state and event concepts.

The proposed event learning approach presents the following limitations:

1. The first limitation is related to the order of instances processed by the learning approach. From the state of the art on incremental concept formation, it can be inferred that the distribution of state and event concepts in the generated hierarchy often depends in certain extent on the processing order of the object state instances. This means that different hierarchies can be obtained from different ordering of the same instances. This situation is not a serious issue as the objective of the learning approach is to build an adequate representation of the states and events occurring in the scene, not to find a unique optimal representation. Nevertheless, it seems interesting in the future to analyse the influence of the instance ordering in the quality of representation.

2. A second limitation can be identified with respect to the capability of the learning approach to represent relations between objects evolving in the scene. As the learning approach utilises the information related to each tracked object evolving in the scene separately, it does not seem inherent to the approach to represent relations between tracked objects. Nevertheless, the flexibility of the proposed approach allows the definition of attributes relating different tracked objects. For example, in a learning context regarding events where an object follows another object evolving in the scene, it is necessary to verify the speed attribute of the analysed object (an object following or followed is not stationary), and to define attributes evaluating the difference in the velocity direction between two objects (objects with similar direction of movement), the difference in the speed magnitude (objects with similar speed), and the distance between the objects (the objects should not be that far between each other).

3. A final limitation is the difficulty on determining the usefulness of the state hierarchy for the user, as state and event frequency is not equivalent to meaningful or interesting states and events.

A partial limitation of the approach is the limited capability of the state and event concept hierarchy on representing interactions between mobile objects, as no explicit way of representing these interactions is still available. This limitation is just partial because in most of the cases it is always possible to define mobile object attributes representing these interactions. For example, the learning context *object following object* can be defined with an attribute describing the distance between the objects and attributes describing the most stable velocity vector difference and distance between a mobile object and the other mobile objects evolving in the scene, and all these attributes can be calculated starting from the currently available attributes of the mobile objects.

The following Chapter 7 presents the evaluation of the complete video understanding framework, applying different experiments for validating general and specific functionalities of the approach.

Chapter 7

Evaluation and Results of the Proposed Approach

In order to evaluate the whole proposed video understanding framework, several experiments have been developed. The main objectives of these experiments are to validate the different phases of the video understanding framework, to highlight interesting characteristics of the approach, and to evaluate the potential of the framework for real world applications.

The performed experiments consist of:

- An evaluation of the classification algorithm for real world applications. In this experiment, two videos were tested for a parking lot and a bank locked chamber application. For more details, refer to Section 7.2.1.

- A comparative performance analysis of the proposed tracking approach, utilising four benchmark videos publicly accessible[1]. The tracking approach has been tested using the evaluation framework proposed in ETISEO project [Nghiem et al. 2007]. ETISEO is a video understanding evaluation project which covers video understanding applications in real contexts, providing ground truth data for all the video sequences, created manually including camera calibration information, evaluation metrics, and automatic evaluation tools. The evaluation of the proposed tracking approach considers the metric for object tracking proposed in ETISEO. This experiment is presented in Section 7.2.2.

- Finally, an evaluation of the complete video understanding framework in a real world application is performed. It consists in analysing video sequences from GERHOME project for elderly care at home [GERHOME 2005], [Zouba et al. 2007] with several learning contexts that can be interesting in real world applications. This experiment has multiple objectives, as evaluating the influence of the utilisation of reliability measures, the processing time performance of the framework, and the capability of

[1] Access to ETISEO project videos at **http://www-sop.inria.fr/orion/ETISEO/download.htm**

the system of bridging the gap between image processing and event learning tasks in video understanding. The experiment is detailed in Section 7.2.3.

This chapter is organised as follows. First, Section 7.1 describes the metrics utilised in the evaluation of the video understanding framework. Second, the different performed experiments are described in Section 7.2. Finally, Section 7.3 presents a conclusion about the experiments.

7.1 Evaluation Metrics

Different metrics have been used according to the nature of the experiment.

For the classification algorithm experiment (Section 7.2.1), the utilised metrics are:

- **True Positive (TP):** It corresponds to the number of objects correctly classified according to the ground truth.
- **False Positive (FP):** It corresponds to the number of objects which class does not correspond to the ground truth.
- **False Negative (FN):** It corresponds to the number of not classified objects, which are present in the ground truth.
- **Sensitivity:** The sensitivity measures the proportion of actual positives which are correctly identified as such. A sensitivity of 100% means that the test recognizes all the actual positives as such. Then, this metric is formally defined as:

$$sensitivity = \frac{TP}{TP + FN} \tag{7.1}$$

- **Precision:** The precision metric can be seen as a measure of exactness or fidelity. The precision for a class corresponds to the number of instances correctly labelled as belonging to the class divided by the total number of elements labelled as belonging to the class. Then, this metric is formally defined as:

$$precision = \frac{TP}{TP + FP} \tag{7.2}$$

Note that, when an object is classified with a class different from the ground truth, this situation is considered as two errors at the same time (one FP and one FN), while not classifying it at all is considered as just one FN.

For the tracking algorithm experiment (Section 7.2.2), the **Tracking Time** metric utilised in ETISEO project for evaluating object tracking has been used. 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 (7.3)).

$$D1 = \frac{2 \cdot area(RD \cap C)}{area(RD) + area(C)} \tag{7.3}$$

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 (7.4).

$$T_{Tracked} = \frac{1}{NB_{RefData}} \sum_{RefData} \frac{card(RD \cap C)}{card(RD)}, \tag{7.4}$$

where the function $card()$ corresponds to the cardinality in terms of frames.

For the video understanding framework experiment (Section 7.2.3), the metrics corresponding to the **Number of States** NB_S and **Number of Events** NB_E of a hierarchy on a given image frame are utilised, in order to analyse the evolution of the growth of the hierarchy in time.

Also, the metric of **Recognition rate** metric is utilised for evaluating the quality of matching between a recognised state and an instance. This metric is defined in Equation (7.5).

$$P_r = \frac{\sum_{f=1}^{F} \left(\frac{\sum_{i=1}^{I} R_i^{(f)} \cdot \mathcal{P}(V_i^{(f)} | v_i^{(f)})}{\sum_{i=1}^{I} R_i^{(f)}} \right)}{F}, \tag{7.5}$$

where $R_i^{(f)}$ is the reliability of the value for the attribute i for the instance associated to a state at the f video frame. $\mathcal{P}(V_i^{(f)} | v_i^{(f)})$ is the probability of occurrence of an instance attribute value $V_i^{(f)}$ given the model of an attribute $v_i^{(f)}$ in the associated state, at video frame f. For numerical attributes this probability follows a Gaussian distribution, while for symbolic attributes this probability is explicit for each possible value of the attribute. The recognition rate metric P_r is the summation of the mean of all probabilities $\mathcal{P}(V_i^{(f)} | v_i^{(f)})$ weighted by the reliability $R_i^{(f)}$ of the instances attribute values V_i for all the F frames that the object has been associated to the state, divided by this number of

frames.

Also, the following processing time performance measures are utilised for evaluating the tracking and learning tasks:

- $\overline{T_p}$ for mean processing time per frame.
- $\overline{F_p}$ for mean frame rate.
- σ_{T_p} for the standard deviation of the processing time per frame.
- $T_p^{(max)}$ for the maximal processing time utilised in a frame.

Next Section 7.2 presents the performed experiments in order to validate this thesis work.

7.2 Performed Experiments

The next sections present the experiments performed to evaluate the proposed video understanding framework. Section 7.2.1 presents an experiment for evaluating the capability of the classification algorithm for real world applications. Then, Section 7.2.2 presents a comparative analysis for the proposed tracking approach. Finally, Section 7.2.3 presents an experiment for evaluating the whole video understanding framework.

7.2.1 Classification Algorithm Applications

In this experiment, two types of videos have been tested for evaluating the object classification approach presented in Chapter 4 in real world applications. The first type of videos corresponds to a parking sequence where cars and persons interact. Two object models are used for this sequence. The evaluation objective of this video is to validate the capability of the approach for coping with the problem of object orientation and relative position to camera.

The second type of videos corresponds to a lock chamber from a bank camera, with high 2D change in shape of the detected blobs because of the proximity of persons to the camera. For these videos, three models representing one person and groups of two and three persons are defined (the space of the chamber allows a maximum of three persons at the same moment). The lock chamber video is used to validate the approach capability to detect objects which highly change in shape, and to differentiate between very similar classes.

Ten short video sequences of 20 frames have been utilised for each type of videos, giving a total of 400 analysed frames. The selected sequences consider situations of different distances between objects and the camera focal point, and different object orientations.

A computer Intel Pentium IV, Xeon 3000 Mhz, has been used for performing these tests. For each sequence, the evaluation counts true positives, false positives, and false negatives.

The *precision* and *sensitivity* evaluation metrics have been also calculated for these tests. In Figures 7.1, 7.2, and 7.3, each detected object is enclosed by a 2D bounding box and by the corresponding 3D parallelepiped. The base of parallelepiped is represented by blue lines, while projected lines in height h are represented by green lines. 2D bounding boxes take different colours according to the classified object (person: red, 2 persons: green, 3 persons: blue, car: brown). Cars in parking sequence that seem not detected are considered as part of the background of the scene.

7.2.1.1 Results

For the parking sequence, 3D models for persons and cars were pre-defined. The results for this sequence are shown in Table 7.1 and images of these results are shown in Figures 7.1 and 7.2. Parking results show a very good performance, obtaining a global precision of 0.98. The encountered errors have been caused by poor segmentation in some frames because of illumination changes and shadows. The method has been able to discriminate objects at different orientations and positions relative to the camera. For instance, Figure 7.1(b) shows the same person in two different frames detected as a person, showing the method capability for coping with different positions relative to the camera.

Figure 7.2(b) shows a very difficult case of person detection, because of its distance to the camera (left image), is successfully detected in the classification task (right image). Figures 7.1(c) and 7.1(d) show the capability of the method for coping with different positions and orientations of cars and for coping with more than one object class at the same frame.

For the bank locked chamber sequence, models for one, two and three persons have

Name	**Description**	**TP**	**FP**	**FN**	**Precision**	**Sensitivity**
Borel 1	1 car parking right	20	0	0	1.0	1.0
Borel 2	1 person to bottom	20	0	0	1.0	1.0
Borel 3	1 car going far	20	0	0	1.0	1.0
Borel 4	1 car parking left	20	3	0	0.86	1.0
Borel 5	1 car and 1 person	39	3	1	0.93	0.98
Borel 6	2 cars	40	0	0	1.0	1.0
Borel 7	2 persons bottom	40	0	0	1.0	1.0
Borel 8	1 person very far	20	0	0	1.0	1.0
Borel 9	2 persons walking	40	0	0	1.0	1.0
Borel 10	2 cars very near	40	0	0	1.0	1.0
	Mean Values	**29.9**	**0.6**	**0.1**	**0.98**	**0.99**

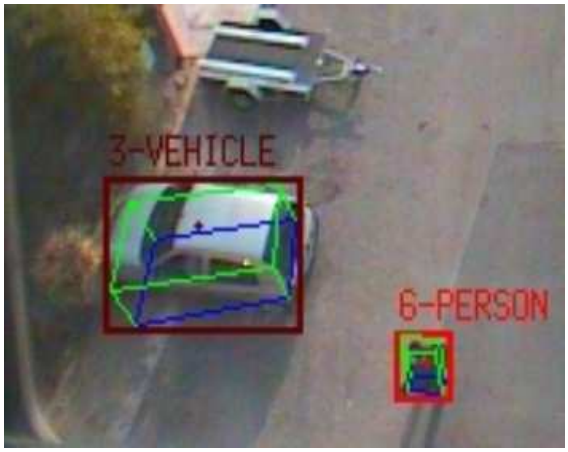
Table 7.1: Obtained classification results for parking video.

(a)

(b)

(c)

(d)



(e)

(f)

Figure 7.1: Results for different frames of the parking video. Figures (b) and (e) correspond to zoomed versions of captured frames. Parked vehicles are considered as background.

(a) (b)

(c) (d)

Figure 7.2: More results for different frames of the parking video. Figure (b) corresponds to a zoomed version of the captured frame. Parked vehicles are considered as background.

been defined, where the model of one person is identical to the person model used in the parking video. The results for the bank locked chamber sequence are shown in Table 7.2 and images of these results are shown in Figure 7.3.

Locked chamber results show a very good performance, obtaining a global precision of 0.95. The encountered errors have been principally caused by the proximity between pre-defined models. The obtained results for some sequences are sometimes very similar with the next class (one person similar with two persons, or two persons similar with three) because of some postures and configurations of persons, that lead to some misclassification. However, in terms of results, the method shows the different configurations with similar likelihood that could occur, which could be a beneficial situation for other purposes.

Table 7.3 shows the confusion matrix of the classification results for bank locked chamber

Name	**Description**	**TP**	**FP**	**FN**	**Precision**	**Sensitivity**
Sas 1	1 p. with folder	20	0	0	1.0	1.0
Sas 2	1 mean height p.	20	0	0	1.0	1.0
Sas 3	1 tall p.	17	3	3	0.85	0.85
Sas 4	2 p. semi-ext. arms	20	0	0	1.0	1.0
Sas 5	2 p. not aligned	18	2	2	0.90	0.90
Sas 6	2 p. aligned	20	0	0	1.0	1.0
Sas 7	2 p. extended arms	15	5	5	0.75	0.75
Sas 8	3 p. 1	20	0	0	1.0	1.0
Sas 9	3 p. 2	19	1	1	0.95	0.95
Sas 10	3 p. 3	20	0	0	1.0	1.0
	Mean Values	**18.9**	**1.1**	**1.1**	**0.95**	**0.95**

Table 7.2: Obtained classification results for bank locked chamber video.

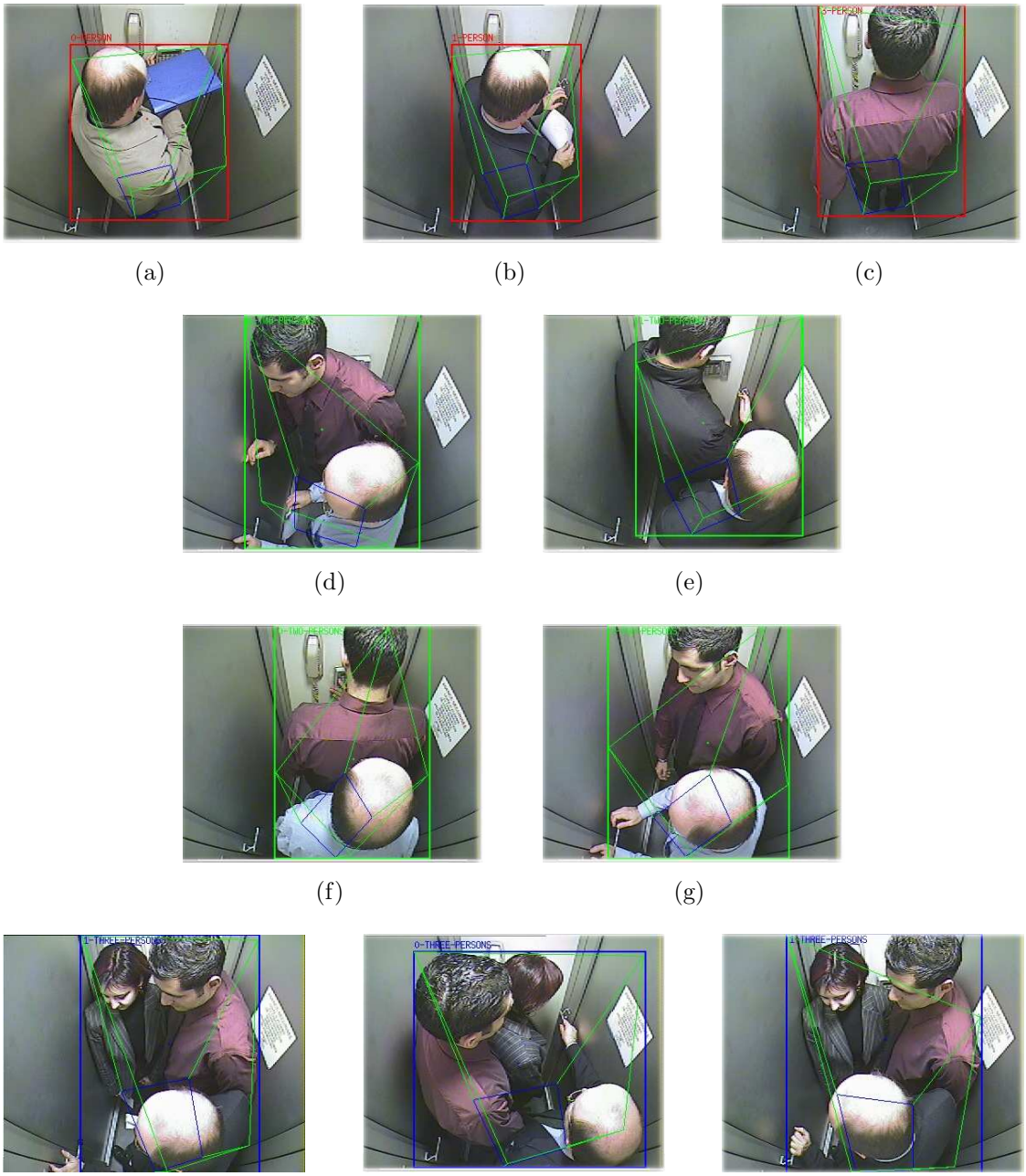
application. Each row represents ground truth and each column represents the detected object. Notice that committed errors were always associated with the detection of more or less one person, compared with the real number of persons.

Another application for the bank locked chamber sequence consists in generating alarms

	1p	**2p**	**3p**
1p	**57**	3	0
2p	0	**73**	7
3p	0	1	**59**

Table 7.3: Confusion matrix for classification results for bank locked chamber video, considering objects one-person (**1p**), two-persons (**2p**) and three-persons (**3p**).

if more than one person is at the same time in the locked chamber. In this case, a TP corresponds to the detection of more than one person when more than one person is



(a) (b) (c)

(d) (e)

(f) (g)

(h) (i) (j)

Figure 7.3: Results for different frames of the bank locked chamber video. Ten frames for the selected sequences are shown. Figures (a), (c), and (d) show examples of classification for the three different classes. Figure (b) shows the case of a tall person, who has been sometimes misclassified as two persons. The bounding box of one person is coloured in red, of two persons in green, and of three persons in blue.

present on the scene, a TN corresponds to the detection of one or zero person when one or zero person is in the scene, a FP corresponds to the detection of more than one person when one or zero persons are present in the scene, and FN corresponds to the detection of one or zero person when more than one person is in the scene. Here, 140 TP, 57 TN, 3 FP and 0 FN were found, giving a precision of 0.98 and a sensitivity of 1.

7.2.1.2 Experiment Conclusion

This experiment has shown good results in object classification, with high success rate for both analysed videos. The proposed approach has been able to cope mainly with the problems of object position relative to the camera position, object orientation and dimensional deformation caused by camera proximity, with high classification rates.

The analysis of the results obtained in the locked-chamber video in the bank application shows that the classification method is able to discriminate even between very similar object models, with very low error rate.

7.2.2 Comparative Analysis of the Object Tracking Algorithm

The objective of this experiment is to evaluate the performance of the proposed tracking approach, presented in Chapter 5. For this purpose four benchmark videos publicly accessible have been evaluated. These videos are part of the evaluation framework proposed in ETISEO project [Nghiem et al. 2007]. The obtained results have been compared with other algorithms which have participated in the ETISEO project.

From the available videos of the ETISEO project, the four chosen videos are:

- **AP-11-C4:** Airport video of an apron (AP) with one person and four vehicles evolving in the scene over 804 frames (Figure 7.4(a)).
- **AP-11-C7:** Airport video of an apron (AP) with five vehicles evolving in the scene over 804 frames (Figure 7.4(b)).
- **RD-6-C7:** Video of a road (RD) with approximately 10 persons and 15 vehicles evolving in the scene over 1200 frames (Figure 7.4(c)).
- **BE-19-C1:** Video of a building entrance (BE) with three persons and one vehicle over 1025 frames (Figure 7.4(d)).

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 $\pi/40[rad]$.

(a) (b)

(c) (d)

Figure 7.4: Benchmark videos utilised for the evaluation of the proposed object tracking approach. Figures (a) and (b) correspond to apron videos. Figure (c) shows a road video. Figure (d) shows a building entrance video.

7.2.2.1 Results

The **Tracking Time** metric $T_{Tracked}$ and the processing time metrics $\overline{T_p}$, $\overline{F_p}$, σ_{T_p}, $T_p^{(min)}$, and $T_p^{(max)}$ (defined in Section 7.1) have been utilised for this experiment.

In terms of the **Tracking Time** metric, the results are summarised in Figure 7.5. The results are very competitive with respect to the other tracking approaches. Over 15 tracking results, the proposed approach has the second best result on the apron videos, and the third best result for the road video. The worst result for the proposed tracking approach has been obtained for the building entrance video, with a fifth position. For understanding these results it is worthy to analyse the videos separately:

- **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 become totally occluded until the end of the sequence.

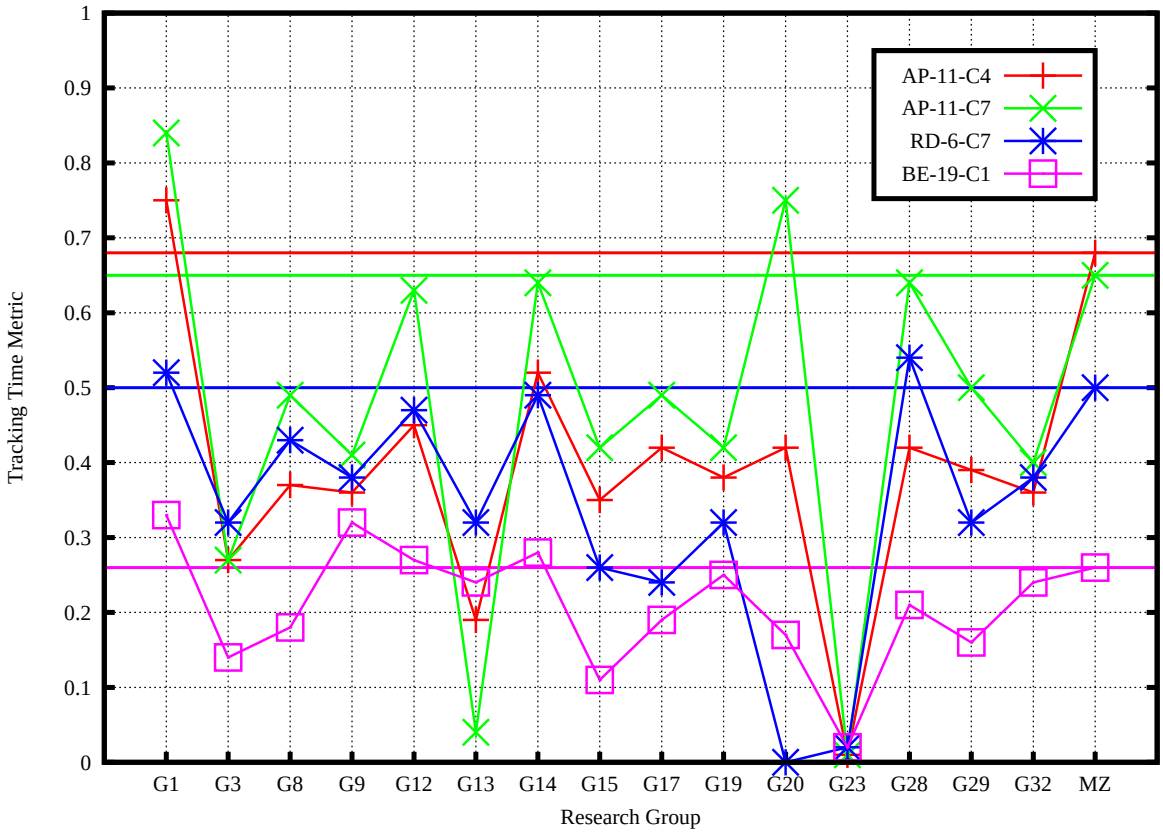


Figure 7.5: Summary of results for the Tracking Time metric $T_{Tracked}$ for the four analysed videos. The labels at the horizontal axis represent the identifiers for anonymous research groups participating to the evaluation, except for the **MZ** label, which represents the proposed tracking approach. Horizontal lines at the level of the obtained results for the proposed approach have been added to help in the comparison of results with other research groups.

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 7.6.

- **AP-11-C7:** For the second apron video, a Time Tracking metric value of 0.65 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 7.7.

- **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 in the scene, 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 reappear

Figure 7.6: Tracking results for the apron video **AP-11-C4**. A green bounding box bounding 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 bounding 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.

Figure 7.7: Tracking results for the apron video **AP-11-C7**. A green bounding box bounding 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 bounding 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.

in the scene. Nevertheless, several objects were appropriately tracked and even the lost objects by static occlusion were correctly tracked after the problem, showing a correct overall behaviour of the tracking approach. Results of the tracking process for this video are shown in Figure 7.8.

- **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 evolving in the scene. Results of the tracking process for this video are shown in Figure 7.9. As only four mobiles were evolving in the scene, a tracking error affected drastically the value of the Time Tracking metric. Moreover, several tracking errors have occurred analysing this video scene:

  - First, a person descending from the vehicle was not detected until she was completely separated from the vehicle. This problem is due that the tracking approach does not utilises appearance models that could be useful for coping with this type of situations.
  - Second, the same person leaving the vehicle, is almost immediately occluded by a second person evolving in the scene. This situation caused that the first person has been immediately lost, and that the track of the second person was lost because of the noise caused by the first person.
  - Finally, the second person arrives to the zone of the vehicle and in some moment the blobs of the person and the vehicle are merged, causing the person track to be lost again. This situation (as the previous lost track situation) was supposed to be solved by the tracking approach as it corresponds to an *over-segmented* object situation, as described in Section 5.3.4. Hence, this situation was an error of implementation of the tracking algorithm which has been corrected after the evaluation. This tracking failure is depicted in Figure 7.10.

The processing time performance of the proposed tracking approach has been also analysed in this experiment. Unfortunately, ETISEO project has not incorporated the processing time performance as one of its evaluation metric, thus it is not possible to compared the results obtained by the proposed approach. Table 7.4 summarises the obtained results for the processing time metrics. The results show a high processing time performance, even for the road video **RD-6-C7** ($\overline{F_p} = 42.7[frames/sec]$), which concentrated several objects simultaneously evolving in the scene. The fastest processing times for videos **AP-11-C7** ($\overline{F_p} = 85.5[frames/sec]$) and **BE-19-C1** ($\overline{F_p} = 86.1[frames/sec]$) are explained from the fact that there was a part of the video where no object was present in the scene, and because of the reduced number of objects. The high performance for the video **AP-11-C4** ($\overline{F_p} = 76.4[frames/sec]$) is because of the reduced number of objects.

The maximal processing time for a frame $T_p^{(max)}$ is never greater than one second, and the $\overline{T_p}$ and σ_{T_p} metrics show that this maximal value can correspond to isolated cases.

Figure 7.8: Tracking results for the road video **RD-6-C7**. A green bounding box bounding 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 bounding 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.

Figure 7.9: Tracking results for the building entrance video **BE-19-C1**. A green bounding box bounding 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 bounding 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.

Figure 7.10: Tracking failure at the building entrance video **BE-19-C1**. The top image shows the beginning of the problems between a tracked person and the tracked vehicle. Note that the 2D bounding box for the person is coloured red, meaning that it has not been classified at the current frame. Nevertheless, the coherence of data allows to keep the correct estimation of the 3D representation (yellow parallelepiped). The bottom image shows some few frames later when the track of the person is lost, and the blob is enclosing both the person and the vehicle.

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

Table 7.4: Evaluation of results obtained for both analysed video clips in terms of processing time performance.

7.2.2.2 Experiment Conclusion

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.

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

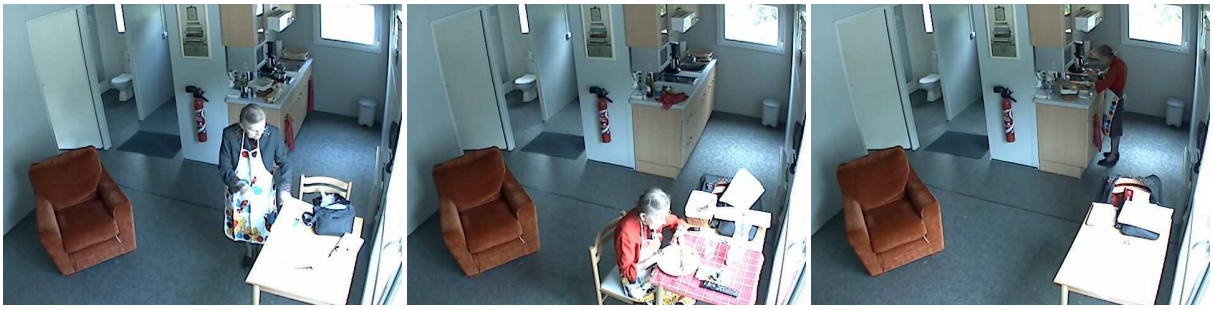
The road and building entrance videos have shown that there are still unsolved issues. The problems found in tracking the objects of the building entrance video highlight deficiencies in the implementation of the algorithm which have to be analysed. Also, both road and building entrance videos show the 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.

7.2.3 Evaluation of the Video Understanding Framework

The objective of this experiment is to evaluate different important aspects for the objectives of this thesis analysing a real world application. For this purpose two videos from GERHOME project for elderly care at home [GERHOME 2005], [Zouba et al. 2007]) are utilised. The video scene corresponds to an apartment with a table, a couch and a visible kitchen, as shown in Figure 7.11. The two utilised videos correspond to an elderly man (Figure 7.11(a)) and an elderly woman (Figure 7.11(b)), both performing performs tasks of everyday life as cooking, sitting, and having lunch. Each video sequence have a length of 40000 frames, giving a total of 80000 analysed frames and approximately two hours of video.

The walls of the apartment and the objects in the video scene (sofa, table, and kitchen)

(a)



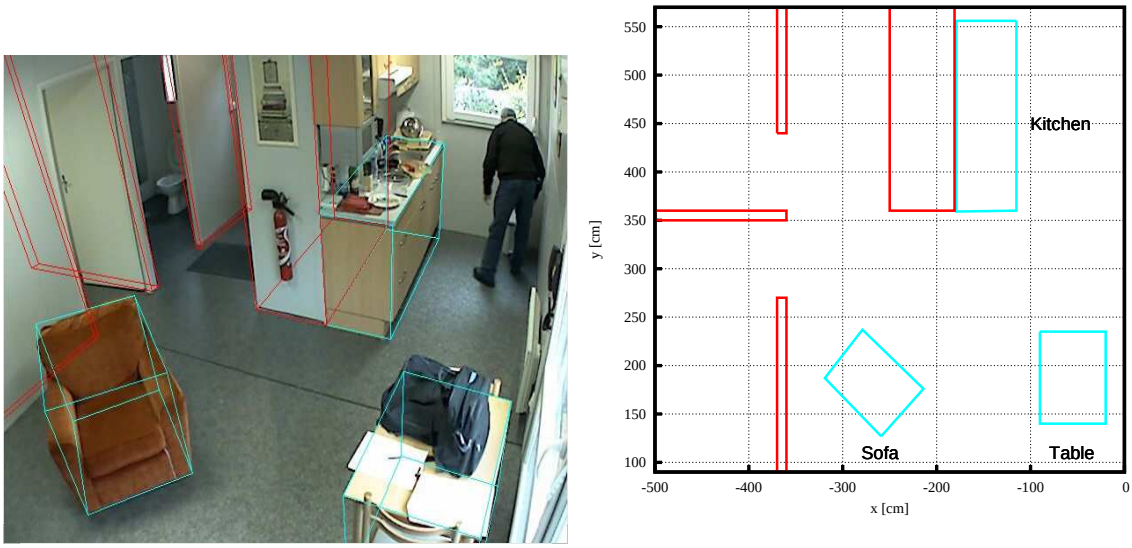
(b)

Figure 7.11: Video sequences selected from GERHOME project for elderly care at home. Figure (a) shows the analysed elderly man, while figure (b) shows the analysed elderly woman.

have been modelled in 3D, as depicted in Figure 7.12. The modelled objects allow to define 3D attributes accounting for the distance between the analysed person and these objects. All the experiments 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, one parallelepiped models have been pre-defined for a person, with standing and crouching postures modelled as follows (values are in centimetres):

- **Standing Posture:**
  $w \sim \mathcal{N}(\mu_w = 50, \sigma_w = 80),\ [min_w = 30; max_w = 100]$
  $l \sim \mathcal{N}(\mu_l = 60, \sigma_l = 40),\ [min_l = 20; max_l = 90]$
  $h \sim \mathcal{N}(\mu_h = 160, \sigma_h = 50),\ [min_h = 100; max_h = 200]$

- **Crouching Posture:**
  $w \sim \mathcal{N}(\mu_w = 60, \sigma_w = 60),\ [min_w = 30; max_w = 100]$
  $l \sim \mathcal{N}(\mu_l = 60, \sigma_l = 40),\ [min_l = 20; max_l = 90]$



(a) (b)

Figure 7.12: Modelled context of the apartment of GERHOME project. Figure (a) shows an image of the modelled scene, while figure (b) shows a top view of the scene showing the coordinates of the ground plane of the scene. Red coloured elements represent the walls in the scene, while cyan coloured elements represent the objects present in the scene.

$$h \sim \mathcal{N}(\mu_h = 110, \sigma_h = 50),\ [min_h = 50; max_h = 130]$$

The main experiment consists in first learning the hierarchy of states and events from the video of the elderly man. Then, the resulting hierarchy is used as input for the second video of the elderly woman, and the hierarchy is updated with the information generated by the analysis of the woman. Then, the results are evaluated in terms of the quality of learnt states on representing real world situations, and the results for learning contexts containing symbolic or numerical attributes is compared. Also, the influence of the reliability measures on guiding the learning process is analysed. This experiment is presented in Section 7.2.3.1.

Another experiment is performed for evaluating the processing time performance of the approach, and to establish the influence of the number of attributes in the computer time performance. This experiment is presented in Section 7.2.3.2.

Finally, another experiment is performed by analysing the same learning context, considering different acuity values for the analysed attributes. This experiment is presented in Section 7.2.3.3.

7.2.3.1 Exploring Learning Results

This experiment has three objectives:

- To illustrate the quality of representation of obtained state and event concepts of real situations.
- To illustrate the capability of the approach on bridging the gap between numerical and symbolic information.
- To evaluate the influence of reliability measures on guiding the learning process.
- To evaluate the capability of the learning approach on recognising the states and events associated to a mobile object.

This experiment first utilises all the 40000 frames of the the elderly man video to learn a hierarchy of states and events. Then, the first 30000 frames of the second video of the elderly woman are utilised for continuing the learning process, starting from the previously learnt hierarchy. Then, the last 10000 frames of the second video are learnt and used to analyse the recognised states and events for the elderly woman, as a way of validated the recognition capability of the approach.

Two learning contexts are utilised in this experiment, in order evaluate the capability of the approach on bridging the gap between numerical and symbolic information:

- **Purely Numerical Learning Context:** This learning context combines the 3D position attributes (x, y), the 3D parallelepiped attributes w, l and h, and the distances D_{table}, D_{sofa}, and $D_{kitchen}$ between the person and three objects present in the scene (table, sofa, and kitchen table). This distances have a maximal value of $100[cm]$, representing the limit for considering a person as far from the object. This learning context allows to relate position of the person, its posture in terms of the dimensions of the parallelepiped, and her/his position relative to objects present in the scene. Formally, this learning context is defined as:

Learning Context $Position - Dimensions - Distance$ {
Involved Objects: Person
Attributes:
Numerical x : 100 [cm]
Numerical y : 100 [cm]
Numerical w : 40 [cm]
Numerical l : 40 [cm]
Numerical h : 50 [cm]
Numerical $D_{kitchen}$: 50 [cm]
Numerical D_{table} : 50 [cm]
Numerical D_{sofa} : 50 [cm]
}

- **Numerical and Symbolic Learning Context:** This learning context combines the 3D position attributes (x, y), a symbolic attribute for standing and crouching postures of a person, and symbolic distance attributes $SymD_{table}$, $SymD_{sofa}$, and $SymD_{kitchen}$ between the person and three objects present in the scene (table, sofa, and kitchen table), considering three possible values: FAR for distances greater than 100[cm], $NEAR$ for distances between 50[cm] and 100[cm], and $VERY\_NEAR$ for distances lower than 50[cm]. As the previously defined context, this learning context also allows to relate position of the person, its posture, and her/his position relative to objects present in the scene, and it has been defined for evaluating the influence of numerical and symbolic attributes representing the same characteristic of a person. Formally, this learning context is defined as:

Learning Context $Position - Posture - SymbolicDistance$ {
Involved Objects: Person
Attributes:
Numerical x : 100 [cm]
Numerical y : 100 [cm]
Symbolic Posture : { $STANDING$, $CROUCHING$ }
Symbolic $SymD_{kitchen}$: { $VERY\_NEAR$, $NEAR$, FAR }
Symbolic $SymD_{table}$: { $VERY\_NEAR$, $NEAR$, FAR }
Symbolic $SymD_{sofa}$: { $VERY\_NEAR$, $NEAR$, FAR }
}

Representation of Real World Situations by the Hierarchical Structure

In order to illustrate the representation of real situations by the obtained hierarchies and the capability of the approach on bridging the gap between numerical and symbolic information, two situations found in the analysed videos are studied after processing the first 40000 frames of the elderly man, establishing a parallel between the obtained results for both analysed learning contexts and the real situation occurred in the scene.

After finishing the learning process for the first video, a hierarchy of 801 states and 33493 events has been learnt for the learning context $Position - Dimensions - Distance$ and a hierarchy of 505 states and 17955 events have been learnt for the learning context $Position - Posture - SymbolicDistance$.

The studied situations and their representations in the obtained hierarchies are now presented:

- **Going from the kitchen to the table:** This situation consists in the analysed person going from the zone near the kitchen, to the table zone, as depicted with the images shown in Figure 7.13. In the hierarchy obtained from the learning context

(a) (b) (c)

Figure 7.13: Situation where the person goes from the kitchen to the table. Figures (a), (b), and (c), in this order, describe the way this situation occurs in the scene.

Position − Posture − SymbolicDistance, the situation is described by the states and events depicted in Figure 7.14.

Note that three states representing each of the displayed images in Figure 7.13. The

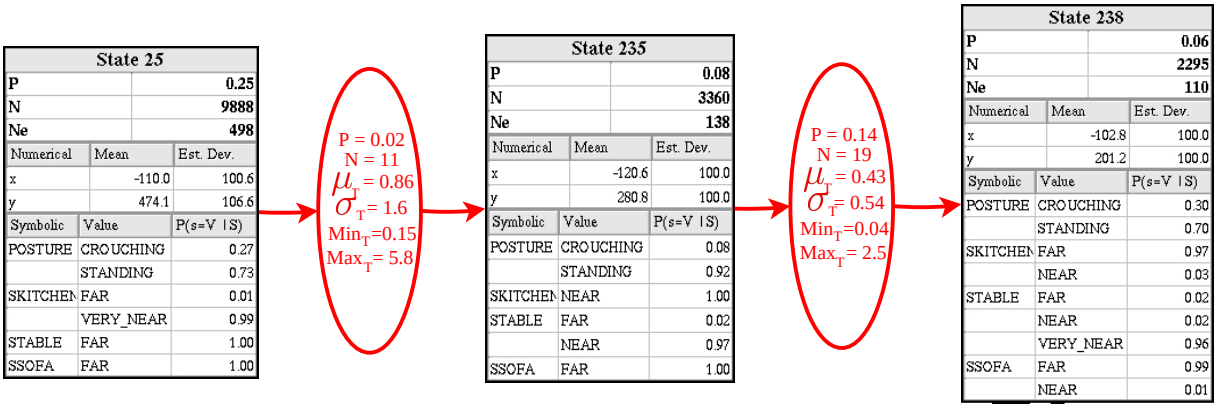


Figure 7.14: Representation of the situation where the person goes from the kitchen to the table in the hierarchy obtained for the learning context *Position − Posture − SymbolicDistance*.

probability of occurrence of the first state 25 is 9888/40000 = 0.25, as the elderly man spends a long time in the kitchen zone. Note that this state is well describing the fact that the man is all the time very near of the kitchen, also showing that at this state the man is not standing all the time, but also crouching approximately a quarter of the total of time spent at this state.

For the same reason that the elderly man spends a long time in the kitchen zone, the events generated for this state are concentrated between states occurring in the kitchen and the conditional probability of the first event is very low (0.02), giving a marginal probability of occurrence for the event of $0.25 \cdot 0.02 = 0.005$. The second state represents an intermediate passage zone near the kitchen and the table, where the person passes most of the time standing. Note that the time staying at the previous state displayed in the second event, denotes also that the second state is just a transition zone between the kitchen and the table as its mean value of 0.43 seconds indicates that the person normally does not stop at this zone.

The conditional probability of the second event is higher (0.14), giving a marginal probability of occurrence for the event (starting from state 25) of $0.25 \cdot 0.02 \cdot 0.14 = 0.0007$, showing that the occurrence of this whole situation is quite infrequent. The third state represents the position very near the table. Here, the person has a crouching posture approximately a third of the total time spent in this state.

In the hierarchy obtained from the learning context *Position − Dimensions − Distance*, the situation is described by the states and events depicted in Figure 7.15.

Note that now just two states represent the same situation. The first state

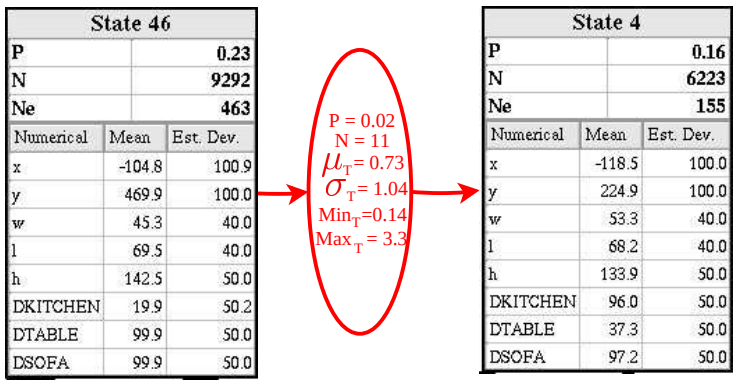


Figure 7.15: Representation of the situation where the person goes from the kitchen to the table in the hierarchy obtained for the learning context *Position − Dimensions − Distance*.

represents almost the same situation represented by the first state for the hierarchy associated to the learning context $Position - Posture - SymbolicDistance$, showing a distance to the kitchen of 19.9$[cm]$. The dimensions of the parallelepiped show intermediate values compared to the pre-defined models of postures, which seems to also represent the fact that both standing and crouching postures occur in this state.

The probability of occurrence of the first state 46 is 9292/40000 = 0.23, a very similar probability compared to the first state of the other representation.

The event of this representation also presents a similar behaviour compared with the first event of the other representation as the conditional probability is also (0.02), giving a marginal probability of occurrence for the event of $0.23 \cdot 0.02 = 0.0046$. The second and last state 4 represents the arrival of the man to the table, presenting a value of distance to the table 0f 37.3[cm]. The dimensions of the parallelepiped also show intermediate values between those of the pre-defined models of postures. More detailed versions of the attributes are available for the children of the State 4.

- **Crouching and then standing at the table:** This situation consists in the analysed person passing to a crouching posture and then returning to the standing posture, at the zone near the table, as depicted with the images shown in Figure 7.16. In the hierarchy obtained from the learning context $Position - Posture -$

(a) (b) (c)

Figure 7.16: Situation where the person passes to the crouching posture and then returns to the standing posture, near the table. Figures (a), (b), and (c), in this order, describe the way this situation occurs in the scene.

SymbolicDistance, the situation is described by the states and events depicted in Figure 7.14.

Note that three states representing each of the displayed images in Figure 7.16. The probability of occurrence of the first state 131 is not very high 0.04, as the elderly man does not spend a long time in the table zone, compared with the time spent in the kitchen zone. This state is describing that the man is all the time very near of the table at a standing posture.

The first event has a high conditional probability (0.4), giving a marginal probability of occurrence for the event of $0.04 \cdot 0.4 = 0.016$. The second state represents a person still very near of the table but now in a crouching posture.

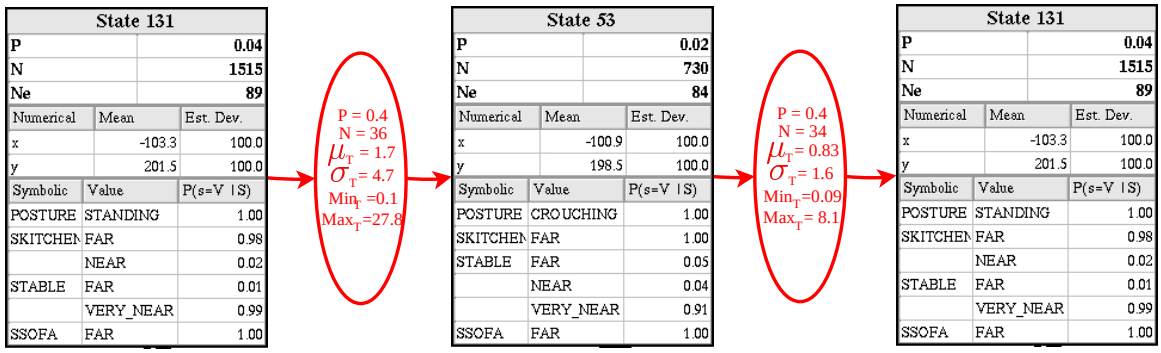


Figure 7.17: Representation of the situation where the person passes to the crouching posture and then returns to the standing posture in the hierarchy obtained for the learning context $Position - Posture - SymbolicDistance$.

The conditional probability of the second event is also high (0.4), giving a marginal probability of occurrence for the event (starting from state 131) of $0.04 \cdot 0.4 \cdot 0.4 = 0.0064$, showing that the occurrence of this whole situation is less infrequent than the first situation. The third state represents the return to the standing posture.

The high number of event transitions between these states, compared with the observed video, together with a high difference between the mean and maximal staying time of the states, highlights a problem inherent to the discretisation process to obtain symbolic attributes: the error is amplified. Here the situation can be that the person, because of errors in the estimation of the dimensions due to a bad segmentation, gave as result the wrong posture, forcing wrong transitions between both states.

In the hierarchy obtained from the learning context $Position - Dimensions - Distance$, the situation is described by the states and events depicted in Figure 7.18.

For this situation, a good representation of three states has also been found. The main difference with the symbolic representation is the number of events between these states which approaches to the observed number of events in the elderly man video. This result denotes a behaviour of the numerical attributes which is more tolerant to errors.

Note that the attributes influence is important in the structure of a hierarchy of states and events. For the same situation both hierarchies are able to represent it appropriately, but the results are far from being identical. As a valid representation can be ground for each representation, the gap between symbolic and numerical information is correctly bridged for the presented situations.

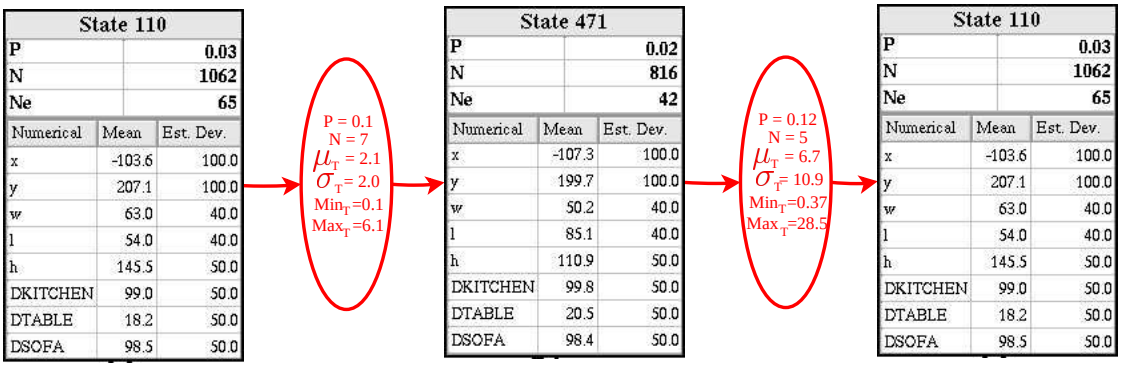


Figure 7.18: Representation of the situation where the person passes to the crouching posture and then returns to the standing posture in the hierarchy obtained for the learning context $Position - Dimensions - Distance$.

The numerical representation gave a more accurate description of the situation, but the symbolic representation is able to give a representation more interpretable for a human, as the values of the symbolic attributes are defined precisely for this purpose. Also, the numerical attributes will allow a hierarchical representation with less errors in the state transitions, as the error of discretisation is avoided.

Explaining Event Recognition Results

One of the most important aspects in the evaluation of the proposed learning approach is the capability of automatically recognising real world situations utilising the learnt event hierarchy. For this purpose, an experiment has been performed which consists in recognising in which events the elderly woman is involved, considering the hierarchy of states and events trained with the first 68000 frames (40000 corresponding to the elderly man video, plus 28000 from the beginning of the elderly woman video) as input for performing the recognition of the event instances. The learning context $Position - Posture - SymbolicDistance$, previously defined, has been utilised for this experiment.

The experiment considers 2000 frames from the elderly woman video for event recognition. The evolution of the elderly woman in the sequence is depicted in Figure 7.19. The recognised events correspond to those events detected in the learning process and associated to the corresponding contextualised object, filtered by a pre-defined temporal stability threshold of $1.0[s]$ for filtering events possibly induced by attribute value changes due to noise in the video.

The recognition process has obtained as result 45 detected events with a duration higher than $1[s]$. From these events, 21(46.7%) were induced by attribute changes due to

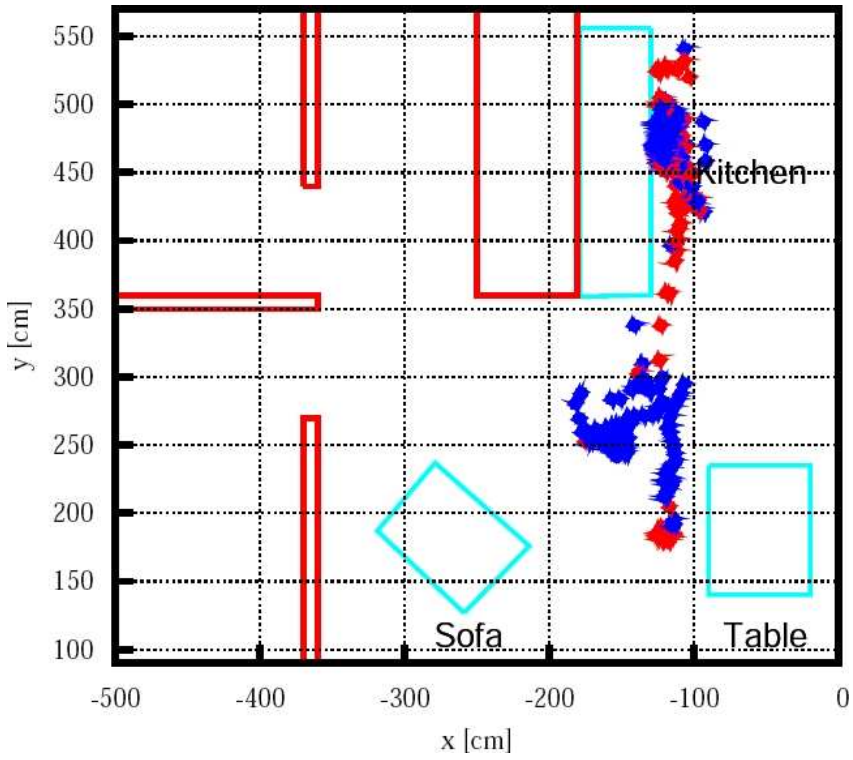


Figure 7.19: Top view of the input for the experiment consisting of the position in the ground plane of the apartment scene and the posture of the elderly woman, in the video sequence utilised for evaluating the event recognition capability of the learning approach. The colour of the dots represents the occurrence of a specific human posture: red dots represent that the woman is in a crouching posture, while blue dots represent the standing posture.

a bad segmentation, while 25(53.3%) were representing real events. The two events with the longest staying time of its starting state are detailed.

- **Recognised Event: Standing from the Table Zone.**

  This event has been detected when the elderly woman has begun to stand from the chair. With the available information it is not possible to say that the elderly woman was sitting in the chair, but just that she has changed her posture after a stable period being in a crouching posture. This event is depicted in Figure 7.20.

- **Recognised Event: Start Going to Kitchen Zone.**

  This event has been detected when the elderly woman has begun to walk to the kitchen, after watching the television (the television is not visible from the camera view). With the available information it is not possible to say that the woman has been watching the television, but just that she has changed her position in a noticeable extent after a stable period being standing approximately in the same position, near the table and the sofa. This event is depicted in Figure 7.21.

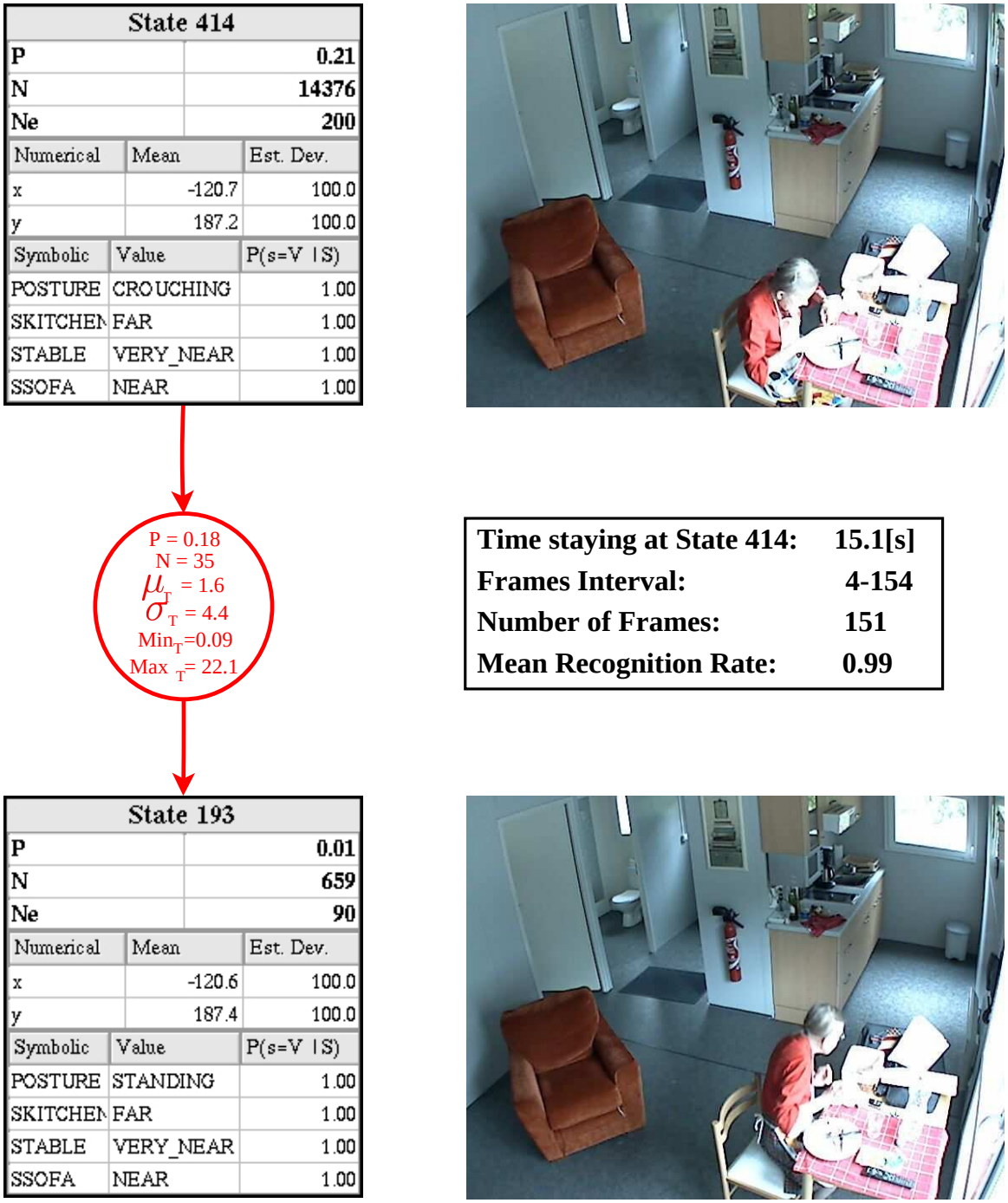


Figure 7.20: Event *standing from the table zone* recognised in the video sequence utilised for evaluating the event recognition capability of the learning approach. The event is coloured red. The right-top image corresponds to the video frame found in the middle of the interval staying in the starting state of the event, while the right-bottom image corresponds to the video frame which has caused the occurrence of the event. The information in the black square, corresponds to the information about the starting state obtained in the moment of occurrence of the event.

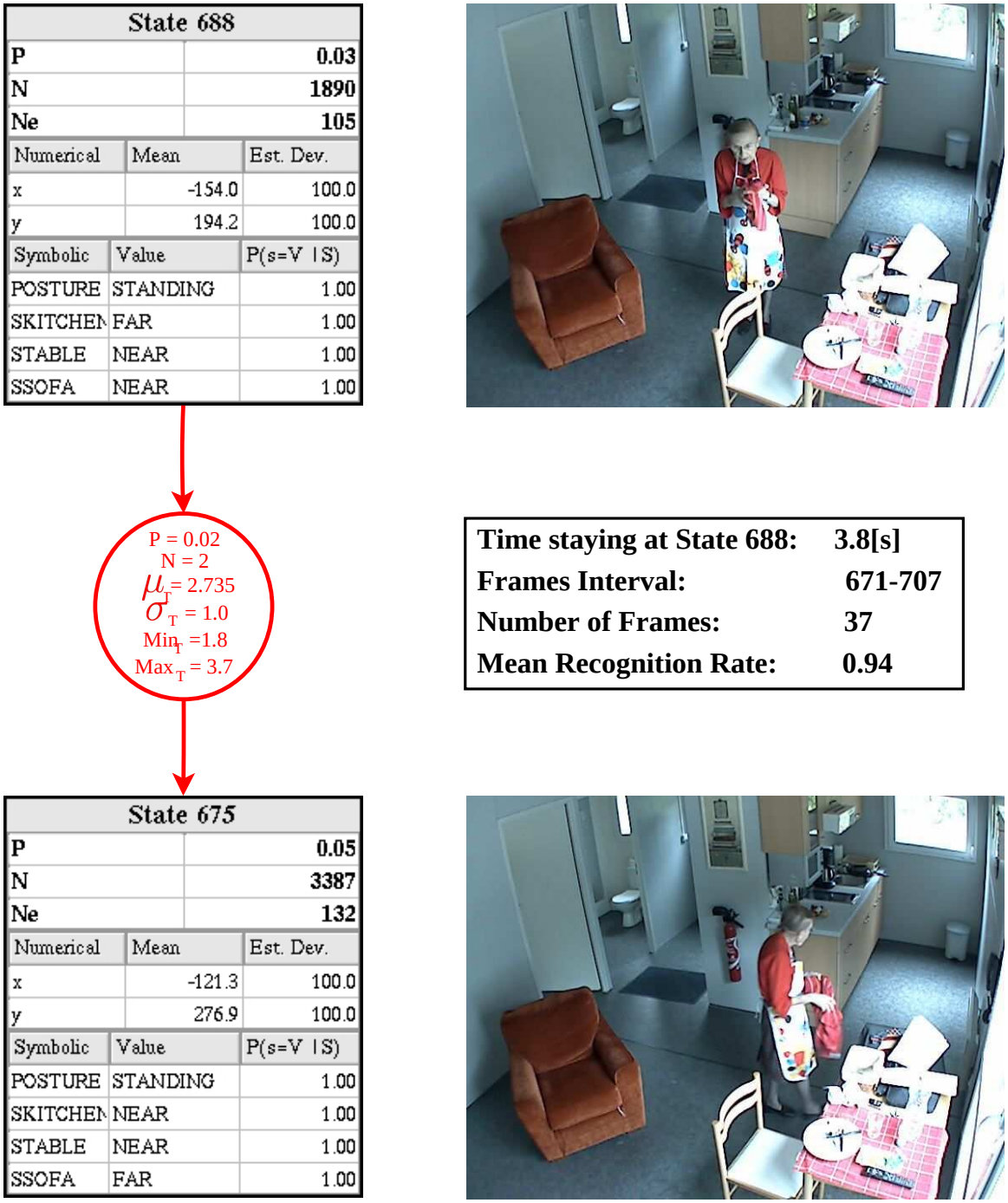
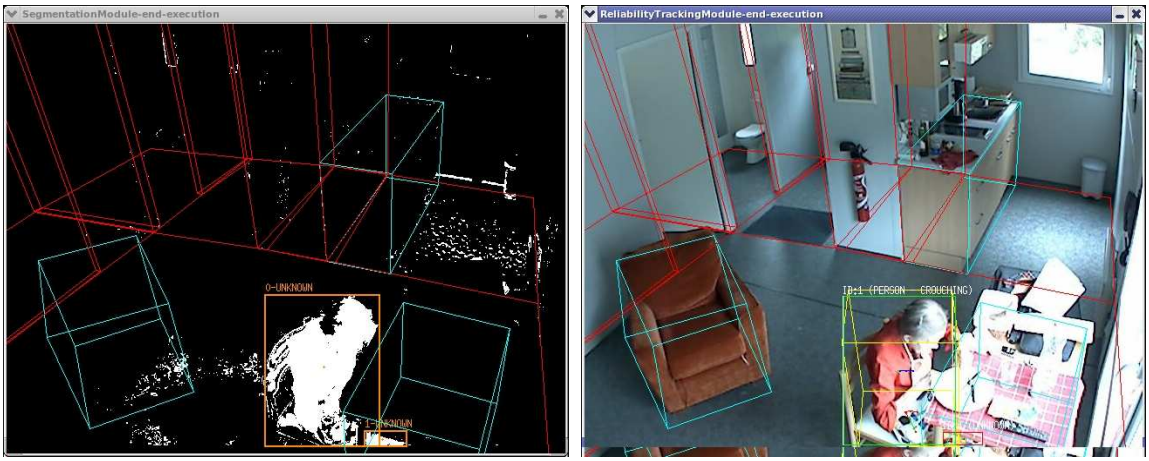


Figure 7.21: Event *start going to the kitchen zone* recognised in the video sequence utilised for evaluating the event recognition capability of the learning approach. The event is coloured red. The right-top image corresponds to the video frame found in the middle of the interval staying in the starting state of the event, while the right-bottom image corresponds to the video frame which has caused the occurrence of the event. The information in the black square, corresponds to the information about the starting state obtained in the moment of occurrence of the event.

The obtained recognition rate metric P_r values (referenced in the Figures 7.20 and 7.21 as *Mean Recognition Rate*) show that the instances are appropriately represented by the attribute models of the states in the hierarchy. The results show that the system is able to recognise real events occurring in the scene. This video presented a real challenge as the segmentation results were always very noisy, producing a lower performance of every task of the video understanding process. This situation has caused a lower number of correctly recognised events. The system is able to manage noisy data in several situations, but when this noisy data become persistent and consistent, the situation is assumed as normal and the framework fails to manage the wrong data. Figures 7.22, 7.23, and 7.24 show segmentation data of different quality in order to explain the capability of the framework on handling noise.

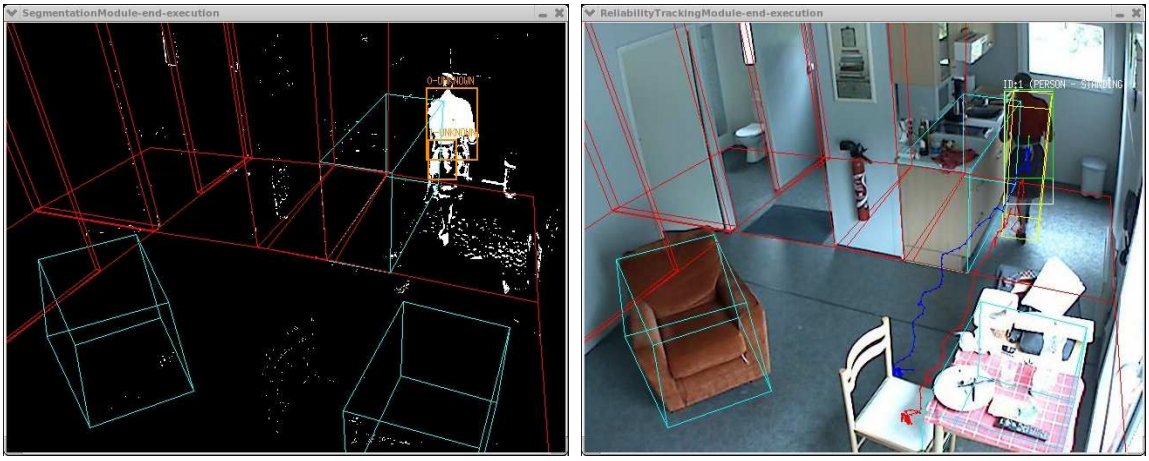
In the presence of segmentation of good quality (Figure 7.22), the recognition process



(a) (b)

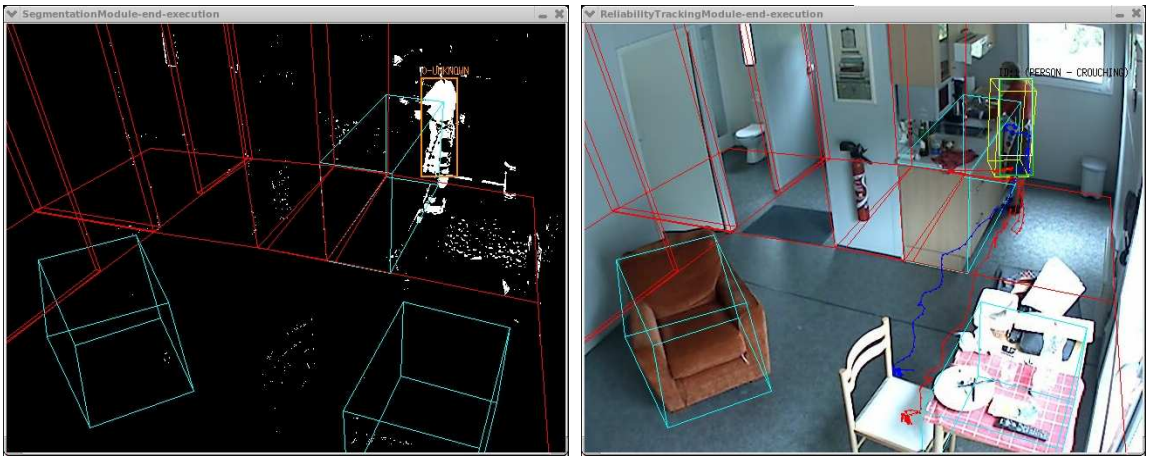
Figure 7.22: Segmentation data of good quality, producing the appropriate results by the tracking approach. Figure (a) shows the result of the segmentation process. Figure (b) shows a correct result of the tracking process, utilising as input the segmented region shown in Figure (a). In Figure (a), the moving pixels are coloured in white, while the blob surrounding the moving region is coloured in orange. In Figures (b), the white bounding box bounding 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. Next similar figures follow the same colour schema.

can be able of recognise events of longer time duration, and to produce a minimal number of recognition errors. In the presence of bad quality segmentation (Figure 7.23), the framework is able to cope with this noise and to provide the appropriate input to the recognition process. If these noisy data are persistent and consistent in time, the framework will interpret that the moving region is sufficiently stable to not be considered as noise, producing a failure in terms of estimation of the attributes or the detection of



(a) (b)

Figure 7.23: Noisy segmentation data of poor quality, not having consequences in the results obtained by the tracking task. Figure (a) shows the poor result of the segmentation process. Figure (b) shows a correct result of the tracking process, utilising as input the poorly segmented regions shown in Figure (a).



(a) (b)

Figure 7.24: Noisy segmentation data persistently and consistently occurring, which induces the tracking approach to erroneous results. Figure (a) shows the result of the poor segmentation process. Figure (b) shows the obtained result, where a standing person is detected as crouching out of the zone of interest of the scene, utilising as input the poorly segmented regions shown in Figure (a).

an object inexistent in the real world, which triggers the recognition of erroneous events, as the event described in Figure 7.25. This erroneous event shows the situation where the segmentation persistently (15 frames) gives as result wrong data (sub-segmented in this case). The tracking task utilises this wrong data and checks that it is coherent in time, and that can be a person crouching in a farther position compared with the position of the real person. After, the person is also erroneously detected as a person standing in the far position, which produces the recognition of an event with an acceptable amount of time staying at the starting state.

In conclusion, the proposed video understanding framework is able to recognise events even in presence of noisy data, but the level of noise can not be excessive. From this fact, it is very important to point as future work the exploration of different segmentation techniques and how the reliability of the obtained data can be estimated.

Reliability versus No Reliability

In this experiment, the influence of the reliability measures on guiding the learning process can also be studied. For this purpose, tests have been made for the same learning contexts defined for this experiment, one considering The evolution of the states number, and events number metrics comparing between contexts considering and not considering reliability measures for guiding the learning process. The results for both learning contexts are summarised in Figure 7.26

The results show that the utilisation of reliability measures in both learning contexts drastically increases the complexity of the hierarchical representations as a higher number of states is generated. This behaviour is explained from the fact that reliability measures diminish the influence of noisy information in the state attributes computation, producing lower standard deviation values for numerical attributes and less erroneous values considered for symbolic attributes, resulting in a better discrimination between state concepts which at the same time induces the creation of a higher number of new concept states.

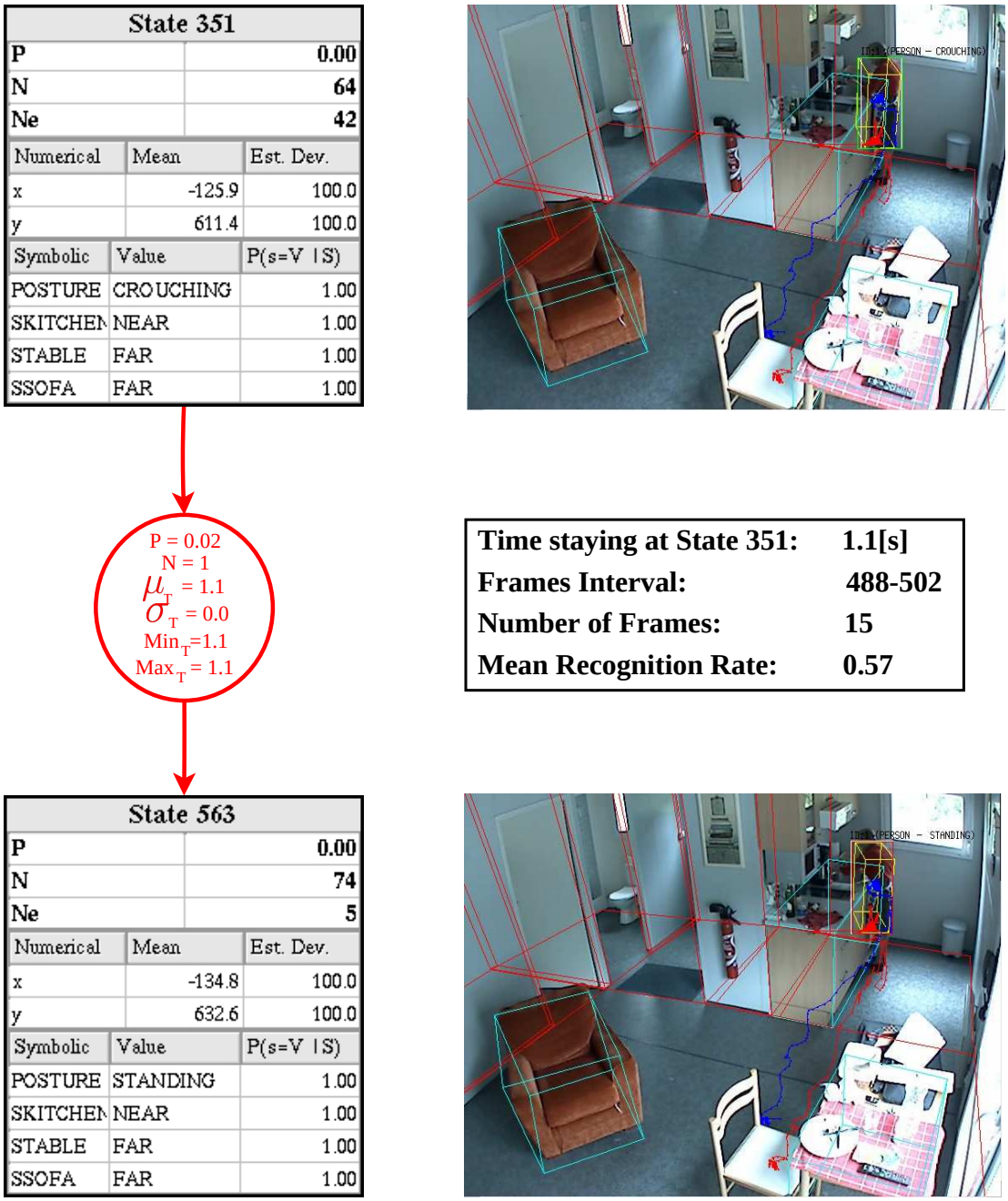


Figure 7.25: Erroneous event detection, caused by noisy data persistently obtained by the segmentation task. The information in the black square, corresponds to the information about the starting state obtained in the moment of occurrence of the event.

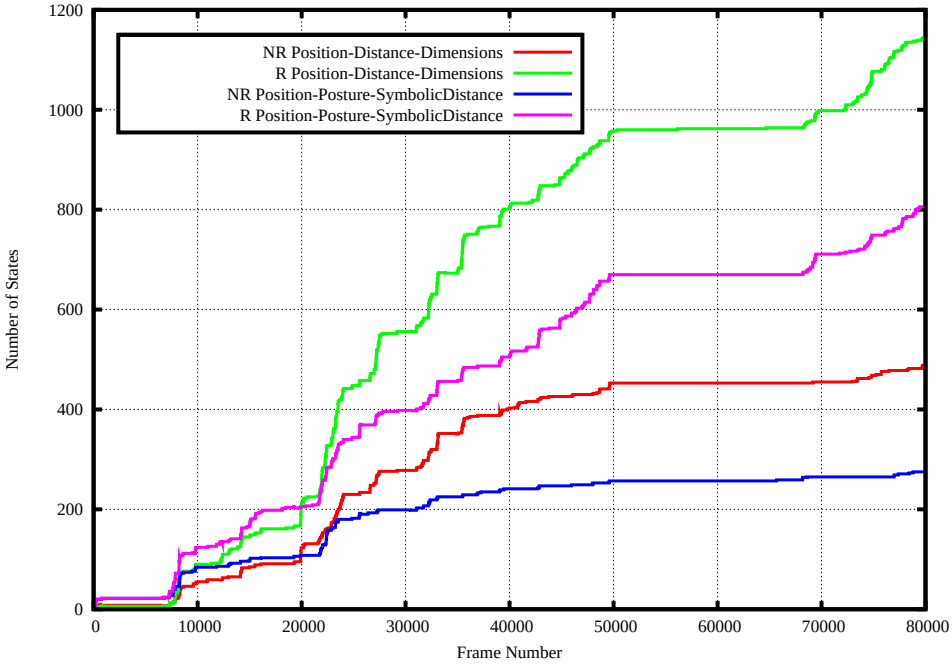


Figure 7.26: Evolution of the number of hierarchy states in time, considering or not considering the utilisation of reliability measures for both studied learning contexts. NR stands for not considering reliability, while R for considering it.

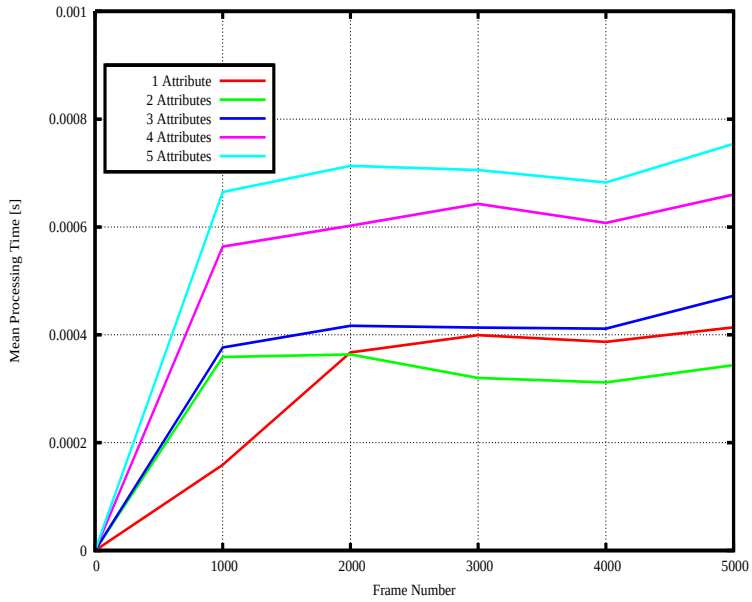
7.2.3.2 Processing Time Performance

In order to analyse the processing time performance, five learning contexts have been evaluated for the first 5000 frames of the elderly man video. The experiment consists in augmenting the number of attributes by one for each learning context, to be also able of evaluating the influence of the number of attributes in the processing time performance of the approach.

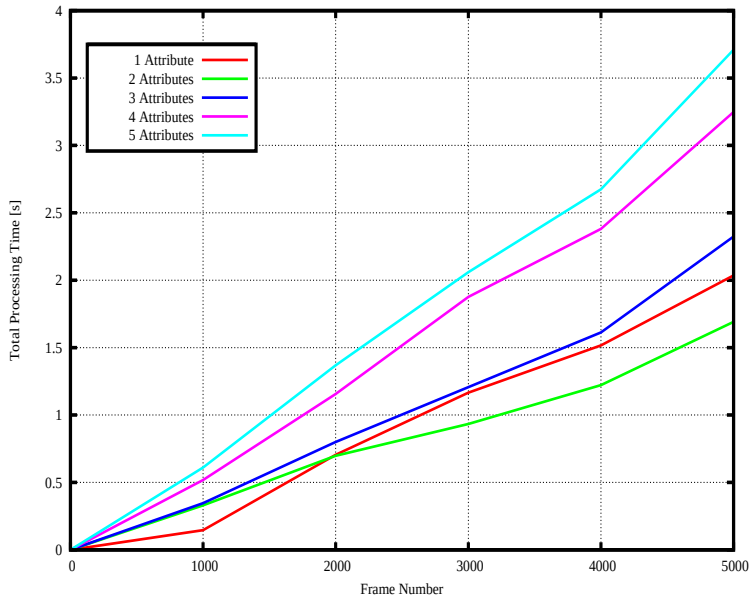
The five considered attributes are the numerical attributes x, y, w, l, and h, obtained from the 3D parallelepiped representation. The results of this experiment are summarised in Figure 7.27.

The results of this experiments show a high computer time performance of the learning approach with a mean processing time of 0.75 milliseconds per frame, or a frame rate of 1326 frames per second, for the largest learning context (five attributes), showing the real-time capability of the learning approach.

The evolution of the processing time performance versus the number of attributes time shows a nearly linear behaviour. As expected, the influence of the number of attributes increases the mean processing time, and it seems that this relation is not linear, probably logarithmic, as depicted in Figure 7.28 for 5000 frames.



(a)



(b)

Figure 7.27: Evolution of the processing time performance over 5000 frames, for learning contexts with 1, 2, 3, 4, and 5 attributes. Figure (a) displays the results in terms of total processing time, while figure (b) shows the results in terms of mean time per frame.

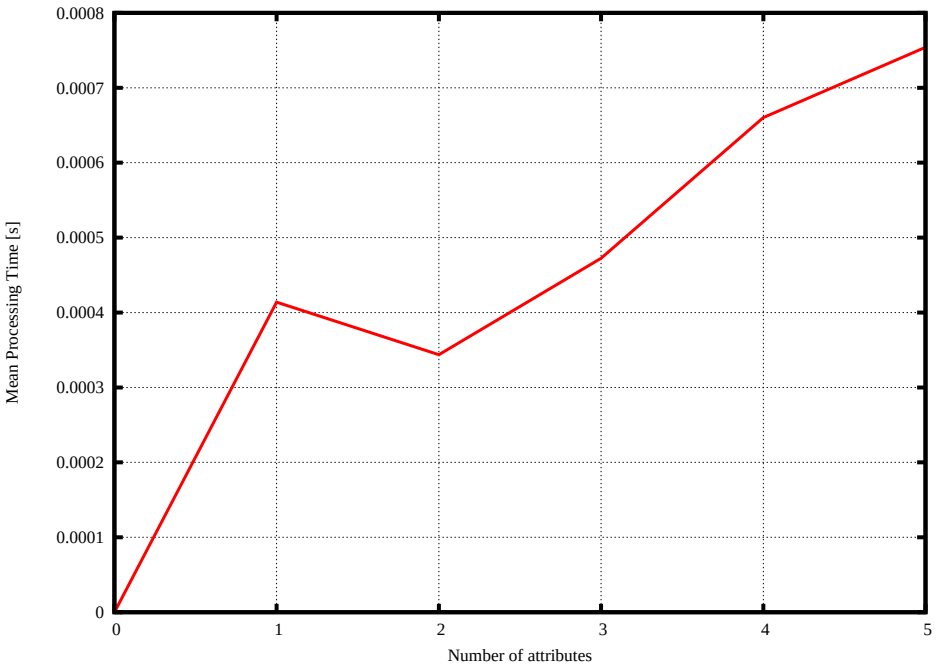


Figure 7.28: Evolution of the mean processing time while augmenting the number of attributes to be learnt, for 5000 processed frames.

7.2.3.3 Influence of the Acuity

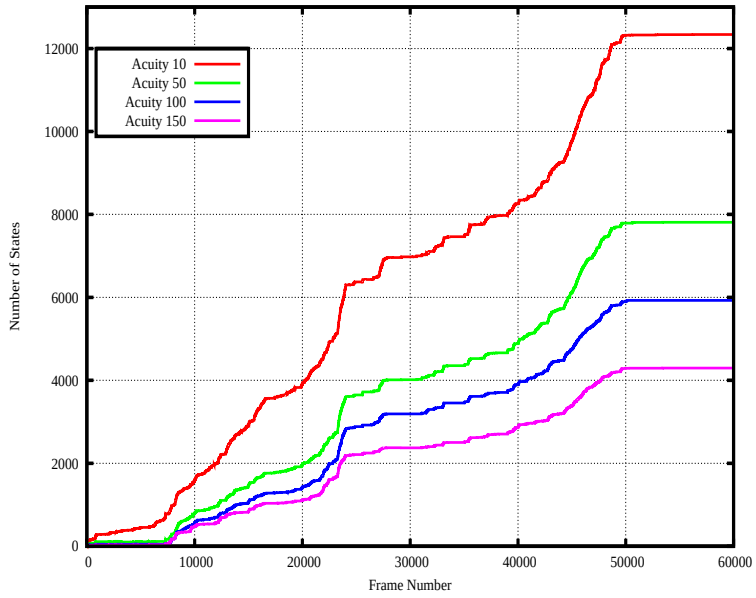
In order to evaluate the influence of the acuity of numerical attributes in the resulting hierarchy of states and events, one learning context is considered, adjusting the acuity values for five different values. The considered learning context is described below:

Learning Context {
Involved Objects: Person
Attributes:
Numerical x : A [cm]
Numerical y : A [cm]
Numerical V : A [cm]
}

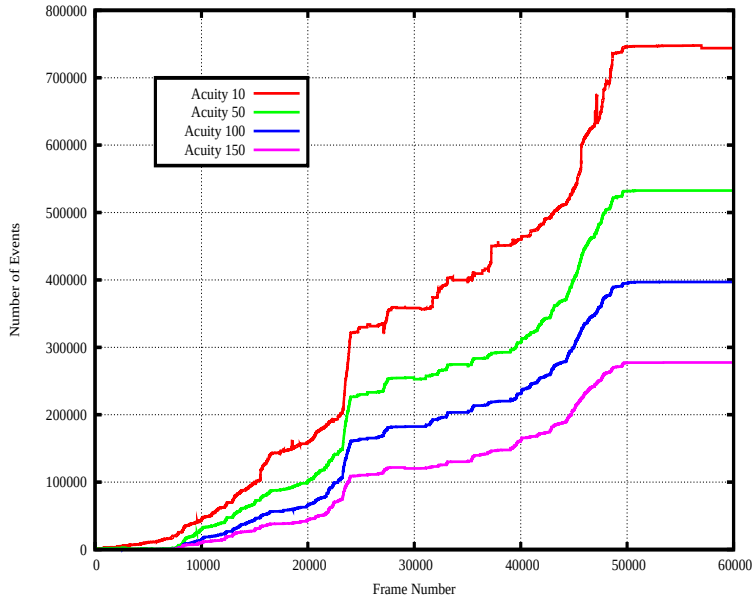
, where (x, y) corresponds to the position of the person in the ground plane of the scene, V corresponds to the velocity magnitude of the person, and A corresponds to the acuity of all attributes, with $A \in \{10, 50, 100, 150\}$.

The number of states and number of events metrics are evaluated for the 40000 frames of the elderly man video and the first 20000 frames of the elderly woman video. The results are summarised in Figure 7.29.

Results show that different acuity values produce a similar behaviour in the evolution of the number of states and events of a hierarchy, but also shows that a lower acuity value



(a)



(b)

Figure 7.29: Evolution of the number of state and number of events metrics over 60000 frames, for a fixed learning context with numerical attributes acuity value of 10, 50 100, 150. Figure (a) displays the results for the number of states metric, while figure (b) shows the results for the number of events metric.

induces the creation of a higher number of states and events. This is due to the fact that higher acuity values make the *cutoff* criteria more constraining, as higher differences in attribute values are considered as non significant. Also, the acuity value influences the decision of the instance incorporation process, giving a higher chance to the best state criteria to incorporate the instance, in despite of of creating a new state.

The evolution of the number of states versus the acuity value seems to show a negative exponential behaviour, as depicted in Figure 7.30 for 60000 frames.

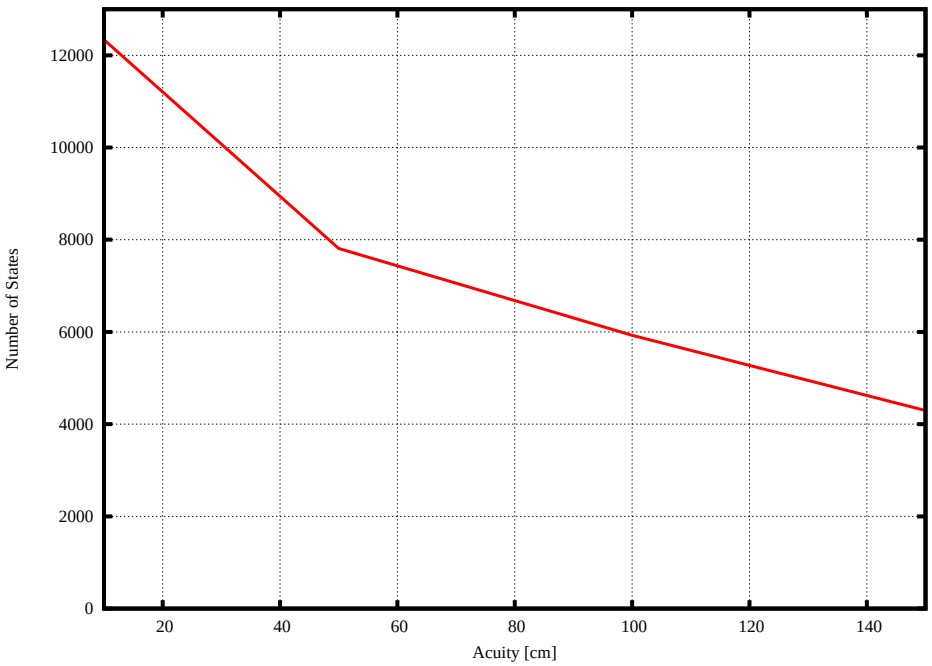


Figure 7.30: Evolution of the mean processing time while augmenting the acuity value, for 60000 processed frames.

7.2.3.4 Experiment Conclusion

The presented experiment for evaluating the incremental learning framework in a real world application has resulted in the following main conclusions:

- The framework seems to be able of bridging the gap between numerical and symbolic information, giving appropriate representations of real world situations.
- The framework is able to recognise events occurring in real world videos, even if the received information is noisy. The level of noise can not be excessive and other segmentation techniques must be tested to improve the event recognition performance.
- The utilisation of reliability measures for guiding the learning process induces more discriminative states by reducing the influence of noisy instance attributes.

- The incremental nature of the learning approach ensures a real-time performance of the learning process.
- The relation of the number of attributes and the number of states metric has nearly linear behaviour, ensuring the scalability of the system.
- The relation of the acuity value and the number of states metric seems to have negative exponential behaviour, highlighting the importance of fixing acuity values coherent with the interest of the user or the scale of the attribute. This concern allows to avoid the explosion in the complexity of the obtained hierarchy and to have a better processing time performance.

7.3 Conclusion from Experiments

This chapter has shown the evaluation of the complete proposed video understanding framework, by also evaluating classification nd tracking tasks.

The classification task has shown its potential to be applied in real world applications.

The tracking task has shown to be very competitive in terms of quality of solutions, compared with other tracking approaches evaluated over benchmark videos publicly accessible. The approach has also shown a processing time performance near real-time.

The event learning approach has shown its capability of representing real world situations in an appropriate way, being also able of bridging the gap between numerical and symbolic information. Its event recognition capability makes this approach interesting for several applications, as automatic human behaviour recognition and the detection of abnormal situations. From the results obtained in event recognition, it can be concluded that is compulsory to integrate other segmentation techniques in order to ensure a minimal quality of the input data. Also, it will be important to study how reliability measures can be utilised to detect the level of noise in the obtained moving regions.

The learning process has shown that can have a real-time processing time performance and the obtained hierarchical representation can be useful as input for other higher-level applications in video understanding, as video data mining [Benhadda et al. 2007], automatic or interactive image and video retrieval [Le et al. 2008], and semantic recognition of composite events [Vu et al. 2003], [Zouba et al. 2007].

The processing time performance of the learning approach has shown its capability of performing in real-time.

Next Chapter 8 presents the conclusion and future work of this thesis.

Chapter 8

Conclusion

The goal of this thesis on proposing a video understanding framework for general event learning addressing real world applications has been achieved. A new video understanding framework has been proposed, which is able to incrementally learn general descriptions of the events occurring in a video scene. The incremental nature of the event learning process is well suited for real world applications as it considers the incorporation of new arriving information with a minimal processing time cost. Incremental learning of events can be useful for abnormal event behaviour recognition and to serve as input for higher level event analysis.

Addressing real world applications also implies that the video understanding framework 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 understanding 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 video understanding framework involves a complete framework for event learning including video frame segmentation, object classification, object tracking, and event learning tasks. This approach have proposed an automatic bridge between the low-level data obtained from objects evolving in the scene and higher level information which considers the temporal aspect.

Next chapters present the conclusion for each task of the video understanding framework. Section 8.1 presents the conclusion for the proposed object classification method. Then, Section 8.2 concludes about the proposed object tracking approach. Next, Section 8.3 presents the conclusion for the new incremental event learning method. Finally, Section 8.4 presents the limitations and future work for the video understanding framework.

8.1 About Object Classification

The proposed classification method is suitable for real world applications for several reasons:

- The method has shown a high processing time performance for scenarios of moderated complexity.
- The classification results are highly independent from the camera view and orientation of the object, having an appropriate flexibility for been utilised in diverse real world applications.
- The method is capable of coping with even severe static occlusion situations.
- The approach proposes methods for disambiguation between several geometrically plausible alternatives.
- The parallelepiped model utilised by the classification approach is capable of representing a large variety of objects, even those which change their posture, with acceptable 3D attribute values. This simple model also allows users to easily define new mobile objects that could be present in the scene.
- Visual reliability measures have been proposed for the parallelepiped model attributes measuring the degree of visibility of these attributes. These measures have been used by the proposed tracking approach to guide the estimation of object features utilising the most reliable information. The estimation of these measures is the first step for estimating the reliability of the information in the whole video understanding framework.

The estimated 3D attributes for the proposed parallelepiped model have allowed the tracking approach to perform a better filtering of hypothesis by evaluating the coherence of these attributes in time.

The evaluation results have shown that the classification approach can even be interesting by itself.

8.2 About Object Tracking

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 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. The dynamic occlusion problem resolution capability has shown limitations that are described in Section 8.4. 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:

- The proposed tracking approach explicitly cooperates with the object classification process, by guiding the classification process using the previously learnt mobile object attributes. This way, the tracking process is able to indicate a starting point and the bounds of search for the parallelepiped attributes to be found by the classification approach. This cooperation scheme allows a considerable reduction in the processing time dedicated to 3D classification. As mobile information can become more reliable as more visual evidence is available, the cooperation scheme can be also considered to improve its quality in time, as more reliability implies a more accurate mobile dynamics model and less variability of mobile attributes, establishing tighter bounds to the search space.

- When a mobile object pass to *ensure* mode, even a better performance can be obtained by the 3D classification process, as the parallelepiped is estimated just for one object class. 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.

- The determination of the involved blob sets allows to control the number of possible blob associations for a mobile object and to separate the tracking problem into sub-problems according to the proximity of the blobs representing the visual evidence. Then, the involved blob sets determination presents a two-fold contribution to the early control of the combinatorial explosion, as less possible associations per mobile and less related mobiles per tracking sub-problem imply the immediate reduction in the number of hypotheses to generate, contributing to the improvement of the

processing time performance.

- The new proposed hypothesis updating process have been oriented to optimise the estimation of the updated hypothesis set, in order to obtain the most likely hypotheses avoiding to generate unlikely hypotheses that must be eliminated later. The new method for generation of the mobile tracks utilises a similar principle, generating the initial solution nearest to the estimated mobile attributes, according to the available visual evidence, and then generating the other mobile track possibilities starting from this initial solution. This way, the generation is focused in optimising the processing time performance by warrantying the generation of good quality solutions, instead of generating all the possible combinations and pruning the solutions with bad quality.

- Even if the hypothesis updating process is focused in generating the minimal possible number of hypotheses, the processing load for the next frame can be reduced by filtering redundant, not useful, or unlikely hypotheses.

- Finally, the split process for hypothesis sets, represents another mechanism to improve the processing time performance as it immediately reduces the number of mobiles in a same hypothesis set, generating different hypothesis sets, which can be treated as separated tracking sub-problems.

The estimation of reliability measures in the tracking approach has a direct impact in the learning task as the tracking approach gives to the event learning task the necessary elements for determining the most valuable object attribute information to be learnt.

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.

8.3 About Event Learning

The proposed event learning approach has been conceived to be able to learn state and event concepts in a general way. The definition of multiple learning contexts endows the learning process with a flexible mechanism for learning events occurring in a video scene. Depending on the availability on tracked object features, the possible combinations for learning contexts is enormous. The attributes already proposed by the object tracking approach give a sufficient flexibility to explore a large variety of scenarios. Anyway, users can always define more object attributes, by either combining existing attributes or by creating new ones from new object descriptors.

For performing the learning process a new incremental event learning algorithm called MILES (Method for Incremental Learning of Events and States) have been proposed. The incremental nature of MILES, allows to obtain a learning performance that can be utilised in on-line learning.

The main contribution of MILES is the utilisation of incremental concept learning models to learn the states as a hierarchy of concepts and to extend the incremental concept learning hierarchy to learn the events as first order temporal relations between the learnt states. The extension to event learning has implied the redefinition of the existing merge and split hierarchy operators.

Another contribution is the way of utilising the concepts of *cutoff* and *acuity*. Before, these concepts were treated as general parameters for an incremental concept learning algorithm, and now the acuity is utilised as a way of defining the difference in an attribute considered interesting for a given learning context, and the cutoff as a function of the defined acuity values, and symbolic attribute differences for the analysed learning context.

The approach has shown its capability on recognising events, starting from noisy image-level data, and with a minimal configuration effort. The multiple possible extensions and applications for this approach are encouraging for exploring the behaviour of the approach in different scenarios and learning contexts.

8.4 Limitations of the Approach and Future Work

The general nature of the proposed video understanding framework for event learning allows that this approach can be extended in an huge number of new studies. The purpose of this section is to analyse the future work for the proposed video understanding framework, as extensions to the approach and as possible solutions to its limitations. These limitations are organised in terms of the period of time it could take to solve them (short term and long term limitations).

8.4.1 Short Term

In short term, the video understanding approach can be extended in several ways:

1. The calculation of reliability measures in the segmentation task can be an interesting extension of the approach. These reliability measures could be associated to the detected moving regions in order to account for the quality of segmentation in terms of the influence of illumination changes, level of contrast between the moving objects and the background of the scene, and the possibility of the presence of shadows, among other aspects.

2. The proposed reliability measures for the object attributes have been arbitrarily defined in this approach. Further analysis on different reliability measures can be performed in order to establish the measures which better represent the quality or coherence of the object attributes.

3. Background updating techniques should be considered in order to be able of coping with illumination changes, and moving background, among other issues on motion

segmentation. The information obtained from the proposed tracking approach, together with the reliability measures, could serve as feedback for a background updating method in order to better determine the background of the scene.

In addition to the presented future work, each task of the proposed video understanding framework presents its own limitations and future work. Next sections are dedicated to analyse the limitations and future work in the short term for the object classification (Section 8.4.1.1) and event learning (Section 8.4.1.2) tasks.

8.4.1.1 On Object Classification

The future work related to the object classification approach in the short term can be summarised as follows:

1. The resolution of the parallelepiped calculation problem presented in Section 4.1.1 has been formulated for focal point positions higher than the objects evolving in the scene. An object higher than the focal point height will lead to an erroneous calculation of the possible parallelepipeds associated to the object. This situation can not be considered as an error, but as a missing feature of the approach that has not been yet solved. The solution of this limitation implies the resolution of a new system of equations for covering this situations. Due to time constraints, this system of equations has not been solved in this thesis, and can be considered as future work.

2. Tests for the object classification approach have shown a lack of precision in the estimation of the object orientation angle α. Future work can point to the utilisation of alternative representation of an object, when this situation is detected.

8.4.1.2 On Event Learning

The future work for the proposed event learning approach in the short term can be summarised as follows:

1. In this thesis, few learning contexts have been utilised. The flexibility in the definition of the learning contexts allows the consideration of infinite possibilities for these contexts. Future work can focus on exploring different learning contexts.

2. The reliability measures utilised in the event learning approach are defined according to the interest of the user. In the future, different ways of defining these reliability measures can be explored.

3. In addition to the merge and split operators utilised by the proposed event learning approach, other operators could be incorporated to the approach, as the operators proposed by the INC learning algorithm presented in Section 2.4.4.

8.4.2 Long Term

In long term, the video understanding approach can be extended in several ways:

1. The mutual cooperation scheme proposed between the classification and tracking tasks can be considered as a first step to the cooperation between different tasks of the video understanding process. Another interesting cooperation scheme can be a feedback process between the tracking and segmentation tasks. The information provided by the tracking approach can be utilised by the segmentation task to focus the attention in the zones of the video image where movement can be more likely to occur. Hence, the segmentation could focus the analysis of movement in the entry zones of the scene and in the zones where moving objects have been detected, in order to improve the processing time performance of the segmentation task.

2. The idea of having two levels of mobile object representation, in the 2D image plane and in the 3D referential of the video scene, leads to the possibility of considering other simultaneous representations for the objects evolving in the scene. This multiple models can allow the video understanding approach to utilise the most reliable information from different possible representations. At the same time, these representations could be calculated or not depending on the availability and pertinence of obtaining this information. For example, an articulated model of a person could be interesting for being calculated if the proximity of the object to the camera is sufficient for appreciating its parts, or an appearance model based on colour could be interesting to be calculated if the level of contrast of the object with respect to the background is sufficient for obtaining valuable information.

3. The video understanding approach has been evaluated utilising one camera view. Multi-camera approaches could be studied in order to analyse how these techniques could improve the estimation of 3D attribute information.

4. The 3D models utilised for determining the class and 3D attributes of an object has been pre-defined. It could be an interesting subject of study to utilise learning techniques for learning these object models.

In addition to the presented future work, each task of the proposed video understanding framework presents its own limitations and future work. Next sections are dedicated to analyse the limitations and future work in the long term for the object classification (Section 8.4.2.1), object tracking (Section 8.4.2.2), and event learning (Section 8.4.2.3) tasks.

8.4.2.1 On Object Classification

The future work related to the object classification approach in the long term can be summarised as follows:

1. Even if the proposed representation of objects serves for describing a large variety of objects, the result from the classification algorithm is a coarse description of the

object. In order to evolve in the interpretation of more complex situations, more detailed and class-specific object models could be utilised when needed. Future work can point to the utilisation of more specific object representations according to the application, as articulated models, object contour, or appearance models, among others.

2. The classification approach has been proposed considering a pin-hole camera model. The adaptation of the object classification method for other calibration models, as the radial distortion model, can be an interesting subject of study.

8.4.2.2 On Object Tracking

The future work for the proposed object tracking approach in the long term can be summarised as follows:

1. The tracking approach is able to cope with dynamic occlusion utilising the object attribute information estimated in the previous frames to estimate the current values for the object attributes. As the tracking approach only estimates the current attributes based on previous information, the behaviour of the objects during the occlusion period can not be determined, which can lead tracking to errors of mistaken tracks. Then, 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.

2. The tracking approach is not capable to identify an object leaving the video scene and the re-entering in the scene as the same object. This is due that the information utilised for tracking is purely geometrical. In the future, the utilisation of appearance models can serve to identify the objects returning to the scene.

3. Even if the hypothesis generation process of the tracking approach has been optimised a large number of objects entering simultaneously entering in the scene can produce a high number of initial object configuration hypotheses as no object information is available when a new object enters in the scene. The use of alternative object representation can also serve to better define the initial hypotheses for the objects entering in the scene.

8.4.2.3 On Event Learning

The future work for the proposed event learning approach in the long term can be summarised as follows:

1. From the state of the art on incremental concept formation, it can be inferred that the distribution of state and event concepts in the generated hierarchy can depend in certain extent to the processing order of the object state instances. This means that

different hierarchies can be obtained from different ordering of the same instances. Future work can point to analyse the influence of the instance ordering in the quality of representation.

2. As the learning approach utilises the information related to each tracked object evolving in the scene separately, it does not seem inherent to the approach to represent relations between tracked objects. In the future, extensions of the proposed hierarchical state and event concept representation could be studied in order to explicitly consider the representation of object relations and interactions.

3. For several applications, the user can be interested in analysing the occurrence of pre-defined events interesting for the application. Future work can focus in the way these pre-defined events can be associated to the obtained hierarchical state and event concepts description.

4. It can be very interesting to study how the obtained hierarchies can serve as input for algorithms of semantic recognition, as building blocks for recognising composite events. Applications as data mining, video retrieval could also use the results of the proposed learning approach as the input data.

5. The potential of the proposed learning approach in applications of human behaviour learning and abnormal behaviour recognition must be studied.

Appendix A

Degenerated Cases for the Parallelepiped Model

Camera calibration is never perfect. For several reasons, the resulting perspective matrix can give undesirable projection results, especially in the image frame borders. This error in projection can be given by a poor calibration process, where selected pairs of calibration points ($X \leftrightarrow Y$) are imprecise, few, or not well distributed for representing the 3D scene correctly. Also, the projection error can be caused by applying a linear calibration process to a camera which presents some kind of distortion as, for example, the fish-eye camera which presents strong radial distortion[1]. In the scope of this thesis, only linear calibration with the Direct Linear Transform (DLT) [Abdel-Aziz and Karara 1971] has been considered, because of its simplicity in calibration, calculation speed, and because the majority of currently available cameras present a despicable level of distortion.

Considering T_j, with $j \in \{L, B, R, T\}$ as the parallelepiped vertexes bounded by a 2D blob b with 2D limits $B = \{X_{left}, Y_{bottom}, X_{right}, Y_{top}\}$. Normally, when no erroneous projection results are occurring, each vertex P_i ($\imath in \{1, 2, 3, 4\}$) associated to a variable T_j is bounded by only one blob limit. When undesirable projection results are obtained while calculating a parallelepiped model, , we can be in presence of a degenerate case, where a same vertex is bounded by two blob limits at the same time, as depicted in Figure A.1.

When in presence of a degenerate case, a simplification to the projection equations presented in Equation (4.7) occurs. Formally, consider blob limits $B_j \in B$ and $B_k \in B$, with $B_j \neq B_k$. Then, consider a vertex T_j bounded by the limit B_j, and a vertex T_k bounded by the limit B_k, so that $T_j = T_k = P_i(x_i, y_i)$ representing a degenerate case vertex. Then, as two limits share the same point (x_i, y_i), considering the two projection Equations from (4.7):

$$(p_{20} \times x_j + p_{21} \times y_j + p_{22} \times h \times In\_h(T_j) + p_{23}) \times B_j$$

[1]To represent appropriately the effect of radial distortion in the mapping between 2D image and 3D scene coordinates, in [Tsai 1986] and [Tsai 1987], authors propose a calibration technique which represent this kind of distortion. The result is a non-linear transform, where a cubic equation must be solved to perform the mapping from 3D scene to 2D image coordinates.

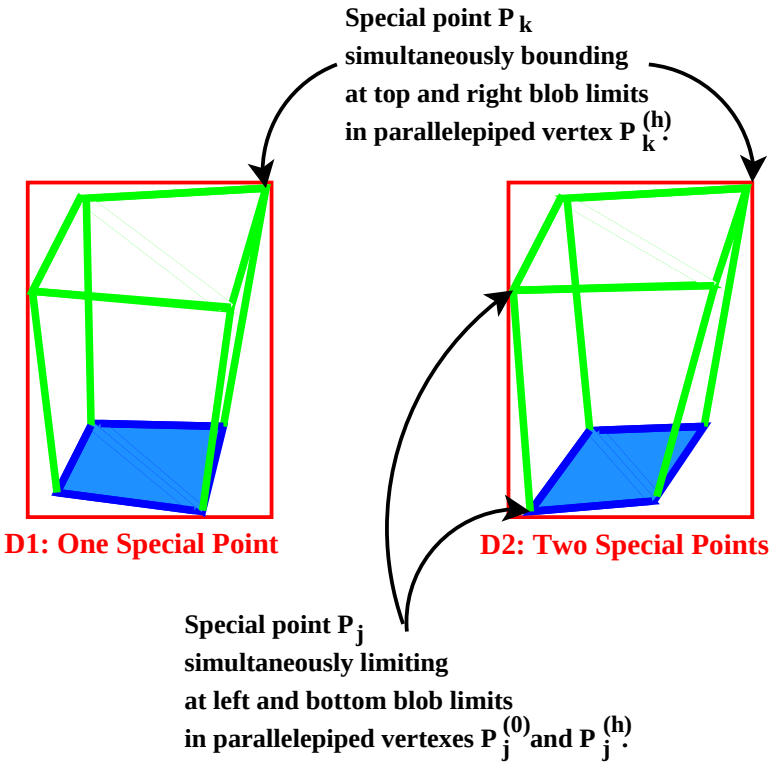


Figure A.1: Degenerated cases for parallelepiped calculation. Two degenerated cases can happen, where case D1 corresponds to one vertex point bounded by two blob limits, while in case D2 two vertexes are limited by two blob limits.

$$= p_{00} \times x_j + p_{01} \times y_j + p_{02} \times h \times In\_h(T_j) + p_{03},$$

$$\begin{aligned}(p_{20} \times x_k + p_{21} \times y_k + p_{22} \times h \times In\_h(T_k) + p_{23}) \times B_k \\ = p_{00} \times x_k + p_{01} \times y_k + p_{02} \times h \times In\_h(T_k) + p_{03},\end{aligned}$$

these Equation can be written as a function of (x_i, y_i), as $(x_i, y_i) = (x_j, y_j) = (x_k, y_k)$:

$$\begin{aligned}(p_{20} \times x_i + p_{21} \times y_i + p_{22} \times h \times In\_h(T_j) + p_{23}) \times B_j \\ = p_{00} \times x_i + p_{01} \times y_i + p_{02} \times h \times In\_h(T_j) + p_{03},\end{aligned}$$

$$\begin{aligned}(p_{20} \times x_i + p_{21} \times y_i + p_{22} \times h \times In\_h(T_k) + p_{23}) \times B_k \\ = p_{00} \times x_i + p_{01} \times y_i + p_{02} \times h \times In\_h(T_k) + p_{03}.\end{aligned}$$

Now y_i can be expressed in terms of x_i, in both equations. And then, both equations can be equalled:

$$\frac{(p_{00}-p_{20}\times B_j)\times x_i+(p_{02}-p_{22}\times B_j)\times h\times In\_h(T_j)+p_{03}-p_{23}\times B_j}{p_{21}\times B_j-p_{01}} = \frac{(p_{00}-p_{20}\times B_k)\times x_i+(p_{02}-p_{22}\times B_k)\times h\times In\_h(T_k)+p_{03}-p_{23}\times B_k}{p_{21}\times B_k-p_{01}}$$

Finally, without yet solving the system of equations, one point has been already determined, as shown in Equation (A.1).

$$x_i=\frac{\Big((p_{01}-p_{21}\times B_j)\times(p_{02}-p_{22}\times B_k)\times In\_h(T_k)-(p_{01}-p_{21}\times B_k)\times(p_{02}-p_{22}\times B_j)\times In_h(T_j)\Big)\times h-(B_j-B_k)\times(p_{03}\times p_{21}-p_{01}\times p_{23})}{(B_j-B_k)\times(p_{00}\times p_{21}-p_{01}\times p_{20})}$$

$$y_i=\frac{(p_{00}-p_{20}\times B_j)\times x_i+(p_{02}-p_{22}\times B_j)\times h\times In\_h(T_j)+p_{03}-p_{23}\times B_j}{p_{21}\times B_j-p_{01}} \tag{A.1}$$

This way, a point $P_i = (x_i, y_i)$ can be determined before the resolution of the system of equations presented in Section 4.1.1. If the situation corresponds to degenerate case $D1$, where only one vertex point is bounded by two blob limits, two variables are solved and two equations are utilised, now having to solve the system for the eight remaining variables with the eight remaining equations.

If the situation corresponds to degenerate case $D2$, where two vertexes are limited by two blob limits, then four variables are solved and the four projection Equations (4.7) are utilised, now having to solve the system for the six remaining variables with the six remaining base Equations (4.8).

Notice that the utilisation of two equation for a degenerate case vertex, means that physically exists one point P_i, with $i \in \{1, 2, 3, 4\}$, which is not bounded in any of its possible vertexes at heights 0 and h. The same happens with two degenerate case vertexes, where two points P_i will not be bounded by the blob limits.

Appendix B

Detailed Formulation of the Object Tracking Process

A pseudo-code representation of the proposed tracking method, presented in Section 5.3, is displayed below:

```
procedure reliabilityTracking (In newBlobs, In oldHypothesesSets,
                               Out mobilesList)
   begin
      newBlobs = preMerge(newBlobs);

      oldHypothesesSets = involvedBlobs(oldHypothesesSets, newBlobs);
      oldHypothesesSets = mergeHypothesesSets(oldHypothesesSets);

      oldHypothesesSets = generateTracks(oldHypothesesSets, newBlobs);
      newHypothesesSets = generateHypotheses(oldHypothesesSets);

      for each hypothesesSet of newHypothesesSet do

         for each hypothesis of hypothesesSet do
            insertNewMobiles(hypothesis, hypothesesSet);
         end for

      end for

      newHypothesesSets = createNewMobiles(newHypothesesSets);
      newHypothesesSets = filterHypotheses(newHypothesesSets);
      newHypothesesSets = splitHypothesesSets(newHypothesesSet);

      return mobilesList = bestMobiles(newHypothesesSets);
end.
```

First, a *preMerge* step performs preliminary merge operations over blobs presenting highly unlikely initial features, reducing the number of blobs to be processed by the tracking procedure. The pre-merge procedure is performed for blobs which size is too small to represent any of the pre-defined 3D object models (e.g. merge body parts to build a tentative mobile corresponding to a person). Blobs contained by another blob are also candidates for immediate merge.

Then, the mobile hypothesis update process starts by the *involvedBlobs* procedure, used for determining the blobs that can participate to the track updating process for a mobile. A blob will be **involved** with a mobile, if this blob can be part of the visual evidence for the mobile in the current analysed frame. A blob can be involved with a mobile according to its proximity to the predicted state of the mobile in the current video frame. Then, *mergeHypothesesSet* procedure merges visually related hypothesis sets, which were separated until the current frame. This processes have been described in Section 5.3.1.

Next, the functions *generateTrack* and *generateHypotheses* are the constituting parts of the hypothesis updating process, which is described in Section B.1.

First, in *generateTrack* process, the most coherent mobile tracks for each mobile are calculated. First, a ranking of the most coherent track for each object is developed. The construction of these tracks uses previous reliable information of the same mobile, in order to start the search of tracks at the most coherent position and with the most coherent object size. The process of mobile track generation is described in detail in Section B.1.1.

Then, in *generateHypotheses* procedure, new hypotheses are generated using the previously calculated best tracks for each mobile. This process immediately generates the optimal hypotheses from the best track rankings, by optimising the hypothesis likelihood measure P_H (Equation (5.1)) for each hypothesis. A hypothesis is accepted if its likelihood measure relative to the hypothesis of highest likelihood exceeds a pre-defined threshold and if the total number of accepted hypotheses does not exceed a pre-defined maximal number of accepted hypotheses. The hypothesis generation process is fully described in Section B.1.3.

If a hypothesis is not complete with respect to all visual evidence considered for the hypothesis set, this visual evidence is then considered as potential new mobile objects entering the scene for the incomplete hypothesis, and the procedure *insertNewMobiles* exhaustively generates all possible combinations of new objects as no assumption can be made about the validity of a mobile hypothesis at the entrance of a new object. In the same way, procedure *createNewMobiles* generates mobiles from visual evidences not matched with any of the mobiles present in the scene. These tasks have been described in Section 5.3.3.

Then, a filter for hypotheses is applied (procedure *filterHypotheses*). Finally, hypothesis sets with just one alternative can be separated in different hypothesis sets with one

hypothesis containing just one mobile, if mobiles are currently not visually related, simplifying later tracking process for these mobiles. The algorithm returns the set of mobile objects with the highest likelihood hypotheses. These tasks have also been described in Section 5.3.3.

B.1 Updating existing Mobile Hypotheses

The process of hypothesis updating can be separated in two parts which correspond to the functions *generateTracks* and *generateHypotheses* defined at the global description of the tracking approach presented in Section B. These functions are intended to update the tracks of the mobiles represented in the hypotheses (Section B.1.1), and to generate the new hypotheses based on the updated mobile alternative track solutions (Section B.1.3).

B.1.1 Generation of Tracks for Mobiles

For each mobile contained in the hypothesis set, the function *generateTracks*, presented in Section B, associates to the mobile a list of the most likely tracks represented also as mobiles updated with the visual evidence extracted from the current video frame.

The track generation method applies two different generation methods according to the number of frames of mobile life-span. The first method is applied with a life-span of one or two frames, as for first and second frames, it is not possible to determine the coherence of the mobile velocity attributes.

This first generation method consists in considering all the blobs belonging to the set of involved blobs, which have been previously obtained with the function *involvedBlobs*, described in Section 5.3.1. This set of blobs is utilised to generate the new evidence associated to the mobile as described in the pseudo-code algorithm *generateInitialMobileTracks* below:

```
procedure generateInitialMobileTracks (In segmentedBlobs, In analysedMobile,
                                       Out generatedMobiles)
begin
   involvedBlobs = getInvolvedBlobs(segmentedBlobs, analysedMobile);
   blobGroups = getBlobGroups(involvedBlobs);

   for each group in blobGroups do
      blobCombinations = getBlobCombinations(group);

      for each combination in blobCombinations do
         mergedBlob = mergeBlobs(combination);

         if mergedBlob not alreadyIncluded(mergedBlob) then
            newMobile = updateMobile(analysedMobile, mergedBlob);
```

```
            if coherentMobile(newMobile) then
               insertMobile(newMobile, generatedMobiles);
            end if
         end if
      end for
   end for

   return generatedMobiles;
end.
```

Using the set of involved blobs obtained with function *getInvolvedBlobs*, and previously determined by function *involvedBlobs* (Section B), the *generateInitialMobileTracks* algorithm generates all the coherent combinations of blobs that can be associated to the analysed mobile.

For this purpose, the first function *getBlobGroups* separates involved blobs in groups according to the possibility of these blobs to be merged between each other, according to their proximity. Then, for each group of blobs, the function *getBlobCombinations* generates all the possible blob combinations, also considering different number of considered blobs in the combination.

Each of these blob combinations is merged by the function *mergeBlobs*, to obtain the visual evidence to be associated to the currently analysed mobile. Before associating the merged blob to the mobile, the function *alreadyIncluded* verifies if this merged blob really represents a new visual evidence. This verification is necessary because in some cases different blob combinations can give the same merged blob result.

Then, function *updateMobile* associates the visual evidence represented by the merged blob to the currently analysed mobile, to generate a possible mobile track, represented by a new mobile. This function represents the mobile updating process described in Section B.1.2.

Minimal coherence of the new mobile is immediately checked by function *coherentMobile*, which performs tests for the following temporal coherence reliability measures:

- $\mathbf{CD_{3D}}$: The temporal coherence reliability measure for 3D dimensional data, presented in Equation (5.16), is evaluated using a pre-defined *MinimalDimensionalCoherence* threshold. This threshold must be low, in order to serve as filter of really invalid solutions. In practise, a value of the threshold equal to 0.1 has shown a good behaviour. This measure is analysed when the number of *classified* blobs in the buffer is higher than one, in order to be able to extract dimensional information from at least two classified blobs and check their coherence.
- $\mathbf{CV_{3D}}$: The temporal coherence reliability measure for 3D velocity data, presented in Equation (5.17), is evaluated using a pre-defined *MinimalVelocityCoherence*

threshold. This threshold must be also low. In practise, a value of the threshold equal to 0.1 has shown a good behaviour. This measure is analysed when the number of *classified* blobs in the buffer is higher than two, in order to be able to calculate at least two instant velocities from data and check their coherence.

- $\mathbf{CD_{2D}}$: The temporal coherence reliability measure for 2D dimensional data, presented in Equation (5.16), is evaluated also using the pre-defined *MinimalDimensionalCoherence* threshold. This measure is analysed when the number of not *lost* blobs in the buffer is higher than one, in order to be able to extract dimensional information from at least two blobs and check their coherence.

- $\mathbf{CV_{2D}}$: The temporal coherence reliability measure for 2D velocity data, presented in Equation (5.17), is also evaluated using a pre-defined *MinimalVelocityCoherence* threshold. This measure is analysed when the number of not *lost* blobs in the buffer is higher than two, in order to be able to calculate at least two instant velocities from data and check their coherence.

If all these four tests are passed, the function *insertMobile* then includes the new mobile in the list of tracks for the analysed mobile ordered from higher to lower likelihood measure p_m (Equation (5.15)), obtaining an ordered list of valid mobiles as a final result of the mobile generation process. If no coherent association has been found for the analysed mobile, a new mobile is created and tagged as *lost*. The treatment for *lost* objects is described in Section 5.3.4.

Finally, the first generation method ends by limiting the number of possible tracks for a mobile. The new mobiles are suppressed if their likelihood measure p_m, normalised by the best p_m measure, is lower than a pre-defined *MinimalRelativeMobileLikelihood* threshold. As the p_m value is normalised, the threshold can have a high value. In practise, values for the threshold around 0.95 have shown good results. Then, the best surviving new mobile number is limited to a pre-defined *MaximumMobileTracks* number.

The second generation method is applied with a life-span of more than two frames, as now is possible to determine the coherence of the velocity attributes for the mobile. This generation method consists in using the set of involved blobs to first generate the new evidence associated to the mobile which best fits the estimated bounding box associated to a mobile from its current attribute values, and then generates other mobile tracks using the remaining involved blobs.

If no involved blobs have been found for the analysed mobile, a new mobile is created and tagged as *lost*. The treatment of this case is the same as described in the first mobile generation method.

If only one involved blob has been found for the currently analysed mobile, a new mobile is immediately generated by updating the analysed mobile dynamics with the information extracted from the involved blob. If the analysed mobile is in ensure mode the occurrence

of the special situations is analysed, as presented in Section 5.3.4.

When the involved blob set size is higher than one blob, the algorithm *generateMobileTracks* is applied, which is described in the pseudo-code algorithm below:

```
procedure generateMobileTracks (In segmentedBlobs, In analysedMobile,
                                Out generatedMobiles)
begin
   involvedBlobs = getInvolvedBlobs(segmentedBlobs, analysedMobile);
   initialBlob = getInitialBlob(involvedBlobs, analysedMobile);

   if initialBlob found then
      mergedBlob = getInitialMergeCombination(initialBlob,
                                              involvedBlobs, analysedMobile);
   else
      mergedBlob = getBlobWithHighestBlobSupport(involvedBlobs, analysedMobile);

      if initialBlob not found then
         lostMobile = generateLostMobile(analysedMobile);
         insertMobile(lostMobile, generatedMobiles);
         return generatedMobiles;
      end if
   end if

   newMobile = updateMobile(analysedMobile, mergedBlob);

   if coherentMobile(newMobile) then
      insertMobile(newMobile, generatedMobiles);
      bestP_m = P_m(newMobile);
   else
      bestP_m = 0.0;
   end if

   if isInEnsureMode(analysedMobile) then
      specialMobile = getSpecialMobile(currentMobile, mergedBlob);
      insertMobile(specialMobile, generatedMobiles);

      if coherentMobile(specialMobile) then
         insertMobile(specialMobile, generatedMobiles);
      end if
   end if

   validBlobs = getValidBlobs(involvedBlobs, mergedBlob, currentMobile);

   blobCombinations = getBlobCombinations(validBlobs);
```

```
   for each combination in blobCombinations do
      mergedBlob = mergeBlobs(combination);

      if mergedBlob not alreadyIncluded(mergedBlob) then
         newMobile = updateMobile(analysedMobile, mergedBlob);

         if     coherentMobile(newMobile)
            and P_m(newMobile)/bestP_m > MinimalRelativeMobileLikelihood then
            insertMobile(newMobile, generatedMobiles);

            if P_m(newMobile) > bestP_m then
               bestP_m = P_m(newMobile);
            end if
         end if
      end if
   end for

   return generatedMobiles;
end.
```

Using the set of involved blobs obtained with function *getInvolvedBlobs*, and previously determined by function *involvedBlobs* (Section B), the *generateInitialBlob* algorithm searches the initial blob with *mobileSupport* higher than *HighVisualSupportRate* which has the best *blobSupport* among the involved blobs, with respect to the estimated bounding box generated with the analysed mobile attributes. This means that the algorithm searches for the visual support blob inside the estimated bounding box which better covers the area of the estimated bounding box.

If the initial blob is found, the function *getInitialMergeCombination* merges this initial blob with other blobs which are near to the initially considered blob and inside the estimated bounding box from the mobile attributes. The resulting blob is used as a new initial blob for finding blobs inside the estimated bounding box which are near to this resulting blob, and so on, until no other blob inside the estimated bounding box are found. This way, the first initial visual support for the mobile is found.

If no initial blob is found, the process tries to get an initial blob with a second function *getBlobWithHighestBlobSupport*, which returns the blob with the highest *blobSupport* measure. This way, the initial blob is considered as the blob best covering the estimated bounding box area, but not necessarily inside of the estimated bounding box. If still no initial blob is found, a mobile representing the case of *lost* visual evidence is generated (function *generateLostMobile*). This mobile track solution is inserted and the mobile track generation stops.

Using the merged blob obtained with function *getInitialMergeCombination* or the initial

blob obtained with function *getBlobWithHighestBlobSupport*, the track generation process builds the updated mobile track (Function *updateMobile*) and inserts the solution to the list of possible mobile tracks if this solution passes the coherency check function *coherentMobile* (as in the first track generation method). If the new mobile track is coherent, its p_m likelihood measure (Equation (5.1)), obtained with function *P_m*, is utilised as the initial best p_m value (*bestP_m*), else the best p_m value is initialised to zero.

If the analysed mobile is in *ensure* mode (function *isInEnsureMode*), possible special cases are analysed for mobile tracks as presented in Section 5.3.4, by the function *getSpecialMobile*. If the function finds that the current visual support blob represents a special situation, the mobile representing the special case is generated and inserted to the mobile track list if this mobile is coherent.

Then, function *getValidBlobs* generates a list of valid blobs from the involved blob list, considering the blobs not considered in the initial blob solution which have a *blobSupport* higher than zero. Then, as in the first generation method, the function *getBlobCombinations* generates all the possible blob combinations, also considering different number of blobs in the combination.

Each of these blob combinations are merged by the function *mergeBlobs*, to obtain the visual evidence to be associated to the currently analysed mobile. Before associating the merged blob to the mobile, the function *alreadyIncluded* verifies whether this merged blob really represents new visual evidences, as in the first generation method.

Then, the function *updateMobile* generates the new mobile. The new track solution is inserted to the list of tracks if the solution passes the coherency check function *coherentMobile* and if the p_m measure of the solution, normalised by the best found p_m measure *bestP_m*, is higher than the *MinimalRelativeMobileLikelihood* threshold (already defined for the first generation method). If the new mobile is inserted, the *bestP_m* is updated if the p_m measure of the new mobile is higher.

Finally, after obtaining the new mobile track list from the algorithm *generateMobileTracks*, the second generation method ends by limiting the number of possible tracks for a mobile, as described at the end of the first mobile track generation method.

Hence, the result of the track generation process is a list of possible mobile tracks ordered by the likelihood measure p_m, for each mobile in every hypothesis contained in the hypothesis sets. This result serves as input for the hypothesis generation process presented in Section B.1.3.

B.1.2 Mobile Initialisation and Updating

In order to track a mobile object evolving in the video scene, its attribute information must be updated with the information given by the visual evidence associated to the object in

the current frame. The process of updating this information is determined by different stages according to the mobile life-span and the coherence of its attribute information.

First, in order to ensure a minimal evidence of the mobile object existence, the visual evidence on the first frames of existence of the tentative mobile is stored in a *blob buffer*. At these first frames only the 2D information updates the dynamics model presented in Section 5.2.2. This way, the unnecessary classification of blobs that are later lost is avoided, improving the processing time performance.

The number of frames to be processed with only the 2D information are customisable, but a reasonable value should be considered between three and the size of the blob buffer associated to the mobile. Three values are necessary for a first verification of the temporal coherency of the attribute velocity, as two pairs of blobs are needed for getting two instant velocities. The blob buffer size is taken as an upper bound, which ensures to avoid the loss of information, as blob information leaving the buffer is lost and next step uses this blob information to estimate the initial 3D information.

Second, when the upper bound for processing only 2D information is reached, the updating process initialises the 3D information as described in the following pseudo-code routine:

```
procedure initialise3DInformation (In blobsBuffer, In initialAttributes,
                                   Out updatedAttributes)
begin
   while (no coherent 3D solution is found) or (blobsBuffer reaches end) do
      initialiseAttributes(updatedAttributes);

      for each blob in blobsBuffer do
         make3DClassification(blob);

         if blob is classified then
            PBest = 0.0;

            for each classified expected object class in blob do
               classAttributes = updatedAttributes;
               updateAttributes(blob, class, classAttributes);

               for each remainingBlob in blobsBuffer after blob do
                  classAttributes = guidedClassification(remainingBlob,
                                                         classAttributes);
                  updateAttributes(blob, class, classAttributes);
               end for

               if     (P(classAttributes) > minimalMobileLikelihood)
                  and (P(classAttributes) > PBest) then
                  PBest = P(updatedClassAttributes);
```

```
                updatedAttributes = classAttributes;
             end if
          end for

          if coherent 3D solution is found then
             return updatedAttributes;
          end if
          update2DInformation(blob, updatedAttributes);
       else
          update2DInformation(blob, updatedAttributes);
       end if
    end for
 end while

 return initialAttributes;
end.
```

The 3D attribute initialisation procedure searches, for each blob in the blob buffer (starting from the oldest one), a coherent 3D solution. When a blob is successfully classified (function *make3DClassification*), the procedure searches for the best configuration among all the classified expected object model classes. If this is the case, for each classified object class, the procedure first executes the *updateAttributes* function, which updates the mobile information according to the currently processed 2D blob and 3D class information in the blob buffer.

Then, for each remaining blob in the blob buffer, the function *guidedClassification* is used to classify these blobs using the updated mobile information for the blob buffer. If the mobile object m likelihood measure p_m (obtained with function $P(\cdot)$) is higher than a pre-defined *minimalMobileLikelihood* threshold, the best 3D configuration in terms of measure p_m is stored and the updated attribute values for the class become the coherent 3D solution for the mobile.

More specifically, the *guidedClassification* function consists in performing the search of the most coherent parallelepiped according to the current mobile attribute values, given a specific object class. Attribute velocity information V_a (Equation (5.11)) is utilised to estimate the current position of the mobile object, attribute value standard deviations σ_a (Equation (5.6)) and σ_{V_a} (Equation (5.14)) are utilised to determine the limits of exploration for the 3D classifier, and mean attribute values $\bar{a}$ (Equation (5.3)) are utilised as the starting point for performing the search of the 3D parallelepiped model.

The utilisation of *guidedClassification* function has a twofold benefit: to search 3D parallelepipeds which are coherent with the currently obtained mobile object information, and to guide the 3D classification task in the search of the 3D solution, improving its processing time performance.

All information about other non-optimal coherent 3D solutions for other object classes is also stored in order to give to the mobile attribute updating process the possibility to change the 3D information in case that another object class becomes more likely than the currently selected one.

If the classification of the initial blob does not give any class label or if no coherent 3D solution is found among all classes, only the 2D information is updated (function *update2DInformation*) and the next blob in the blob buffer sequence is considered as starting point to search for a coherent 3D solution. If no coherent 3D solution is found at all, the attribute values obtained before starting the 3D information initialisation procedure is considered, considering the mobile as an object of unknown class.

Third, for the following blob visual evidence associated to the mobile, after obtaining the result from procedure *initialise3DInformation*, the attribute updating process continues to apply the *guidedClassification* to classes with a previous found 3D solution, while the function *make3DClassification* is applied to classes without associated 3D information in order to find initial 3D information.

Fourth, if the number of classified blobs for the currently most coherent class arrives to a pre-defined *minimalNumberOfClassifiedBlobs* and the mobile measure p_m is higher than a pre-defined *minimalMobileLikelihoodToEnsure* threshold, the mobile passes to **ensure mode**. In this updating mode, just the currently most coherent class is evaluated with the *guidedClassification* function, optimising the performance of the updating process by considering that the currently associated class is the correct one for the mobile object.

B.1.3 Generation of Hypothesis from Mobile Tracks

The hypothesis generation process utilises as input the result of the mobile track generation process described in previous Section B.1. This process consists in generating for each hypothesis set, the new set of hypotheses with updated mobile information which maximises the hypothesis likelihood measure P_H presented in Equation (5.1). The idea is to immediately generate these best hypothesis sets, instead of generating all the possible hypotheses and then pruning the ones with lower P_H.

For performing this process, the function *generateHypotheses*, presented in Section B, is explained in detail. A pseudo-code representation of this algorithm is presented below:

```
procedure generateHypotheses (In currentHypothesesSets,
                              Out updatedHypothesesSets)
begin

  for each hypothesesSet in currentHypothesesSets do
    clearNewGeneralHypothesesList(newHypotheses);

    for each hypothesis in hypothesesSet do
```

```
addedHypotheses = 0;
clearNewHypothesesList(hypothesis);
clearCombinationsList(combinationsToAnalyse);
mobilesList = getBestPHNewMobiles(hypothesis);
currentPH = getPHValue(mobilesList);
mobileContributions = getContributionsToPM(mobilesList);
indexToModify = 0;
bestCombination = makeCombination(bestMobilesList, currentPH,
                                  indexToModify, mobileContributions);
markAdded(bestCombination);
insertCombination(bestCombination, combinationsToAnalyse);
newHypothesis = getHypothesisFromCombination(bestCombination);

if validHypothesis(newHypothesis) then
  insertNewHypothesis(newHypothesis, hypothesis);
  addedHypotheses = addedHypotheses + 1;
end if
numMobiles = numberOfMobiles(hypothesis);
maxHypotheses = numMobiles * maxPerMobile;

totalNumberOfFrames = 0;

for index = 0 to (numMobiles - 1) do
  totalNumberOfFrames += getNumFrames(mobilesList[index]);
end for

while     addedHypotheses < maximumHypothesisNumber
      and addedHypotheses < maximumRetainedHypotheses
      and combinationsToAnalyse not empty             do

  for each combination in combinationsToAnalyse tagged added do
    currentIndex = getMobileIndexToModify(combination);
    for indexToModify = currentIndex to (numMobiles - 1) do
      if mobileListNotEnding(indexToModify, combination) then
        mobilesList = getMobilesList(combination);
        mobilesList[indexToModify]
                    = nextMobile(mobilesList[indexToModify]);
        currentPH = getPH(combination);
        mobileContributions = getContributions(combination);
        currentPH -= mobileContributions[indexToModify];
        mobileFrames = getNumFrames(mobilesList[indexToModify]);
        mobileContributions[indexToModify]
                     = mobileFrames * P_m(mobilesList[indexToModify])
                                    / totalNumberOfFrames;
```

```
            currentPH += mobileContributions[indexToModify];
            newCombination = makeCombination(mobilesList, currentPH,
                                             indexToModify,
                                             mobileContributions);
            insertCombination(newCombination, combinationsToAnalyse);
          end if
        end for
      end for
      eliminateMarkAddedCombinations(combinationsToAnalyse);

      if combinationsToAnalyse is not empty then
        bestPH = 0;

        for each combination in combinationsToAnalyse do
          currentPH = getProbabilityValue(combination);

          if currentPH < bestPH then
            break for;
          end if
          bestPH = currentPH;
          markAdded(combination);
          newHypothesis = getHypothesisFromCombination(combination);

          if validHypothesis(newHypothesis) then
            insertNewHypothesis(newHypothesis, hypothesis);
            addedHypotheses = addedHypotheses + 1;
          end if
        end for
      end if
    end while
  end for

  for each hypothesis in hypothesesSet do
      insertNewHypothesesList(hypothesis, newHypotheses);
  end for

  eliminateExcessOfHypotheses(newHypotheses, maximumRetainedHypotheses);
  newHypothesesSet = makeHypothesesSet(newHypotheses);
  insertHypothesesSet(newHypothesesSet, updatedHypothesesSets);
end for

return updatedHypothesesSets;
end.
```

The hypothesis generation process is independent for each hypothesis set. First, function *clearNewGeneralHypothesesList* resets the list of new hypotheses for the currently analysed hypothesis set. Then, for each hypothesis of the set, a counter of the new inserted hypotheses *addedHypotheses*, a list of the new hypotheses associated to the analysed hypothesis ordered by the likelihood measure P_H, and a list of combinations to analyse by the hypothesis generation process *combinationsToAnalyse* are considered.

Each combination in the *combinationsToAnalyse* list consists of the analysed mobile track for each mobile, the value of the measure P_H, the index of the currently analysed list of tracks for a mobile, and the contribution to each of the analysed mobile tracks. The *combinationsToAnalyse* list is also ordered by the measure P_H.

The initial combination of mobiles is constructed by the function *makeCombination* (and stored in *bestCombination*), which utilises several inputs:

- *mobilesList:* Returned by the function *getBestPHNewMobiles*, corresponds to the leading positions for the track lists for each mobile in the hypothesis. As the track lists are ordered by the p_m measure, this first combination corresponds to the one giving the highest P_H measure for the hypothesis.
- *currentPH:* Returned by the function *getPH*, corresponds to the P_H measure for the given mobiles combination.
- *mobileContributions:* Returned by function *getContributions*, corresponds to the list of contributions of each mobile to the measure P_H, given by $p_m \cdot T_m$, as deduced from Equations (5.1) and (5.2), for a mobile m.
- The *indexToModify* value which is initially set to zero, representing the mobile track list currently analysed in the combination.

Then, the function *markAdded* tags the initial best combination as *added*, which means that this combination has been already analysed and that new mobile combinations can be generated from it. Next, this initial combination is inserted to the list *combinationsToAnalyse* by the function *insertCombination*. The hypothesis associated to this combination is generated (function *getHypothesisFromCombination*) and tested for validation by function *validHypothesis*.

A hypothesis is considered valid if there is no severe collisions between the parallelepiped bases of the mobile objects which have available and reliable 3D information. If this is the case, the hypothesis is inserted in the list of new hypotheses of the currently analysed hypothesis by the function *insertNewHypothesis*, and the *addedHypotheses* counter is incremented.

Then, the variable *totalNumberOfFrames* is calculated, which accounts for the total number of frames considering all the mobiles in the analysed hypothesis. This variable is used as the normalising factor for the P_H measure, as T_m is normalised for each mobile

m in Equation (5.2).

At this point, the hypothesis generation process starts. The process will stop if the *addedHypotheses* counter reaches the *maximumHypothesisNumber* bound or the *maximumRetainedHypotheses* bound, or when the *combinationsToAnalyse* list is empty. The *maximumHypothesisNumber* bound is a particular bound for the analysed hypothesis which authorises a pre-defined number of hypotheses per mobile *maxPerMobile*, for each mobile forming the analysed hypothesis. The *maximumRetainedHypotheses* bound is a pre-defined maximum number of total hypotheses for each hypothesis set.

For each combination in the *combinationsToAnalyse* list tagged as *added* a list of new combinations is generated. Starting from the mobile track lists index for the added combination, a new combination is generated considering the modification of a mobile at different mobile track lists. Each of these new combinations are updated by advancing in the track list of the given track lists index to the next mobile, subtracting the contribution of the previous mobile in the list from the P_H measure, and updating the mobile contribution to the P_H measure, as the value given by the new analysed mobile from the list.

All these new combinations, generated from all the currently tagged *added* combinations, are stored in the *combinationsToAnalyse* list, and the currently tagged *added* combinations are eliminated. Then, if the *combinationsToAnalyse* list is not empty, the next combinations with the best P_H value are converted to a hypothesis (function *getHypothesisFromCombination*). This hypothesis is inserted in the list of new hypotheses of the currently analysed hypothesis (function *insertNewHypothesis*), and the *addedHypotheses* counter is incremented, if the hypothesis passes the test of function *validHypothesis*.

Finally, the new hypotheses generated for each analysed hypothesis are stored in the global *newHypotheses* list for the hypothesis set (function *insertNewHypothesesList*). Then, *eliminateExcessOfHypotheses* leaves the *newHypotheses* list with the hypotheses with best P_H measures not exceeding the *maximumRetainedHypotheses* number, and the hypothesis set is reconstructed using the final *newHypotheses* list (function *makeHypothesesSet*). Then, this new hypothesis set is added to the *updatedHypothesesSets* list.

Appendix C

Introduction: Version Française

L'un des problèmes les plus difficiles dans le domaine de la vision par ordinateur et l'intelligence artificielle est l'interprétation automatique des séquences d'images ou de compréhension de la vidéo. La recherche dans ce domaine se concentre principalement sur le développement de méthodes pour l'analyze des données visuelles à extraire et sur le traitement des informations sur le comportement des objets physiques dans une scène du monde réel.

L'avancement dans l'extraction des données visuelles de bas niveau dans la vidéo a permis aux chercheurs de se concentrer sur des analyzes de plus haut niveau impliquant des aspects temporels, comme la reconnaissance et l'apprentissage des événements. Dans les dernières années, l'analyze des événements dans la vidéo est devenu l'un des plus grands domaines d'intérêt dans la communauté de compréhension de la vidéo [Hu et al. 2004a], même si le nombre d'études dans ce domaine est encore faible, par rapport aux autres domaines de compréhension de la vidéo. L'extraction de l'information sur les événements en vidéo implique généralement le traitement approprié des tâches du bas niveau, comme la détection de mouvement, le classement des objets, et la suivi des objets, afin de générer l'entrée appropriée pour les tâches d'analyze des événements.

L'objectif principal de cette thèse est de **proposer un cadre de travail dans la compréhension de la vidéo pour l'apprentissage et la reconnaissance des événements en général, pour des applications du monde réel.**

Un nombre croissant des approches pour l'analyze des événements ont été proposées dans les dernières années. L'intérêt des chercheurs a été essentiellement focalisée sur la reconnaissance des événements pré-définis [Howarth and Buxton 2000], [Medioni et al. 2001], l'apprentissage hors ligne des relations entre des événements pré-définis [Hongeng et al. 2004], [Chan et al. 2006a], [Hamid et al. 2005], [Toshev et al. 2006]), et l'apprentissage hors ligne des événements [Fernyhough et al. 2000], [Remagnino and Jones 2001], [Hu et al. 2006], [Niebles et al. 2006], [Xiang and Gong 2008]. À ce jour, très peu d'attention a été accordée à l'apprentissage incrémental des événements dans la vidéo [Mugurel et al. 2000], [Piciarelli and Foresti 2006], qui devrait être la suivante étape pour des applications en

temps réel pour la reconnaissance des événements imprévus, ou pour la détection des comportements anormaux.

L'analyze des événements en vidéo dispose de plusieurs applications intéressantes. La surveillance vidéo est l'un des plus importants domaines d'application. Pour la sécurité des lieux publics, la surveillance vidéo est couramment utilisé, mais la augmentation du nombre de caméras a conduit à la saturation des moyens de transmission et de analyze de l'information, car il est difficile de surveiller simultanément plusieurs centaines d'écrans. Pour aider à l'utilisateur dans cette tâche difficile, des techniques de compréhension de la vidéo peuvent être utilisées pour le filtrage et le tri des scènes qui peuvent être intéressantes pour un opérateur humain. Par exemple, le projet AVITRACK de surveillance vidéo dans les aéroports [AVITRACK 2002], génère des rapports aux opérateurs sur les activités qui se produisent dans l'aire de trafic aérien (par exemple, l'opération de ravitaillement), et génère des alarmes en cas de situations indésirables (par exemple, la collision entre un véhicule de fret et un avion). Comme autre exemple, le projet CARETAKER pour l'analyze des comportements dans les espaces publics [CARETAKER 2006], [Carincotte et al. 2006], génère des alarmes en cas de situations indésirables (par exemple, des personnes se battent dans un parc de stationnement), et effectue l'extraction des données sur des séquences vidéo de longue durée pour analyser des schémas de comportement des objets qui évoluent dans la scène.

Un autre domaine d'application intéressant est celui de surveillance de la santé des personnes. Elle consiste dans la surveillance de l'activité d'une personne en utilisant des caméras et de capteurs afin d'assurer son intégrité physique et mentale. Pour ces applications, des techniques de compréhension de la vidéo peuvent être utilisées pour générer automatiquement des alarmes en cas que la santé de la personne surveillée est en danger. Par exemple, le projet GERHOME pour la garde des personnes âgées à domicile [GERHOME 2005], [Zouba et al. 2007], utilise des capteurs de chaleur, de son et de porte, avec des caméras vidéo pour surveiller les personnes âgées. Le système de compréhension de la vidéo proposé dans le cadre du projet GERHOME est capable d'alerter la famille ou de demande de soutien médical dans le cas où un accident est détecté (par exemple, la personne tombe), et de surveiller le comportement de la personne pour alerter si certaines actions nécessaires n'ont pas été effectués (par exemple, la personne n'a pas pris ses médicaments, ou la personne n'a pas pris de l'eau pour une longue période dans une saison chaude).

L'utilisation de l'apprentissage incrémental des événements en vidéo permet d'obtenir la probabilité d'occurrence des événements dans une scène vidéo, qui peut être utilisée pour la détection des situations anormales sur la base d'un modèle adaptative de la fréquence des événements dans une scène vidéo. La détection de situations anormales peut être une caractéristique intéressante pour des nombreuses applications pour la vidéo-surveillance et pour la surveillance de la santé des personnes, car elle permet d'alerter un opérateur sur l'apparition d'une nouvelle situation inconnue, qui pourrait être indésirable ou dangereuse.

Cette thèse concentre son intérêt dans des applications pour l'apprentissage incrémental des événements, où plusieurs objets de type divers peuvent interagir dans la scène (par exemple, des personnes, des véhicules). Les événements d'intérêt sont également diverses (par exemple, les événements liés à des trajectoires, la posture), car l'intérêt se concentre dans l'apprentissage des événements en général. Les objets qui évoluent simultanément dans la scène peuvent être nombreux, mais l'intérêt est centré sur les objets qui peuvent être suivis individuellement afin d'être en mesure de reconnaître les événements de chaque objet.

Pour la réalisation de l'objectif de cette thèse, une nouvelle approche de compréhension de la vidéo pour l'apprentissage et la reconnaissance des événements en général est proposée. Cette approche implique un cadre complet pour l'apprentissage des événements qui comprends les tâches de segmentation d'images vidéo, de classification des objets, de suivi des objets, et d'apprentissage des événements:

1. En premier lieu, pour chaque frame de la vidéo, une tâche de segmentation consiste à détecter les régions mobiles, lesquelles sont représentées par des boîtes englobantes qui les délimitent.

2. En deuxième lieu, une nouvelle méthode de classification 3D associe à chaque région mobile un label de la classe d'objet (par exemple, personne, voiture) et un parallélépipède 3D décrit par sa largeur, sa hauteur, sa longueur, sa position, son orientation, et des mesures de fiabilité associées à ces attributs.

3. En troisième lieu, une nouvelle approche de suivi d'objets multiples utilise ces descriptions d'objet pour générer des hypothèses de suivi par rapport aux objets évoluant dans la scène.

4. En dernier lieu, une nouvelle approche d'apprentissage incrémental d'événements agrège en ligne les attributs et l'information de fiabilité des objets suivis afin d'apprendre des concepts qui décrivent les événements se déroulant dans la scène. Des mesures de fiabilité sont utilisées pour focaliser le processus d'apprentissage sur l'information la plus pertinente. Simultanément, l'approche d'apprentissage d'événements reconnaît des événements associés aux objets suivis dans la scène.

La suivante Section 1.1 présente les hypothèses et les objectifs de ce travail de thèse. Ensuite, la section 1.2 décrit la structure de cette thèse, où une brève description du contenu de chaque chapitre est présenté.

C.1 Hypothèses et Objectifs de la Thèse

L'approche proposée prend les hypothèses suivantes:

- **Application Mono-caméra:** L'approche a été conçu pour considérer une seule caméra comme entrée. Cette approche fait une estimation des informations 3D des objets physiques qui évoluent dans la scène, en utilisant les connaissances a priori sur les objets qui devraient être présents dans la scène. Même si la contrainte de mono-caméra semble très restrictive, dans les applications du monde réel, il est souvent le cas de traiter séparément les caméras d'un grand réseau.

- **Hypothèse de caméra fixe:** L'approche considère une configuration de caméra fixe. Cette hypothèse implique la disponibilité d'un modèle de transformation de référentiel image 2D à un référentiel de points 3D dans la scène. Le processus de recherche de cette transformation est connu dans la domaine du traitement de la vidéo comme la *calibration*. Dans le cadre de cette thèse, un modèle de caméra *pin-hole* est utilisée, lequel considère la correspondance entre les points d'image 2D et les points 3D de la scène comme une transformation linéaire représenté par une matrice de projection. Pour l'exécution du processus de calibration, un processus off-line appelé l'algorithme de transformation linéaire directe (DLT) [Abdel-Aziz and Karara 1971] est utilisé. DLT consiste à trouver la matrice de projection par la résolution du problème linéaire $X = AY$, où chaque colonne $x_k \in X$ correspond à un point 2D dans l'image, chaque colonne $y_k \in Y$ correspond au point 3D dans la référentielle de la scène, et A correspond à la transformation à trouver. La matrice de projection mentionné est souvent appelée la matrice perspective.

- **Modèles 3D d'objets disponibles:** Cette hypothèse est plus souhaitable que obligatoire, car la disponibilité de modèles 3D d'objets permet aux différentes tâches de l'approache d'effectuer une meilleure analyze de l'évolution des objets dans la scène. La disponibilité de modèles 3D d'objet permet à la tâche de classification de nourrir le processus de suivi avec une description plus précise des objets mobiles présents dans la scène, permet à la tâche de suivi des objets de réaliser un analyse plus détaillée des configurations possibles pour le suivi des objets, et permet à la tâche d'apprentissage des événements d'apprendre à partir des attributs le plus intéressants de l'objet.

- **Applications du monde réel:** L'application de l'approche doit être adaptée pour apprendre des événements à partir de la vidéo. Cette aptitude implique que plusieurs facteurs doivent être considérés:

  - *Qualité de la séquence vidéo:* La qualité de la séquence vidéo analysée doit être suffisante pour détecter l'évolution des objets dans la scène avec un niveau acceptable de fiabilité. Un niveau excessif de bruit dans la vidéo, une taux trop faible d'acquisition des images vidéo, ou un gros manque de contraste entre les objets et l'arrière-plan de la scène, parmi d'autres, peuvent être les facteurs qui empêchent la bonne détection d'un objet. Cette contrainte ne signifie pas que l'intérêt est uniquement centré sur des séquences vidéo haute définition et qualité. Tout au contraire, cette contrainte signifie que des mécanismes sont prévus pour contrôler plusieurs de ces facteurs si leurs conséquences dans la séquence vidéo ne sont pas graves.

- *Niveau de la surpopulation:* Le nombre d'objets qui peuvent évoluer simultanément dans la scène n'est pas limité, mais il est un fait que cela peut affecter les performances, et alors celui est un aspect à prendre en considération. La divisibilité des objets évoluant dans la scène est un facteur plus important, car l'approche a besoin de l'information d'événements pour chaque objet individuellement. Ce facteur ne signifie pas que l'occlusion dynamique entre des objets ne peut pas se produire. Tout au contraire, ce facteur signifie que des mécanismes existent dans l'approche pour faire face à l'occlusion. Ces mécanismes fonctionneront correctement en fonction de la fiabilité obtenue pour les attributs de l'objet dans les images précédentes.
- *Des performances en temps réel:* La performance en temps réel est un facteur souhaitable dans l'approche proposé. Plusieurs aspects peuvent empêcher le cadre de l'accomplissement de ce facteur, comme par exemple un nombre excessif des objets évoluant dans la scène, une très haute précision démandée pour les attributs d'objets, ou un très grand nombre de classes d'objets possibles. Selon si une application a besoin ou non d'une réponse en ligne de l'approche, ce facteur devient plus ou moins souhaitable.

Compte tenu de la complexité du problème à résoudre, ce travail de thèse tente de répondre à plusieurs questions d'ordre général:

1. **Comment faire pour diminuer l'écart entre les tâches bas niveau de traitement vidéo et l'apprentissage des événements?** Actuellement, la reconnaissance et l'apprentissage des événements complexes en général est réalisé en utilisant des événements d'intérêt basiques pré-définis par l'utilisateur. Lorsque l'intérêt est également porté dans l'apprentissage de ces événements basiques, les études ont centré leur attention en des types d'événements en particulier (par exemple, les trajectoires).

2. **Comment des événements génériques fréquents survenus dans une scène peuvent être appris et reconnus en ligne, en gardant une performance en temps de calcul suffisante pour des applications du monde réel?**

3. **Comment les informations nécessaires pour l'apprentissage d'événements peuvent être extraites à partir de vidéos bruitées d'une façon robuste?**

Pour répondre à ces questions, l'approche proposée établit deux objectifs globaux:

1. **Proposer une approche générale pour l'apprentissage des événements fréquents, capable de fonctionner correctement dans des applications du monde réel.** À cette fin, une approche d'apprentissage incrémental est proposé afin d'être capable d'apprendre en ligne des événements simples, directement de l'information des attributs des objets mobiles, avec un minimum de temps de traitement pour l'apprentissage lorsque de nouvelles informations arrivent dans le système. Les événements appris peuvent être utilisés pour réduire l'écart entre

les tâches bas niveau de traitement vidéo et l'analyse haut niveau des événements complexes pour des événements génériques, en considérant ces événements simples comme des éléments qui peuvent être une partie des événements plus complexes.

2. **Proposer une approche d'apprentissage capable de traiter l'information bruité d'une façon robuste.** Pour atteindre cet robustesse, une approche complet a été proposé, qui utilise des mesures de fiabilité pour mesurer la qualité et la cohérence des données acquises. La fiabilité des informations est associé aux attributs des objets suivis, et calculées pour les différentes tâches de l'approche.

Ainsi, la contribution de cette approche sont les suivants:

1. **Une nouvelle approche d'apprentissage incrémental des événements capable d'apprendre la fréquence des événements génériques à partir d'une séquence vidéo.** Cette approche propose un lien automatique entre les données de bas niveau obtenues à partir des objets qui évoluent dans la scène et des informations de plus haut niveau qui considèrent l'aspect temporel. L'apprentissage incrémental des événements peut être utile pour la reconnaissance des événements anormaux et sa sortie peut servir comme entrée pour des analyses de plus haut niveau.

2. **Une nouvelle façon de gérer l'information bruité.** L'approche propose d'associer des mesures de fiabilité à l'information obtenue, afin d'être en mesure de comptabiliser la qualité, la cohérence et la fiabilité de cette information. De cette façon, les informations les plus valables peuvent être identifiées afin d'augumenter la robustesse de la suivi, en concentrant l'attention du processus de suivi d'objets sur les attributs les plus cohérents et précis, et d'orienter le processus d'apprentissage sur les informations les plus fiables.

C.2 Structure de la Thèse

En premier lieu, le chapitre 2 décrit l'état de l'art lié à l'approche proposé. Comme l'approche aborde plusieurs aspects liés au domaine de compréhension de la vidéo, ce chapitre a été séparé en cinq sous-parties portant sur: la représentation des objets, la suivi multi-objet, l'utilisation des mesures de fiabilité dans le domaine de compréhension de la vidéo, apprentissage incrémental des concepts, et l'apprentissage des événements à partir de la vidéo .

En deuxième lieu, le chapitre 3 présente une vue globale de l'approche proposée, en donnant une description détaillée du problème à résoudre. Ce chapitre donne une description générale de l'approche. Aussi, les solutions proposées pour résoudre les problèmes présents à chaque tâche de l'approche sont mis en place. Les possibilités d'interaction de l'utilisateur avec l'approche sont également décrites. Les trois chapitres suivants donnent une description détaillée de chaque tâche de l'approche proposée.

Dans le chapitre 4, la représentation d'objets utilisée est décrite en détail. Cette description comprend la formulation mathématique du modèle de parallélépipède, le calcul des différents modèles alternatifs, la détection des situations d'occlusion statique, et la validation de la représentation de son utilisation dans des applications du monde réel.

Dans le chapitre 5, l'approche de suivi multi-object est décrite en détail. Cette description comprend un cadre pour la modélisation des hypothèses, l'algorithme de suivi et des méthodes de génération d'hypothèses.

Dans le chapitre 6, l'algorithme pour l'apprentissage et la reconnaissance des événements proposée est décrit en détail. Cette description comprend la definition de l'entrée, la représentation des états et des événement, et l'algorithme incrémental pour la reconnaissance et l'apprentissage des événements.

Après, le chapitre 7 présente l'évaluation de l'approche proposée. L'évaluation pour les tâches de classification et suivi ont été également effectués. Une évaluation complète de l'approche a été réalisée, tenant en compte de différents aspects comme la capacité d'apprentissage et de reconnaissance des événements, le temps de traitement, ainsi que l'influence des mesures de fiabilité, entre autres études.

En dernier lieu, le chapitre 8 présente les conclusions de ce travail de thèse et les perspectives de recherche futures pour les différentes contributions émanant de ce travail.

Appendix D

Conclusion: Version Française

L'objectif de cette thèse de proposer une approche pour l'apprentissage des événements en général dans des applications du monde réel a été atteint. Une nouvelle approche a été proposé, qui est capable d'apprendre de façon incrémentale une description générale des événements qui se produisent dans une séquence vidéo. La nature incrémentale du processus d'apprentissage des événements est bien adapté pour les applications du monde réel, car il considère l'intégration de nouvelles informations qui arrivent avec un minimum de temps de traitement. L'apprentissage incrémental des événements peut être utile pour la reconnaissance des comportements anormals et peut servir comme entrée pour des analyses de plus haut niveau.

Traiter des applications du monde réel implique également que l'approche doit être capable de gérer correctement les informations extraites de vidéos bruités. Cette exigence a été considérée, en proposant un mécanisme générique permettant de mesurer de manière cohérente la fiabilité de l'information dans l'ensemble du processus de compréhension vidéo. Plus concrètement, des mesures de fiabilité associées aux attributs des objets ont été proposées afin de mesurer la qualité et la cohérence de cette information.

L'approche est un cadre complet pour l'apprentissage des événements, y compris les tâches de segmentation des images vidéo, classification des objets, suivi des objets, et d'apprentissage des événements. Cette approche a proposé une passerelle automatique entre les données bas niveau obtenues à partir des objets qui évoluent dans la scène et des informations de plus haut niveau qui considèrent l'aspect temporel.

Les chapitres suivantes présentent la conclusion de chaque tâche de l'approche. La section D.1 présente la conclusion de la méthode de classification d'objets proposée. Ensuite, la section D.2 conclut sur l'approche de suivi d'objets proposée. Après, la section D.3 présente la conclusion de la nouvelle méthode d'apprentissage incrémentale des événements. Enfin, la section D.4 présente les limitations et les travaux futurs de l'approche.

D.1 À propos de la Classification d'Objets

La méthode de classification proposée est appropriée pour des applications du monde réel, pour plusieurs raisons:

- La méthode a montré une haute performance en temps de traitement pour des scénarios de complexité modérée.
- Les résultats de classification sont indépendants par rapport à la vue de la caméra et à l'orientation de l'objet. La méthode a donc une flexibilité appropriée pour être utilisée dans diverses applications du monde réel.
- La méthode est capable de faire face à des situations d'occultation statique sévères.
- L'approche propose des méthodes pour résoudre des situations ambigus entre plusieurs alternatives géométriquement plausibles. item Le modèle de parallélépipède utilisé par la classification est capable de représenter une grande variété d'objets, même ceux qui changent de posture, avec des valeurs acceptables pour les attributs 3D. Ce modèle simple permet également aux utilisateurs de facilement définir de nouveaux objets mobiles qui peuvent être présents dans la scène.
- Les mesures de fiabilité visuelle ont été proposées pour les attributs du modèle de parallélépipède pour mesurer le degré de visibilité de ces attributs. Ces mesures ont été utilisées par la tâche de suivi d'objets pour orienter l'estimation des attributs de un objet utilisant les informations les plus fiables. L'estimation de ces mesures est la première étape d'estimation de la fiabilité de l'information dans l'ensemble de l'approche.

Les attributs 3D estimés pour le modèle de parallélépipède ont permis à l'approche de suivi d'effectuer un meilleur filtrage des hypothèses par l'évaluation de la cohérence de ces attributs dans le temps.

Les résultats de l'évaluation ont montré que la classification peut être intéressant par elle-même.

D.2 À propos du Suivi d'Objets

La méthode de suivi proposé présente des idées similaires à la structure pour la création, la production, et l'élimination des hypothèses des objets mobiles par rapport aux méthodes MHT. Les principales différences de ces méthodes sont induites par la représentation de l'objet utilisé pour le suivi et le fait que cette représentation diffère de la représentation normalement utilisée dans les méthodes MHT. L'utilisation d'une représentation différente d'une représentation de point implique l'examen de la possibilité que plusieurs morceaux visuelles peuvent être associées à un objet mobile. Cela implique la conception de nouvelles

méthodes de création et de mise à jour des hypothèses pour un objet.

L'approche de suivi propose un nouveau modèle de dynamique de suivi d'un objet qui permet une redondance dans l'information de suivi par les attributs 2D et 3D de l'objet, afin d'accroître la robustesse. Ce modèle dynamique intègre des mesures de fiabilité pour chaque attribut de l'objet suivi, qui représente la qualité et la cohérence des informations utilisées. Le calcul de ces attributs considère une fonction d'oublie (ou fonction de refroidissement) pour renforcer les informations plus actuelles. Les mesures de fiabilité sont utilisées pour le contrôle de l'incertitude dans les informations obtenues, l'apprentissage plus robuste des attributs d'objets et obtenir une estimation de la qualité des informations obtenues. Ces mesures de fiabilité sont aussi utilisées dans la tâche d'apprentissage d'événements afin de déterminer les informations les plus valables à apprendre.

La méthode de suivi proposée a montré qui est capable d'avoir une haute performance en temps de traitement pour des séquences de complexité modérée. Cependant, rien ne peut encore être dit pour des situations plus complexes. L'approche a également montré sa capacité pour résoudre des problèmes d'occultation statique, de sous-segmentation, et de segmentation de un objet par plusieurs morceaux. La capacité de résolution problème d'occultation dynamique a montré des limitations qui sont décrites dans la section D.4. Plusieurs caractéristiques de la approche de suivi proposée pointent à l'objectif d'obtenir une performance en temps de traitement qui puisse être considéré comme approprié pour des applications du monde réel:

- L'approche de suivi coopère explicitement avec le processus de classification d'objets, guidant le processus de classification avec les attributs d'objets mobiles appris antérieurement. De cette façon, le processus de suivi est en mesure d'indiquer un point de départ et les limites de la recherche pour les attributs du parallélépipède à trouver par le processus de classification. Cette coopération permet une réduction considérable du temps de traitement dédié à la classification 3D. Comme l'information du mobile peut devenir plus fiable avec l'arrivée de plus des evidences visuelles disponibles, la coopération peut également être envisagée pour améliorer la qualité de l'information dans le temps, car plus de fiabilité implique une plus précis modèle dynamique du mobile et moins de la variabilité des attributs du mobile, ce qui permet d'établir de limites plus strictes à la espace de recherche.

- Quand un objet mobile passe au mode *rassuré*, une encore meilleure performance peut être obtenue par le processus de classification 3D, car le parallélépipède est estimé seulement pour une classe d'objet. À l'autre extrême, lorsque l'information est encore peu fiable pour effectuer la classification 3D, les attributs 2D du mobile sont seulement mis à jour, comme un moyen d'éviter les calculs provisoires de mauvaise qualité pour les attributs des mobiles.

- La détermination des *ensembles de blobs impliqués* permet de contrôler le nombre d'associations de blob possibles pour un objet mobile et de séparer le problème

de suivi en sous-problèmes en fonction de la proximité des blobs. Alors, la determination des *ensembles de blobs impliqués* présente une double contribution au control de l'explosion combinatoire, car le moins possible des associations par mobile et le moins mobiles liés par le sous-problème de suivi impliquent une réduction immédiate du nombre d'hypothèses à générer, ce qui contribue à l'amélioration de la performance en temps de traitement.

- Le nouveau processus de mise à jour des hypothèses a été orienté à optimiser l'estimation des ensembles des hypothèses, en vue d'obtenir le plus de chances d'éviter de générer des hypothèses peu probables qui doivent être éliminées plus tard. La nouvelle méthode de génération de configurations possibles pour les mobiles utilise un principe similaire, créant la solution la plus proche de l'estimation des attributs des mobiles selon les évidences visuelles disponibles, et puis générant les autres configurations possibles des mobiles à partir de cette première solution. Ainsi, la production est orientée sur l'optimisation de la performance en temps de traitement en générant des solutions de bonne qualité, plutôt que de générer toutes les combinaisons possibles et après de supprimer des solutions de mauvaise qualité.

- Même si la mise à jour des hypothèses est porté à générer le minimum possible des hypothèses, la charge de traitement pour l'image suivante peut être réduite par un filtrage des hypothèses superflues, inutiles, ou peu probables.

- Enfin, le processus de séparation des ensembles des hypothèses représente un autre mécanisme permettant d'améliorer la performance en temps de traitement, car il permet de réduire immédiatement le nombre de mobiles dans un même ensemble des hypothèses, générant ensembles des hypothèses qui peuvent être considérés comme sous-problèmes de suivi indépendents.

L'estimation des mesures de fiabilité dans l'approche de suivi a un impact direct dans la tâche d'apprentissage d'événements, car l'approche de suivi donne à la tâche d'apprentissage d'événements les éléments nécessaires pour déterminer les attributs les plus valables à apprendre.

Les résultats sur le suivi d'objets ont montré d'être réellement compétitif par rapport à d'autres méthodes de suivi dans des vidéos de référence. Cependant, il ya encore du travail à faire dans la capacité de l'approche pour faire face aux situations d'occultation.

D.3 À propos de l'Apprentissage d'Événements

L'approche d'apprentissage des événements proposée a été conçu pour être en mesure d'apprendre les concepts des états et des événements d'une manière générale. La définition de multiples contextes d'apprentissage dote le processus d'apprentissage d'un mécanisme flexible pour l'apprentissage des événements survenant dans une séquence vidéo. Selon la disponibilité sur les attributs des objets suivis, les combinaisons possibles pour des contextes d'apprentissage est énorme. Les attributs déjà proposés dans l'approche de

suivi d'objets donnent suffisamment de flexibilité pour explorer une grande variété de scénarios. Quoi qu'il en soit, les utilisateurs peuvent toujours définir plus des attributs d'objets, soit en combinant les attributs existants ou en créer de nouveaux attributs à partir de nouveaux descripteurs d'objet.

Pour effectuer le processus d'apprentissage une nouvelle approche d'apprentissage incrémental des événements appelé MILES (méthode d'apprentissage incrémental des événements et des états) a été proposée. La nature incrémentale de MILES, permet d'obtenir une performance d'apprentissage qui peut être utilisée dans l'apprentissage en ligne.

La principale contribution de MILES est l'utilisation des modèles d'apprentissage incrémental des concepts pour apprendre les états comme une hiérarchie de concepts et d'étendre la hiérarchie d'apprentissage incrémental des concepts pour apprendre des événements comme les relations temporelles de premier ordre entre les états appris. L'extension vers l'd'apprentissage des événement a impliqué la redéfinition des operateurs de *merge* et de *split* utilisés pour modifier la structure de la hiérarchie.

Une autre contribution est la façon d'utiliser les concepts de *cutoff* et d'*acuity*. Avant, ces concepts ont été traités comme des paramètres généraux d'un algorithme d'apprentissage incrémental des concepts, et maintenant, l'*acuity* est utilisée comme un moyen de définir la différence dans un attribut à être considérée comme intéressante dans un contexte d'apprentissage, et le *cutoff* comme une fonction des valeurs d'*acuity* et les différences pour les attributs symboliques analysés.

Cette approche a démontré sa capacité de reconnaître des événements, à partir des données bruitées au niveau des images, et avec un minimum d'effort de configuration. Les multiples extensions et applications possibles de cette approche sont encourageants pour explorer le comportement de l'approche dans des différents scénarios et contextes d'apprentissage.

D.4 Limitations de l'Approche et Travail Futur

La nature générale de l'approche proposée permet qu'elle puisse être étendue à un grand nombre de nouvelles études. Le but de cette section est d'analyser les travaux futurs de l'approche, comme des extensions de l'approche et des solutions possibles à ses limitations. Ces limitations sont organisées en fonction de la période de temps pour les résoudre (des limitations à court terme et à long terme).

D.4.1 Court Terme

À court terme, l'approche peut être extendu de plusieurs façons:

1. Le calcul des mesures de fiabilité dans la tâche de segmentation peut être une extension intéressante de l'approche. Ces mesures de fiabilité pourrait être associées

aux régions mobiles détectés afin de tenir compte de la qualité de la segmentation en fonction de l'influence de changements d'illumination, le niveau de contraste entre les objets en mouvement et le fond de la scène, et la possibilité de la présence d'ombres, entre autres aspects.

2. Les mesures de fiabilité proposées pour les attributs des objets ont été arbitrairement définis dans cette approche. Une analyse plus approfondie sur des différentes mesures de fiabilité peut être réalisé en vue d'établir les mesures qui permettent de mieux représenter la qualité ou la cohérence des attributs des objets.

En plus du travail futur présenté, chaque tâche de l'approche présente ses propres limitations et travail futur. Les sections suivantes sont consacrés à analyser ces limitations et à proposer le travail futur à court terme pour les tâches de classification d'objets (Section D.4.1.1) et d'apprentissage d'événements (Section D.4.1.2).

D.4.1.1 Sur la Classification d'Objets

Le travail futur relatif à la classification d'objets dans le court terme peut être résumé comme suit:

1. La résolution du problème de calcul du parallélépipède présenté dans la section 4.1.1 a été formulé pour une position du point focal plus élevée par rapport aux objets évoluant dans la scène. Un objet plus élevé que le point focal se traduira en une erreur dans le calcul des parallélépipèdes possibles associées à un objet. Cette situation ne peut pas être considérée comme une erreur, mais comme un élément manquant de l'approche qui n'a pas encore été résolu. La solution de ce problème implique la résolution d'un nouveau système d'équations pour couvrir cette situation. Faute de temps, ce système d'équations n'a pas été résolu pendant cette thèse, et peut être considéré comme de travail futur.

2. Les tests réalisés pour la tâche de classification d'objets ont montré un manque de précision dans l'estimation de l'angle d'orientation α des objets. Du travail futur peut pointer à l'utilisation d'une représentation d'objet alternative, lorsque cette situation est détectée.

D.4.1.2 Sur l'Apprentissage d'Événements

Le travail futur relatif à l'apprentissage d'événements dans le court terme peut être résumé comme suit:

1. Dans cette thèse, seulement quelques contextes d'apprentissage ont été utilisées. La flexibilité dans la définition des contextes d'apprentissage permet considérer des possibilités infinies pour ces contextes. Le travail futur peut se concentrer sur l'étude des différents contextes d'apprentissage.

2. Les mesures de fiabilité utilisées dans l'approche d'apprentissage d'événements sont définis en fonction de l'intérêt de l'utilisateur. À l'avenir, des différentes façons de définir ces mesures de fiabilité peuvent être envisagées.

3. En plus des opérateurs de *merge* et de *split* utilisés par l'approche d'apprentissage d'événements, d'autres opérateurs pourraient être intégrées à l'approche, comme les opérateurs proposés par l'algorithme d'apprentissage INC présenté dans la section 2.4.4.

D.4.2 Long Terme

À long terme, l'approche peut être extendu de plusieurs façons:

1. Le système de coopération mutuelle proposé entre les tâches de classification et de suivi peut être considérée comme une première étape dans la coopération entre des différentes tâches du processus de compréhension de la vidéo. Un autre point intéressant de coopération peut être un processus de rétroaction entre les tâches de suivi et de segmentation. Les informations fournies par l'approche de suivi peut être utilisées par la tâche de segmentation pour attirer l'attention sur les zones de l'image vidéo où le mouvement peut être plus susceptible de se produire. Ainsi, la segmentation peut se concentrer dans l'analyse du mouvement dans les zones d'entrée de la scène et dans les zones où les objets en mouvement ont été détectés, dans le but d'améliorer la performance en temps de traitement de la tâche de segmentation.

2. L'idée d'avoir deux niveaux de représentation pour un objet mobile, dans le plan image 2D et dans le référentiel 3D de la scène vidéo, conduit à la possibilité d'examiner simultanément d'autres représentations des objets évoluant dans la scène. Ces multiples modèles peuvent permettre à l'approche d'utiliser les informations les plus fiables à partir de différentes représentations. En même temps, ces observations pourraient être calculés ou non en fonction de la disponibilité et la pertinence de l'obtention de cette information. Par exemple, un modèle articulé d'une personne pourrait être intéressant d'être calculé si la proximité de l'objet à la caméra est suffisante pour apprécier ses parties, ou un modèle basé sur l'apparence de couleur pourrait être intéressant d'être calculé si le niveau de contraste de l'objet à l'égard de l'arrière-plan est suffisant pour obtenir des informations valables.

3. L'approche a été évaluée en utilisant une seule caméra. Des approches multi-caméra pourrait être étudiées afin d'analyser comment ces techniques pourraient améliorer l'estimation des attributs 3D.

4. Les modèles 3D utilisés pour la détermination de la classe et les attributs 3D d'un objet ont été pré-définies. L'utilisation de techniques d'apprentissage pour apprendre ces modèles d'objet pourrait être un intéressant sujet d'étude.

En plus du travail futur présenté, chaque tâche de l'approche présente ses propres limitations et travail futur. Les sections suivantes sont consacrés à analyser ces limitations et à proposer le travail futur à long terme pour les tâches de classification d'objets (Section D.4.2.1), de suivi d'objets (Section D.4.2.2), et d'apprentissage d'événements (Section D.4.2.3).

D.4.2.1 Sur la Classification d'Objets

Le travail futur relatif à la classification d'objets dans le long terme peut être résumé comme suit:

1. Même si la représentation des objets proposé sert pour décrire une grande variété d'objets, le résultat de l'algorithme de classification est une description grossière de l'objet. Afin d'évoluer dans l'interprétation des situations plus complexes, des modèles plus détaillées et plus spécifiques à la classe objet pourraient être utilisés en cas de besoin. Le travail futur peut pointer à l'utilisation de représentations d'objets plus spécifiques selon l'application, comme par example des modèles articulés, le contour d'un objet, ou les modèles d'apparence, entre autres.

2. L'approche de classification a a été proposée pour un modèle de caméra *pin-hole*. L'adaptation de la méthode de classification d'objets pour d'autres modèles de calibration, comme le modèle de distorsion radiale, peut être un intéressant sujet d'étude.

D.4.2.2 Sur le Suivi d'Objets

Le travail futur relatif à la suivi d'objets dans le long terme peut être résumé comme suit:

1. L'approche de suivi est en mesure de faire face à l'occultation dynamique en utilisant les attributs d'un objet estimés dans les frames précédentes pour estimer les valeurs actuelles des attributs de l'objet. Comme l'approche de suivi seulement fait une estimation des valeurs actuelles des attributs fondé sur des informations antérieures, le comportement des objets au cours de la période d'occultation ne peut pas être déterminé, ce qui peut conduire à des erreurs de suivi. Alors, l'approche de suivi proposée est en mesure de faire face aux situations d'occultation dynamique où les objets concernés maintient la cohérence dans le comportement observé précédente à la situation d'occultation. Le travail futur peut pointer à l'utilisation des modèles d'apparence, utilisés dans ces situations de façon pertinante afin de déterminer quelle partie des evidences visuelles appartient à chaque objet.

2. La méthode de suivi n'est pas capable d'identifier qu'un objet qui quitte la scène vidéo et le même objet à la ré-entrée. Cela est dû que les informations utilisées pour le suivi sont purement géométriques. À l'avenir, l'utilisation des modèles d'apparence peut servir à identifier les objets qui retournent à la scène.

3. Même si le processus de génération d'hypothèses de l'approche de suivi a été optimisé, un grand nombre d'objets entrant en même temps dans la scène peut produire un grand nombre initial d'hypothèses sur la configuration des objets dans la scène, car aucune information n'est disponible sur les nouveaux objets entrant dans la scène. L'utilisation d'autres représentations d'objet peut également servir à mieux définir les hypothèses initiales pour les objets qui entrent dans la scène.

D.4.2.3 Sur l'Apprentissage d'Événements

Le travail futur relatif à l'apprentissage d'événements dans le long terme peut être résumé comme suit:

1. À partir de l'état de l'art sur la formation incrémentale de concepts, il peut être déduit que la distribution des concepts d'états et d'événements dans la hiérarchie générée peuvent dépendre, dans certaine mesure, de l'ordre de processement des instances d'état. Ceci signifie que différentes hiérarchies peuvent être obtenues auprès de différentes ordres de processement pour les mêmes instances. Le travail futur peut pointer à analyser l'influence de l'ordres de processement dans la qualité de la représentation.

2. Comme l'approche d'apprentissage utilise les informations relatives à l'évolution de chaque objet suivi dans la scène séparément, il ne semble pas être inhérent à l'approche de représenter des relations entre les objets suivis. Dans l'avenir, des extensions de la représentation hiérarchique d'états et d'événements notion pourraient être étudiées afin d'envisager explicitement la représentation des relations et interactions entre les objets.

3. Pour plusieurs applications, l'utilisateur peut être intéressé à l'analyse de la survenue d'événements pré-définis intéressants pour l'application. Le travail futur peut se concentrer dans la façon dont ces événements pré-définis peuvent être associés à la description hiérarchique de concepts d'états et d'événements obtenue.

4. Il peut être très intéressant d'étudier comment les hiérarchies obtenus peuvent servir comme entrée à des algorithmes de reconnaissance sémantique, comme des éléments de base pour la reconnaissance des événements composés. Des applications comme *data mining*, et *video retrieval* pourrait aussi utiliser les résultats de l'apprentissage en tant que données d'entrée.

5. Le potentiel de l'approche d'apprentissage dans des applications d'apprentissage des comportements humains et de reconnaissance de comportements anormals doit être étudié.

Bibliography

Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range Photogrammetry*, pages 1–18, Falls Church, VA, 1971. American Society of Photogrammetry.

R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the* 11^{th} *International Conference on Data Engineering*, pages 3–14, 1995.

A. Ali and J. K. Aggarwal. Segmentation and recognition of continuous human activity. *IEEE Workshop on Detection and Recognition of Events in Video*, pages 28–35, 2001.

A. Alomary and M. Jamil. A new approach of clustering based machine-learning algorithm. *Knowledge-Based Systems*, 19(4):248–258, 2006.

P. O. Arambel, J. Silver, J. Krant, M. Antone, and T. Strat. Multiple-hypothesis tracking of multiple ground targets from aerial video with dynamic sensor control. In I. Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition XIII. Proceedings of the SPIE.*, volume 5429 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 23–32, August 2004.

A. Avanzi, F. Brémond, and M. Thonnat. Tracking multiple individuals for video communication. In *Proceedings of the International Conference on Image Processing (ICIP01), Tessaloniki, Greece*, October 2001.

A. Avanzi, F. Brémond, C. Tornieri, and M. Thonnat. Design and assessment of an intelligent activity monitoring platform. *EURASIP Journal on Applied Signal Processing, Special Issue on "Advances in Intelligent Vision Systems: Methods and Applications"*, 14(8):2359–2374, 2005.

AVITRACK, 2002. European Research Project, http://www.avitrack.net .

D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, 1982. Chapter 8.

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

M. Ben-Ezra, S. Peleg, and B. Rousso. Motion segmentation using convergence properties. *Proceedings of APRA Image Understanding Workshop (IUW'94)*, 2:1233–1235, November 1994.

H. Benhadda, J. Patino, E. Corvee, F. Bremond, and M. Thonnat. Data mining on large video recordings. In *Veille Strategique Scientifique et Technologique VSST 2007*, Marrakech, Morocco, 21st-25th October 2007.

J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision (ECCV'92)*, pages 237–252, London, UK, 1992. Springer-Verlag.

M. Black, Y. Yacoob, and X. Ju. Recognizing human motion using parameterized models of optical flow. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, pages 245–269. Kluwer Academic Publishers, Boston, 1997.

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

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

M. Borg, D. Thirde, J. Ferryman, F. Fusier, V. Valentin, F. Brémond, and M. Thonnat. A real-time scene understanding system for airport apron monitoring. In *Proceedings of 2006 IEEE International Conference on Computer Vision Systems (ICVS 2006)*, New York, USA, January 5-7 2006. IEEE Computer Society.

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

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 2, page 2690, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

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

F. Brémond and M. Thonnat. Issues of representing context illustrated by video-surveillance applications. *International Journal of Human-Computer Studies Special Issue on Context*, 48:375–391, 1998b.

F. Brémond, N. Maillot, M. Thonnat, and T. V. Vu. Rr5189 - ontologies for video events. Technical report, Orion Team, Institut National de Recherche en Informatique et Automatique (INRIA), May 2004.

I. V. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals and objects. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2000)*, pages 140–149, New York, NY, USA, 2000. ACM.

J. Carbonell, editor. *MACHINE LEARNING. Paradigms and Methods.* MIT/Elsevier, 1990.

CARETAKER, 2006. European Research Project, http://sceptre.king.ac.uk/caretaker .

C. Carincotte, X. Desurmont, B. Ravera, F. Bremond, J. Orwell, S. A. Velastin, J. M. Odobez, B. Corbucci, J. Palo, and J. Cernocky. Toward generic intelligent knowledge extraction from video and audio: the eu-funded caretaker project. In *Proceedings of the Institution of Engineering and Technology Conference on CRIME AND SECURITY, Imaging for Crime Detection and Prevention (ICDP)*, pages 470–475, Savoy Place, London, UK, 13-14 June 2006.

C. Carpineto and G. Romano. Galois: An order-theoretic approach to conceptual clustering. In *Proceedings of 10th International Conference on Machine Learning*, pages 33–40, Amherst, 1993.

M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR06), Volume II*, pages 1615–1622, New York, NY, 17-22 June 2006a.

M. Chan, A. Hoogs, Z. Sun, J. Schmiederer, R. Bhotika, and G. Doretto. Event recognition with fragmented object tracks. In *Proceedings of The 18th International Conference on Pattern Recognition (ICPR 2006), Volume I*, pages 412–416, Hong Kong, 20-24 August 2006b.

M. T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with hmm. In *Proceedings of the* 17^{th} *International Conference on Pattern Recognition (ICPR'04), Volume 4*, pages 150–154, Washington, DC, USA, 2004. IEEE Computer Society.

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

I. Cox and S. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

R. Cucchiara, R. Melli, A. Prati, and L. D. Cock. Predictive and probabilistic tracking to detect stopped vehicles. In *Proceedings of Workshop on Applications of Computer Vision (WACV)*, pages 388–393, Breckenridge, USA, 4-7 January 2005a.

R. Cucchiara, A. Prati, and R. Vezzani. Posture classification in a multi-camera indoor environment. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 725–728, Genova, Italy, 11-14 September 2005b.

F. Cupillard, F. Brémond, and M. Thonnat. Tracking groups of people for video surveillance. In *Proceedings of the European Workshop on Advanced Video Based Surveillance Systems (AVBSS01), Kingston, United Kingdom*, September 2001.

M. Devaney and A. Ram. Dynamically adjusting categories to accommodate changing contexts. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94)*, volume 2, page 1441, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

E. Durucan and T. Ebrahimi. Change detection and background extraction by linear algebra. *Proceedings of the IEEE*, 89(10):1368–1381, October 2001.

E. Erzin, Y. Yemez, and A. M. Tekalp. Multimodal speaker identification using an adaptive classifier cascade based on modality reliability. *IEEE Transactions on Multimedia*, 7(5):840–852, 2005.

E. Erzin, Y. Yemez, A. M. Tekalp, A. Ercil, H. Erdogan, and H. Abut. Multimodal person recognition for human-vehicle interaction. *IEEE MultiMedia*, 13(2):18–31, April 2006.

E. A. Feigenbaum. An information processing theory of verbal learning. Technical report, The RAND Corporation Paper P-1817, October 1959.

E. A. Feigenbaum and H. A. Simon. Generalization of an elementary perceiving and memorizing machine. In *IFIP Congress 1962*, pages 401–406, 1962.

J. Fernyhough, A. Cohn, and D. Hogg. Constructing qualitative event models automatically from video input. *Image and Vision Computing*, 18(2):81–103, January 2000.

D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.

G. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1045–1062, October 1999.

G. Foresti and C. Regazzoni. A real-time model-based method for 3-d object orientation estimation in outdoor scenes. *IEEE Signal Processing Letters*, 4(9):248–251, September 1997.

S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99)*, pages 63–72, New York, NY, USA, 1999. ACM.

A. Galata, A. Cohn, D. Magee, and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *Proceedings of European Conference on Artificial Intelligence (ECAI 2002)*, pages 741–745, 2002.

J. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3):11 – 61, 1989.

J. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. In J. Carbonell, editor, *Machine Learning: Paradigms and Methods*, pages 11 – 61, Cambridge, MA, 1990. MIT Press.

B. Georis, M. Mazière, F. Brémond, and 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*, pages 46–50, February 2004.

GERHOME, 2005. Research Project, http://gerhome.cstb.fr .

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

M. Gluck and J. Corter. Information, uncertainty, and the utility of categories. In E. L., editor, *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, pages 283–287, New York, 1985. Academic Press.

S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proceedings of the* 9^{th} *IEEE International Conference on Computer Vision (ICCV 2003)*, page 742, Washington, DC, USA, 2003. IEEE Computer Society.

D. Goodall. A new similarity index based on probability. *Biometric*, 22:882–907, 1966.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, April 1984.

G. J. H. A survey of clustering methods. Technical report, ICS-TR-89-38, University of California, Irvine, Department of Information and Computer Science, October 1989.

M. Hadzikadic and B. F. Bohren. Learning to predict: Inc2.5. *IEEE Transactions on Knowledge and Data Engineering*, 9(1):168–173, 1997.

G. Haipeng. *Algorithm selection for sorting and probabilistic inference: a machine learning approach*. PhD thesis, Department of Computing and Information Sciences, College of Engineering, Kansas State University, 2003.

R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1031–1038, Washington, DC, USA, 2005. IEEE Computer Society.

J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In *Proceedings of the Second IEEE Workshop on Visual Surveillance*, pages 74–81, Fort Collins, Colorado, June 1999.

B. Heisele. Motion-based object detection and tracking in color image sequences. In *Proceedings of the Fourth Asian Conference on Computer Vision (ACCV2000)*, pages 1028–1033, Taipei, Taiwan, 8-11 January 2000.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99:)*, pages 50–57, New York, NY, USA, 1999. ACM.

S. Hongeng, F. Bremond, and R. Nevatia. Bayesian framework for video surveillance application. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR2000)*, pages Vol I: 164–170, Barcelona, Spain, 2000.

S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding (CVIU)*, 96(2):129–162, November 2004.

R. Howarth and H. Buxton. Conceptual descriptions from monitoring and watching image sequences. *Image and Vision Computing*, 18(2):105–135, January 2000.

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

W. Hu, D. Xie, and T. Tan. A hierarchical self-organizing approach for learning the patterns of motion trajectories. *IEEE Transactions on Neural Networks*, 15(1):135–144, 2004b.

W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (9):1450–1464, September 2006.

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

C. Hue, J.-P. L. Cadre, and P. Perez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791–812, July 2002b.

W. Iba. Learning to classify observed motor behavior. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI91)*, pages 732–738, Sidney, Australia, 1991.

W. Iba and P. Langley. Unsupervised learning of probabilistic concept hierarchies. In G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, editors, *Machine Learning and Its Applications*, volume 2049 of *Lecture Notes in Computer Science*, pages 39–70. Springer, 2001.

M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision (IJCV)*, 12(1):5–16, February 1994.

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

M. Isard and J. Maccormick. Bramble: a bayesian multiple-blob tracker. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, volume 2, pages 34–41, Vancouver, Canada, July 9-12 2001.

Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

R. Jain, D. Militzer, and N. H.-H. Separating non-stationary from stationary scene components in a sequence of real world tv images. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 612–618, 1977.

F. Jiang, Y. Wu, and A. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *Proceedings of the International Conference on Image Processing (ICIP07)*, volume 5, pages 145–148, San Antonio, TX, September 2007.

Y. Jin and F. Mokhtarian. Variational particle filter for multi-object tracking. In *International Conference on Computer Vision (ICCV'07)*, pages 1–8, Rio de Janeiro, Brasil, October 2007.

N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

P. Kelly, N. O'Connor, and A. Smeaton. Pedestrian detection in uncontrolled environments using stereo and biometric information. In *Proceedings of the 4^{th} ACM international workshop on Video surveillance and sensor networks (VSSN '06)*, pages 161–170, New York, NY, USA, 2006. ACM.

J. L. Kolodner. Maintaining organization in a dynamic long-term memory. *Cognitive Science*, 7:243–280, 1983.

D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *Proceedings of the British Machine Vision Conference (BMVC 2005)*, pages 1–10, Oxford, U.K., September 2005.

D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *Proceedings of the 18^{th} International Conference on Pattern Recognition (ICPR '06)*, pages 1187–1190, Washington, DC, USA, 2006. IEEE Computer Society.

S. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: a review of classification techniques. *Artificial Intelligence Review*, 26:159–190, 2006.

M. Kukar and I. Kononenko. Reliable classifications with machine learning. In *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, volume 2430 of *Lecture Notes In Computer Science*, pages 219–231, London, UK, 2002. Springer-Verlag.

T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications, chapter 3*, volume 1, pages 43–83, Norwood, MA, 1990. Artech House.

A. Lai, G. Fung, and N. Yung. Vehicle type classification from visual-based dimension estimation. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITS 2001)*, pages 201–206, 25-29 August 2001.

M. Lazarescu, S. Venkatesh, and G. West. Incremental learning with forgetting (i.l.f.). In *Proceedings of ICML-99 Workshop on Machine Learning in Computer Vision*, Slovenia, June 1999.

T. L. Le, M. Thonnat, A. Boucher, and F. Bremond. A query language combining object features and semantic events. In *The 14th International MultiMedia Modelling Conference (MMM)*, Kyoto, January 2008.

M. Lebowitz. Generalization from natural language text. *Cognitive Science*, 7(1):1–40, January 1983.

M. Lebowitz. Categorizing numeric information for generalization. *Cognitive Science*, 9 (3):285–308, 1985.

M. Lebowitz. Concept learning in a rich input domain: Generalization-based memory. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, Volume II*, pages 193–214, Palo Alto, CA, 1986. Tioga.

M. Lebowitz. Experiments with incremental concept formation: Unimem. *Machine Learning*, 2(2):103–138, 1987.

B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *Proceedings of the* 26^{th} *Pattern Recognition Symposium (DAGM'04)*, volume 3175 of *Springer LNCS*, pages 145–153, Tubingen, Germany, August 2004.

B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

C. Li and G. Biswas. Unsupervised learning with mixed numeric and nominal data. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):673–690, 2002.

E. Loutas, I. Pitas, and C. Nikou. Information theory-based analysis of partial and total occlusion in object tracking. In *Proceedings of the International Conference on Image Processing (ICIP2002)*, volume 2, pages 309–312, 2002.

Y. Ma, P. Buddharaju, and M. Bazakos. Pattern discovery for video surveillance. In G. Bebis, R. D. Boyle, D. Koracin, and B. Parvin, editors, *Proceedings of the First International Symposium on Advances in Visual Computing (ISVC 2005)*, volume 3804 of *Lecture Notes in Computer Science*, pages 347–354, Lake Tahoe, NV, USA, December 5-7 2005. Springer.

Y. Ma, M. Bazakos, B. Miller, and P. Buddharaju. Activity awareness: from predefined events to new pattern discovery. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)*, page 11, St. Johns University, Manhattan, New York City, New York, NY, USA, January 5-7 2006. IEEE Computer Society.

E. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, 2001.

J. D. Martin and D. O. Billman. Acquiring and combining overlapping concepts. *Machine Learning*, 16(1-2):121–155, 1994.

A. McIvor. Background subtraction techniques. In *Proceedings of the Conference on Image and Vision Computing (IVCNZ 2000)*, Hamilton, New Zealand, November 27-29 2000.

K. McKusick and P. Langley. Constraints on tree structure in concept formation. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pages 810–816, Sydney, Australia, 1991.

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

G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, August 2001.

R. S. Michalski and R. E. Stepp. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 331–363, Palo Alto, CA, 1983. Tioga.

T. M. Mitchell. *Version spaces: an approach to concept learning.* PhD thesis, Stanford University, Stanford, CA, USA, 1979.

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

B. Morris and M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8): 1114–1127, August 2008.

M. Mugurel, S. Venkatesh, and G. West. On the incremental learning and recognition of the pattern of movement of multiple labelled objects in dynamic scenes. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR2000)*, pages Vol II: 652–655, 2000.

K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning.* PhD thesis, University of California, Berkeley, 2002. Chair-Stuart Russell.

K. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–686, 1968.

A.-T. Nghiem, F. Brémond, M. Thonnat, and V. Valentin. Etiseo, performance evaluation for video surveillance systems. In *Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007)*, pages 476–481, London (United Kingdom), 5-7 September 2007.

J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proceedings of the British Machine Vision Conference (BMVC 2006)*, volume 3, pages 1249–1258, Edinburgh, Scotland, 4-7 September 2006.

P. Nordlund and J.-O. Eklundh. Maintenance of figure-ground segmentation by cue-selection. In *Proceedings of the First International Workshop on Cooperative Distributed Vision*, pages 93–123, Kyoto, Japan, 13-15 January 1997.

P. Nordlund and J.-O. Eklundh. Real-time maintenance of figure-ground segmentation. In *Proceedings of the First International Conference on Computer Vision Systems (ICVS'99)*, volume 1542 of *Lecture Notes in Computer Science*, pages 115–134, Las Palmas, Gran Canaria, Spain, 13-15 January 1999.

J. Owens and A. Hunter. Application of the self-organizing map to trajectory classification. In *Proceedings of the 3rd IEEE International Workshop on Visual Surveillance (VS2000)*, pages 77–83, Dublin, Ireland, 2000.

J. R. Parker. Gray level thresholding in badly illuminated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):813–819, August 1991.

K. R. Pattipati, R. L. Popp, and T. Kirubarajan. Survey of assignment techniques for multitarget tracking. In Y. Bar-Shalom and W. D. Blair, editors, *Multitarget-Multisensor Tracking: Advanced Applications, chapter 2*, volume 3, pages 77–159, Norwood, MA, 2000. Artech House.

J. Piater, S. Richetto, and J. Crowley. Event-based activity analysis in live video using a generic object tracker. In *Proceedings of The Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS02)*, pages 1–8, June 2002.

C. Piciarelli and G. Foresti. On-line trajectory clustering for anomalous event detection. *Pattern Recognition Letters*, 15:1835–1842, 2006.

C. Piciarelli, C. Micheloni, and G. Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1544–1554, November 2008.

T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *International Conference on Computer Vision (ICCV'07)*, Rio de Janeiro, Brasil, October 2007.

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

C. Regazzoni, G. Foresti, and A. Venetsanopoulos. Coding of noisy binary images by using statistical morphological skeleton. In *IEEE Workshop on Non Linear Signal Processing*, pages 354–359, Cyprus, Greece, 1995.

Y. Reich. Constructive induction by incremental concept formation. In Y. A. Feldman and A. Bruckstein, editors, *Artificial Intelligence and Computer Vision*, pages 191–204, Amsterdam, 1991. Elsevier Science Publishers.

Y. Reich and S. J. Fenves. The formation and use of abstract concepts in design. *Concept formation knowledge and experience in unsupervised learning*, pages 323–353, 1991.

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

P. Remagnino and G. Jones. Classifying surveillance events from attributes and behaviour. In T. F. Cootes and C. J. Taylor, editors, *Proceedings of the British Machine Vision Conference (BMVC 2001), Session 8: Modelling Behaviour*, pages 685–694, Manchester, UK, 10-13 September 2001. British Machine Vision Association.

R. Reulke, F. Meysel, and S. Bauer. Situation analysis and atypical event detection with multiple cameras and multi-object tracking. In G. Sommer and R. Klette, editors, *Proceedings of The Second International Workshop on Robot Vision (RobVis 2008)*, volume 4931 of *Lecture Notes in Computer Science*, pages 234–247, Auckland, New Zealand, 18-20 February 2008. Springer.

P. L. Rosin. Thresholding for change detection. *Computer Vision and Image Understanding*, 86(2):79–95, May 2002.

G. Scotti, A. Cuocolo, C. Coelho, and 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)*, volume 3, pages 880–883, Genova, Italy, 11-14 September 2005.

E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1582–1588, Washington, DC, USA, 2006. IEEE Computer Society.

L. Snidaro and G. L. Foresti. Real-time thresholding with euler numbers. *Pattern Recognition Letters*, 24(9-10):1533–1544, June 2003.

I. C. Society, editor. *IEEE International Series of Workshops on Performance Evaluation of Tracking and Surveillance (PETS)*. IEEE Computer Society, 2007. http://visualsurveillance.org.

C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceedings of the International Society for Optical Engineering (SPIE)*, volume 2235, pages 394–405, 1994.

L. Talavera and J. Béjar. Generality-based conceptual clustering with probabilistic concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:196–206, 2001.

K. Thompson and P. Langley. Concept formation in structured domains. In *Concept formation knowledge and experience in unsupervised learning*, pages 127–161, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

L. Torresani and C. Bregler. Space-time tracking. In *Proceedings of the 7th European Conference on Computer Vision (ECCV02)*, pages 801–812, 2002.

L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 493–500, Kauai, HI, USA, 8-14 December 2001. IEEE Computer Society.

L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, Cambridge, MA, 2004.

A. Toshev, F. Brémond, and M. Thonnat. Unsupervised learning of scenario models in the context of video surveillance. In *Proceedings of the IEEE International Conference on Computer Vision Systems (ICCV 2006)*, page 10, January 2006.

S. Treetasanatavorn, U. Rauschenbach, J. Heuer, and A. Kaup. Bayesian method for motion segmentation and tracking in compressed videos. In W. G. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *DAGM-Symposium*, volume 3663 of *Lecture Notes in Computer Science (LNCS) on Pattern Recognition and Image Processing*, pages 277–284. Springer, August/September 2005.

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

C.-J. Tsai, N. P. Galatsanos, and A. K. Katsaggelos. Maximum-likelihood optical flow estimation using differential constraints. In A. E. Çetin, L. Akarun, A. Ertüzün, M. N. Gurcan, and Y. Yardimci, editors, *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99)*, pages 53–56, Antalya, Turkey, 20-23 June 1999. Bogaziçi University Printhouse.

R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1986)*, pages 364–374, Miami Beach, FL, 1986.

R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.

C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition (CVPR 2001), volume 1, pages 511–518, Kauai, HI, USA, 8-14 December 2001.

T. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: a novel algorithm for temporal scenario recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI03), Acapulco, Mexico*, August 2003.

V. T. Vu, F. Bremond, G. Davini, M. Thonnat, Q. C. Pham, N. Allezard, P. Sayd, J. L. Rouas, S. Ambellouis, and A. Flancquart. Audio-video event recognition system for public transport security. In *Proceedings of IET Conference on Imaging for Crime Detection and Prevention (ICDP 2006)*, London, UK, June 2006.

P. Winston. Learning by managing multiple models. In P. Winston, editor, *Artificial Intelligence*, pages 411–422. Addison-Wesley Publishing Company, 1992.

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

A. Yilmaz, X. Li, and M. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

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

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

F. Ziliani and A. Cavallaro. Image analysis for video surveillance based on spatial regularization of a statistical model-based change detection. *Real-Time Imaging*, 7 (5):389–399, 2001.

N. Zouba, F. Bremond, M. Thonnat, and V. T. 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, March 2007.