Universitat
Autònoma
de Barcelona

Towards Robust Multiple-Target Tracking in Unconstrained Human-Populated Environments

A dissertation submitted by **Daniel Rowe** at Universitat Autònoma de Barcelona to fulfil the degree of **PhD in Computer Science**.

Barcelona, December 2007

Director:	**Juan José Villanueva Pipaón**
	Universitat Autònoma de Barcelona
	Computer Vision Centre
Co-director:	**Jordi Gonzàlez i Sabaté**
	UPC-CSIC
	Institut de Robòtica i Informàtica Industrial

Centre de Visió
per Computador

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Centre, Universitat Autònoma de Barcelona.

ISBN-13: 978-84-935251-5-6
ISBN-10: 84-935251-5-4

Printed by Ediciones Gráficas Rey, S.L.

“Siempre que enseñes, enseña a
la vez a dudar de lo que enseñas.”

—José Ortega y Gasset

Acknowledgements

Llegado este momento en el que voy a poner el punto y final a esta tesis —y a esta aventura utópica que constituye la investigación entendida como vocación— es de justicia emplear las últimas frases escritas para este documento a agradecer el esfuerzo, dedicación y apoyo a todos aquellos que han hecho posible el mismo, tanto en su esencia como en su contexto.

Así, en primer lugar, mi más sincero agradecimiento a Juanjo por darme la oportunidad de intentar este sueño, por embarcarme en el CVC para dirigirme una tesis, con la apuesta que ello conlleva. También por el trato campechano que dispensa, y por hacer posible que la ciencia sea una lugar de discusión, y no de dogma, aceptando entrar al trapo en los múltiples 'debates' en los que se convertían nuestras reuniones.

Quiero dedicar un muy afectuoso agradecimiento a Poal porque da gusto trabajar con él y tenerlo a la vez, o sobre todo, como amigo. Porque ha estado ahí, aguantando un chaparrón que no tenía por qué, cada vez que mi frustración por la imposibilidad de cambiar un sistema inamovible se escapaba buscando un desahogo. Y porque a pesar de la carga brutal de trabajo que continuamente lleva, siempre está dispuesto a emplear el tiempo que haga falta en ayudar en lo que pueda, solucionando problemas, revisando minuciosamente un documento, montando un artículo, repasando un código o encargándose directamente de las gestiones que haga falta. Es realmente grato que, en lugar de darte trabajo, te lo quiten.

Yo no entiendo la investigación si no es como un trabajo de equipo. Así, agradezco a todos aquellos que han hecho esto posible, aunque desgraciadamente en momentos más puntuales de lo que nos hubiera gustado. Por tanto, gracias a Natxo, compañero de dichas y desdichas, apoyo personal en nuestros problemas comunes, y en la investigación a través de sus magníficas 'ideas felices', o en su disposición a escuchar cualquier nueva idea en detalle. Y también, claro está, a todos con los que he tenido que la oportunidad de colaborar o intentar 'dirigir'. Esto último no es nada fácil cuando uno dista mucho de ser el experto en la materia que se supone que se tiene que ser para guiar a otro por ella. Y se agradece que se toleren con paciencia y comprensión errores o cambios de dirección a lo largo del desarrollo. Así, ha sido una grata y enriquecedora experiencia, aunque no exenta de sobresaltos, toda colaboración y dirección desde el primer proyecto de Sandra, pasando sobre todo por la dirección del proyecto fin de carrera y la tesina de Iván, y también llevar con Toni en el proyecto de anotación, con Josep el de optimización de código, colaborar con Pau algún artículo, y con Ariel, Xavi Roca y Migue y con todo ISE en general. También agradecer haber estado ahí a los compañeros que han pasado por el despacho, a Juan, Edu, Marçal, Aura, David Masip, Jaume...

Por supuesto, quiero darle las gracias a Montse, que con una sonrisa siempre en la cara y cargada de buen humor, te resuelve todo tipo de gestiones y horribles problemas burocráticos.

Hago extensivo estos agradecimientos a todo el resto del personal del CVC con los que he compartido estos años, tanto en la cafetería en el día a día, o en un congreso, o en la barbacoa y fiestas varias. Y especialmente a las niñas de administración, que tanta paciencia nos tienen.

Fuera de este mundillo que es el CVC, tengo que agradecer de manera muy especial a mi madre. Es evidente, y ella lo sabe, que sin su apoyo constante, sin esas charlas sin fin, y sin su pragmatismo yo no hubiera hecho tesis doctoral alguna, y probablemente ni siquiera sería ingeniero.

Gracias a mi otro gran punto de apoyo, mi padre. Y si a alguien debo este afán de búsqueda, la curiosidad, la crítica y ese intentar siempre perfeccionar el detalle, es sin duda a él.

Agradecer el apoyo incondicional de mi familia y colegas, que siempre han confiado, incluso a ciegas, en mí. Gracias a Em, a Nacho, a Concha, a Servando, a Luis, por supuesto a mi abuela, y a Javi, Gonzalo, Marta y a un largo etcétera.

Por último, gracias a Sol, la persona más importante en mi vida, y que ha estado aquí, en el día a día, aguantando y confortando, escuchando y apoyando sin límite. Sin ella, dudo mucho que hubiera llegado a buen puerto. Es, además, la persona que más ha sacrificado de su vida por esta tesis, desde venir a vivir a Barcelona cuando el cuerpo le pedía India o Kenya —mientras asumía la incertidumbre de un proyecto que no se sabe nunca cuando se acaba y está sujeto a múltiples prórrogas y retrasos— hasta aguantar que el trabajo de tu pareja te robe muchas veces tus momentos de ocio, y tu estado de ánimo, o a soportar la carga económica que supone una tesis. Muchas gracias, de verdad.

Abstract

Natural Vision Systems have reached incredible performances in detecting and tracking multiple moving objects simultaneously. Accurate and robust multiple-target tracking is also a key task in many promising Computer-Vision applications. Practical usages of proposed algorithms can now be tackled in real time thanks to recent technological advances. Further, this represents a huge challenge because of the numerous particular problems involved in such a task. Thus, proposals must deal with multiple highly non-rigid targets which move in an unforeseeable manner through unconstrained dynamic open-world scenarios.

In this thesis, a principled hierarchical architecture which fulfills multiple-target tracking is presented. Further, another tracking approach is previously developed and evaluated.

The first approach developed in this document focuses on tracking by means of *particle filtering*. In this case, the problem is formulated as a sequence of inferences with a temporal probability model by means of Bayesian filters. No assumption about linearity or gaussianity is made on the involved pdf's.

Although this paradigm presents some remarkable advantages, it has several important drawbacks. In this document, these are highlighted, and some ways of solutions are proposed, also to handle the aforementioned expected inherent problems. Thus, a new weight normalisation is used to cope with *sampling impoverishment* in a multiple target-tracking scenario. Dynamics updating and state estimation are well studied in order to deal with unknown target's dynamics, presumably highly non-linear. A method is presented to handle partial and complete occlusions by considering target predicted trajectories, and their likelihoods. Model drift is tackled by careful updating, based on the history of likelihood measures. A colour-based likelihood, computed from histogram similarity, is used. However, despite the great efforts spent on this approach, it still lacks from a robust performance due to the drawbacks of the particle filtering framework, and the inherent complexity involved in non-supervised multiple-human tracking.

Thus, a second approach is developed to tackle this complex open problem. A novel architecture inscribed in a principled framework is proposed. It follows in many ways a biological paradigm. A modular and hierarchically-organised system is designed. It is conformed by a detection level which feeds a two-level tracking subsystem. Co-operating modules, distributed through this architecture, work following both bottom-up and top-down approaches.

Contributions include both the architecture itself, and the development, improvement and integration of the different modules. The proposed architecture introduces the necessary synergies which allow the system to tackle such a problem as unconstrained multiple-target tracking. With respect to the different modules, the main focus is placed on high-level tracking algorithms. Since a careful analysis of motion events is a critical issue for tracking successful, a module for principled event management is proposed, and embedded in the system. Multiple-target interaction events, and a proper scheme for tracker instantiation and removal according to scene events, are considered. Thus, the system is allowed to switch among the two different operation modes implemented, namely motion-based tracking and appearance-based tracking. This entails another remarkable characteristic of the system: its ability to continuous and independently track numerous targets while they group and split. Multiple appearance models are built and constantly updated. A special attention is paid to maximise the discrimination between the target and potential distracters by means of an appropriate feature selection, and a wise combination of all available sources of information.

This tracking architecture works as a stand-alone application in a non-friendly, complex and dynamic scenario. No a-priori knowledge about either the scene or the targets, based on a previous off-line training period is needed. Hence, the scenario could be completely unknown beforehand. No camera calibration is required since tracking is achieved without the need of 3D information.

Successful tracking has been demonstrated in multiple sequences of both indoor and outdoor scenarios, from own and public well-known databases. Accurate and robust localisations have been yielded even during long-term target clustering and occlusions. Results are comprehensively analysed.

Keywords: *Multiple-target tracking; Trajectory analysis; Kalman filter; Particle filtering; Feature evaluation and selection; Probabilistic colour appearance models; Event management; Motion segmentation; Appearance-based tracking.*

Topics: *Image Processing; Computer Vision; Scene Understanding; Machine Intelligence; Machine Vision Applications; Video-Sequence Evaluation*

Resum

Los Sistemes de Visió Naturals (SVN) han assolit uns resultats increïbles pel que fa a la detecció i seguiment de múltiples objectes simultàniament en moviment. Aquest seguiment precís i robust de múltiples agents (objectes i persones) és també una tasca clau en moltes aplicacions prometedores basades en la Visió per Computador (VC). Los algorismes teòrics proposats durant estos últims anys poden ser ara aplicats a la pràctica i en temps real gràcies als últims avenços tecnològics. Nogensmenys, això ha representat ser un gran repte degut als nombrosos problemes que han anat sorgint durant el desenvolupament d'aquesta tasca, bàsicament pel fet d'haver de tractar amb múltiples persones, que són altament no-rígids, que es mouen d'una manera imprevisible per escenaris oberts, dinàmics i no-restringits.

En esta Tesis se presenta una arquitectura jeràrquica que realitza el seguiment de múltiples agents. A més, s'han desenvolupat i avaluat dues aproximacions teòriques al seguiment d'agents.

La primera aproximació desenvolupada en aquest document se centra en el seguiment basat en el *filtratge de partícules*. En aquest cas, el problema se formula com una seqüència d'inferències utilitzant un model probabilístic temporal a partir de filtres Bayesians. A més, no se fa cap assumpció sobre la linearitat o Gaussianitat de les pdf's involucrades.

Malgrat que aquest paradigma té avantatges remarcables, també té inconvenients importants. Així doncs, aquests inconvenients se ressalten en aquest document i se proposen vies de solució, també per a manegar els problemes inherents i per tant previstos, abans esmentats. Així, s'utilitza una nova normalització dels pesos de les partícules per a evitar el problema anomenat *empobriment del sampleig* que s'esdevé en escenaris on hi ha seguiment de múltiples agents. S'han estudiat les actualitzacions de les dinàmiques i les estimacions dels estats per tal de tractar les dinàmiques desconegudes dels agents, presumiblement altament no-linials. Com a resultat, se presenta un mètode per a manipular oclusions parcials i complertes, a partir de prediccions sobre les trajectòries dels agents i de la seva versemblança. La deriva del model emprat està contemplada a partir d'actualitzacions curoses basades en la història de les mesures de versemblança. Per a això, utilitzem un càlcul de la versemblança basada en color a partir de la similitud d'histogrames. Nogensmenys, malgrat tots els esforços emprats en aquest paradigma, el seu rendiment no és massa robust degut als inconvenients intrínsecs dels filtres de partícules i per la inherent complexitat involucrada en el seguiment no-supervisat de múltiples humans.

Per tant, s'ha desenvolupat una segona aproximació per afrontar aquest problema tan complex i obert. Es proposa una arquitectura nova inscrita en un marc estructurat. Segueix en molts aspectes un paradigma biològic: ha estat dissenyat un sistema modular i jeràrquicament organitzat que és format per un nivell de detecció que alimenta un subsistema de dos nivells. Los mòduls que cooperen, distribuïts a través d'aquesta arquitectura, funcionen seguint enfocaments tant de baix a dalt com de dalt a baix.

Les contribucions inclouen l'arquitectura en si i el desenvolupament, millora i integració dels diferents mòduls. L'arquitectura proposada introdueix les sinergies necessàries per permetre al sistema tractar el problema del seguiment de múltiples agents. Respecte als diferents mòduls, el focus principal es posa en els algoritmes de seguiment d'alt nivell. Ja que una anàlisi prudent dels esdeveniments de moviment és un assumpte crític per un seguiment d'èxit, es proposa un mòdul per a la gestió d'aquests esdeveniments, que està incrustat en el sistema. Així, es consideren els esdeveniments d'interacció entre múltiples agents, i un esquema propi per a la instanciació dels diferents algorismes de seguiment o la seva supressió segons els esdeveniments de l'escena. Així, lo sistema permet canviar-se entre dos modes diferents de funcionament implementats, és a dir, basat en lo moviment i basat en l'aparença. Això suposa una altra característica notable del sistema: la seva habilitat per seguir continua i independentment múltiples objectius mentre aquests s'agrupen i es separen. Es construeixen models d'aparença que s'actualitzen constantment. Es para una atenció especial per maximitzar la discriminació entre l'objectiu seguit i los distractors potencials, per mitjà d'una selecció de les característiques més apropiades i una combinació assenyada de totes les fonts d'informació disponibles.

Aquesta arquitectura de seguiment treballa com a aplicació autònoma en un escenari no amistós, complex i dinàmic. No es necessita cap tipus de coneixement a-priori al voltant de l'escena o dels agents, basat en un període d'entrenament previ. Per això, l'escenari podria ser completament desconegut per endavant. No s'exigeix cap calibració de càmeres ja que lo seguiment és aconseguit sense la necessitat d'informació 3D.

S'ha demostrat un seguiment correcte en múltiples seqüències de tant interiors com a l'aire lliure, de bases de dades pròpies i públiques molt conegudes. Han estat assolides unes localitzacions acurades i robustes fins i tot durant agrupaments llargs i oclusions. Los resultats obtinguts s'han analitzat extensa i completament.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Human beings, as well as a great diversity of animal species, have developed an amazing capability of processing complex and continuous varying visual stimuli. Millions of years of evolution have led to highly efficient visual systems which show, in such an apparently easy way, incredible performances.

The ability of motion detection must be undoubtedly mentioned among the most powerful faculties of Natural Visual Systems [68]. This skill is crucially involved in numerous critical issues for the survival of the species, such as in tracking moving objects, despite partial occlusions and drastic illumination changes; in extracting the depth structure of the world by taking advantage of the motion parallax; in detecting objects which are camouflaged in a cluttered background of similar colour and texture; or in recognising objects from the relative motion of their parts. Moreover, this faculty is implied in several associated motor responses such as the stabilisation of the gaze, or the control of limbs.

Thus, the ability of perceiving the motion of potential predators and preys has been unavoidable linked to self-motion capabilities [54]. These entail the necessity of a nervous and sensory system. The visual system is the most important sensory system in organisms at the highest level of the phylogenetic scale. In particular, the *Visual Cortex* is the most massive system in the human brain.

A novel interdisciplinary domain which aims to emulate some of such capabilities has raised within Computer Science in the last three decades [63]. It comprehends techniques of Image Processing and Analysis, Pattern Recognition, Artificial Intelligence, Computer Graphics, and Robotics, among others. This new domain analyses and evaluates sequences of images concerning human-populated scenes. Impressive developments have also been possible thanks to a large number of technological advances in the hardware field. Emerging capabilities have led to a wide range of scientific contribution, and, subsequently, to new software implementations.

The ultimate aim of this novel domain is to interpret people behaviour. This goal requires detecting and tracking moving objects, and identifying people among them. The analysis of human motion is currently being thoroughly studied, and new domain taxonomies replace the previous ones as the state of the art makes progress. Thus, taxonomies have evolved from simple classifications according to various criteria such as the space dimensionality or the type of sensor used [9, 1, 23, 69] to complex ones based on required system functionalities organised in a hierarchical manner [8, 93, 62]. While in the former surveys reviewed algorithms aimed to estimate the quantitative parameters which describes *when* and *where* motion was detected, in the latters high-level processes are incorporated in order to analyse *which* kind of motion is being performed, and *how* it is carried out.

Thus, in 2000 and according to Nagel [65], an *Image-Sequence Evaluation* (ISE) system would transform image-sequence data into semantic descriptions; subsequently, these descriptions would be processed, and the system would react in terms of signal triggers or conceptual terms. His system is clearly inspired in the ideas of Kanade in the early eighties [49].

In 2004, Gonzàlez [25] proposed the term *Human-Sequence Evaluation* (HSE) to denote the analysis of human motion in order to achieve the understanding of human behaviour, that is, the explanation and reasoning about *why* motion is performed. Further, it would be able to provide *Natural-Language* (NL) scene descriptions, and to generate synthetic views of the environment in order to visualise recognised behaviours and simulate potential situations. Therefore, HSE defines an extensive *Cognitive Vision System* (CVS) which transforms acquired image values into semantic descriptions of human behaviour and synthetic visual representations. Hence, HSE represents a huge challenge in which the aim is to emulate the fascinating performances of a Natural Vision System, and the reasoning and communication skills of a human observer.

In this work, the focus is placed on one of the main HSE tasks: target tracking. Understanding the behaviour of human beings requires the potential targets to be detected and tracked. Tracking can be loosely defined as detecting and keeping lock over time on any object of interest. Consequently, special stress is placed on tracking moving objects in generic human-populated scenes. The problem is tackled without setting any kind of restrictions on the nature of the scene. The proposal should also scale with the number of objects being tracked, which are a-priori unknown.

1.1 Motivation

Robust Multiple-Target Tracking (MTT) in unconstrained dynamic scenes is a complex task, specially when it concerns human-populated environments. Trying to emulate the astonishing performances of such a perfect system as the Natural Vision System represents, without any doubt, a real challenge.

The tracking task is even more complicated when it deals with human beings, thereby making it particularly appealing. In spite of the numerous difficulties involved —or perhaps, because of them— target tracking in human-populated scenes has become a very active research field: it has already generated a vast number of scientific contributions in recent years [62]. However, despite this interest and the substantial developments achieved, this still constitutes an ambitious open problem which is far from being solved.

Further, this interest is also prompted by the increasing number of potential applications within the HSE framework [46, 11, 33, 75, 30, 84, 6, 96, 70]. These include smart video safety and video surveillance, automatic sport-statistics computation, intelligent human-computer interfaces, machine content annotation, or efficient athlete training and orthopedic therapy, among others. Thus, the numerous promising applications constitute an important practical motivation which raises significant funds for HSE research.

1.2 Basic Concepts

In this section, the sense in which basic concepts are used throughout the whole document is introduced.

The main concept —*tracking*— has been above loosely defined as detecting and keeping lock over time on any object of interest; this definition will be subsequently refined, once necessary new concepts have been introduced.

Tracking is performed through a *scene*, which is the piece of the real world that a particular visual sensor can capture. *Mosaics* built from multiple-camera systems or from cameras in motion are also considered as the scene.

Active vision means that it is possible to modify in a controlled way according to what is happening in the scene some camera parameters —such as the zoom, the orientation, the focus, or the diaphragm aperture.

Any entity present within the scene which could be subject to special interest and, consequently subject to be detected and tracked, is called a *target*. Further, any target with intentional capabilities is referred as an *agent*. Depending on the application, the term may include people, manned or unmanned vehicles, or even animals.

The target *state* can be defined as the parametrised knowledge which characterises the target evolution, i.e, all the information required to successfully perform the tracking task. Under certain assumptions, which will be latter stated, the *state* could be defined as the information needed to make the future independent from the past given the present.

The tracking definition can be now detailed by considering tracking as establishing coherent relations among *targets* between frames; or as inferring the target *state* over time using all evidence up to date. *Monitoring* can be broadly defined as observing and keeping record of some processes. Within the Human-Sequence Evaluation context, it refers to a high-level processing of recognised patterns of motion, possibly also including the generation of Natural-Language texts, and the synthesisation of visualisations of the recognised motion patterns within the scene.

The *foreground* is composed of those objects —present in the scene— in which we have a special interest and, consequently, the focus is placed on them. Therefore, it can be seen as conjunction of all targets within the camera field of view. The *background* can be defined as the complement of the foreground concept. Thus, what is considered foreground and background will depend on the current application, and the border between both concepts can be seen as fuzzy.

Agents and mobile objects are usually considered as foreground, whereas fixed objects are commonly referred as background[1]. A mobile object is generally one that

[1] It is worth to notice that any part of the background that moves is considered from then on as foreground —this is the case of a car parked before the application starts that resumed

has been moved or carried by an agent, such a suitcase or a bag[2]. It can have been taken from the scene or placed on it. Nevertheless, it should be noticed that the background may be in motion, such as in the case of waving branches, or flowing water[3]. Although there is certainly motion, these entities must not be detected, since they are not objects of interest. Alternatively, an agent or a mobile object do not have to be in motion as they could have momentarily stopped, or objects could have been left.

The term *scenario* includes all the conditions in which a sequence is acquired. These are related not only to the background characteristics, but also to those derived from the foreground objects that may be within the scene. For instance, it includes issues such as if it considers an *open world*, which means that the number of people and objects within the scene is expected to be variable. On the contrary, a *closed world* refers to a region where all objects within it are assumed to be known at any time. Thus, in an open-world application, people may enter into the scene while others may exit, also removing, leaving, or carrying objects with them.

The scenario certainly includes also the *context*: "any information that can be used to characterise the situation of an entity; an entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves" [17]. Other background conditions must be also taking into account, such as the nature of the illumination and its potential variability. Thus, the scenario can be seen as the *environment* in which the scene is recorded, and therefore it determines the *performance* of the visual system.

The performance can be defined in several terms: the *accuracy,* a measure of how close is the estimated motion to the actual motion performed by the target; the *robustness*, which denotes the system capability of functioning correctly —or at least not failing catastrophically— under a great number of conditions; and other issues related to a particular system implementation such as *real-time* processing — where frames are processed faster than acquired. This is usually determined by other requirements related to cost, energy consumption, and future viability and scalability.

1.3 Potential Applications of HSE

Recent developments in Human-Sequence Evaluation have made possible to consider a huge number of promising applications. Moreover, the benefits that can be obtained from these applications are promoting research in this particular computer-vision area.

Thus, for instance, smart video safety could assist remote elderly care, and the prevention of children drown in unattended swimming-pools; automatic video surveillance increases the security against vandalism, thefts or terrorism; traffic monitoring assists in congestion avoidance; advanced vehicle control systems help preventing

motion, or a moved piece of furniture.

[2] Objects in motion without a known agent mediation —such as a rolling ball or something which falls— are also considered as foreground.

[3] This motion may have some oscillation nature —like leaves moving in the wind— or not —like clouds. In any case, here is assumed that the supposed application do not intend to track leaves or clouds, which would entail considering this entities as foreground.

Domain	**Area**	**Specific application**
Analysis	Diagnosis	Orthopedics
		Athlete training
		Choreography
	Monitoring	People counting
		Video surveillance
		Traffic monitoring
		Video retrieval
	Control and HCI	Domotics
		Driving assistance
		Signaling
Synthesis	Communications and HCI	Teleconferencing
		Virtual reality
		Augmented reality
		Tele-surgery
	Education and Entertainment	Simulators
		Video games
		Animations
	Video compression	Transmissions
		Storage

Table 1.1: HSE applications.

collisions and off-lane accidents; intelligent gestural user-computer interfaces provide driving assistance, and allow domotics applications; orthopedic therapy, athlete training or computer animation benefit from an accurate motion analysis; sports can also profit from on-line computed statistics; automatic content annotation based on motion semantics brings new information search capabilities.

These applications can be classified according to their aims, see Table 1.1. A division between analysis and synthesis applications is here considered. The former attempts to process an input video signal, whereas the goal of the latter is to generate synthetic scenes, agents, and their motion. Notwithstanding, complex applications may comprehend several of the following categories.

The system requirements could be rather different depending on the desired application, as well as the considered assumptions and system capabilities.

1.3.1 Analysis applications

This area covers three kinds of applications, depending on the nature of the data to be extracted from the image sequence, and how these data is going to be subsequently analysed. Thus, diagnosis, monitoring and control applications are intended:

1. **Diagnosis**: in which the aim is to evaluate the subject performance. Potential

applications include the fields of orthopedics, athlete training, or choreography enhancement. Strong accuracy requirements should be expected. However, some assumptions can be made, such as expecting only one person with special clothes in a controlled environment. In this case, the system capabilities must include target tracking and body-pose recovery.

2. **Monitoring**: in which the goal is to detect and track people within the scene, perhaps identify them, or recognise particular actions. We can consider within this area people counting, video surveillance, traffic monitoring, or video-retrieval tasks. In surveillance applications real-time requirements are usually necessary. These applications also need extreme robust performances. Thus, the considered assumptions should be minimum. The system must be able to deal with open worlds, illumination changes, background in motion, etc. On the other hand, accuracy requirements can be relaxed. The expected capabilities could include presence detection, target tracking, and people identification.

3. **Control / Human-Computer Interactions (HCI):** in which the data extracted from the sequence is going to be used to provide command functionalities. Potential applications could include gestures, facial-expression and body-pose interfaces. These interfaces would be used in domotics, driving assistance, or signaling in noisy environments. In this area real-time processing will be the higher requirement although, depending on the applications, both accuracy and robustness could also be needed. In this case, system capabilities such as action recognition and gesture interpretation are essential.

1.3.2 Synthesis applications

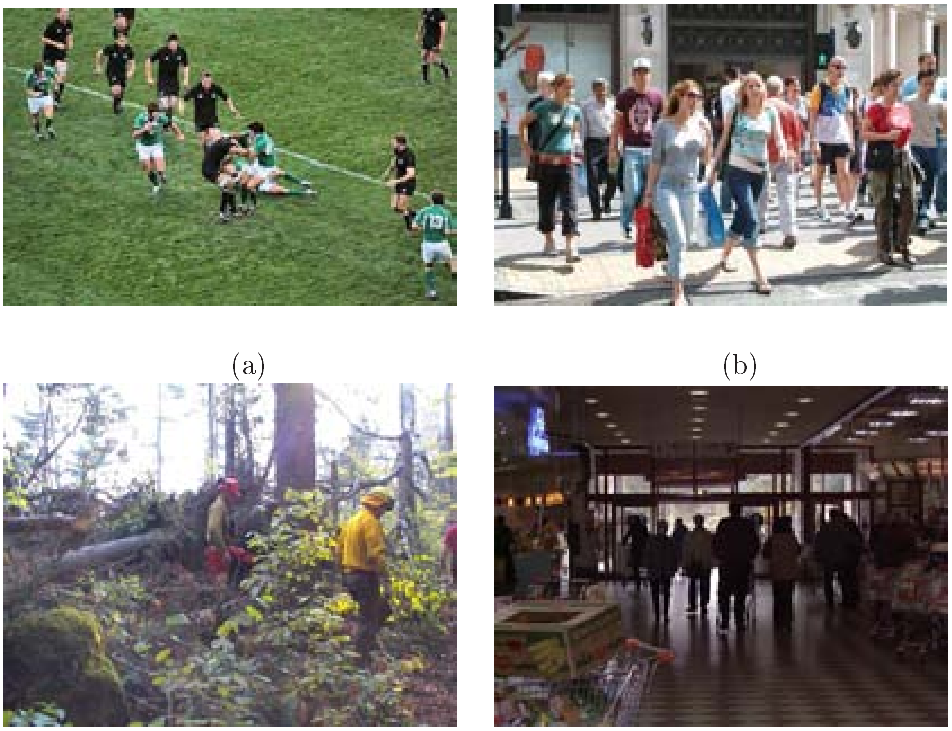
Synthesis applications are concerned with generating new sequences. This can be done from real images, computer drawings, natural language sentences, or from a mixture of these[4]. Thus, the following application categories are considered:

1. **Communications / HCI:** in which the goal of the generated sequence is to provide some information to a user who may or may not be at the same place where the original sequence was taken. This area includes teleconferencing and virtual and augmented reality. Real-time processing is an essential fact in this kind of application in order not to introduce delays in the communication. Some applications such as tele-surgery would also need special accuracy requirements.

2. **Entertainment / education:** in which the sequences serve leisure purposes, such as animations or video-games, or education and training, such as simulators.

3. **Video compression:** in which the goal is to build a new sequence from another one by minimising the required storage space or the bit transmission rate.

[4] This kind of application usually follow the *synthesis from analysis* paradigm. Therefore, a previous motion detection and modelling stage from real data is frequently mandatory.



(a) (b)

(c) (d)

Figure 1.1: Example of some MTT inherent difficulties, as mentioned in the text. (a) Highly non-linear dynamics and non-rigid shape. (b) High appearance variability. (c) Cluttered scenario. (d) Illumination-related difficulties.

1.4 Problem Overview: Which are the Difficulties?

Multiple-Target Tracking is extremely complex and time-consuming. Further, strong requirements may be mandatory, like extreme robust performances, high accuracy, or real-time processing.

This task being so ambitious, serious difficulties should be expected. First of all, adversities common to other Computer Vision areas could cause system failures, such as uncontrolled changing illumination, shadows, cluttered backgrounds —also possibly in motion— target variability, etc.

In addition, MTT entails numerous special difficulties:

1. it involves dealing with remarkably non-rigid targets; they are not only highly articulated, but also elastically deformable, and usually wear loose-fitting clothes;

2. neither their appearance, nor their shape can be specified in advance; there is a considerable target diversity, not only due to the presence of several classes of targets —like people, vehicles, animals, or any object— but also within a

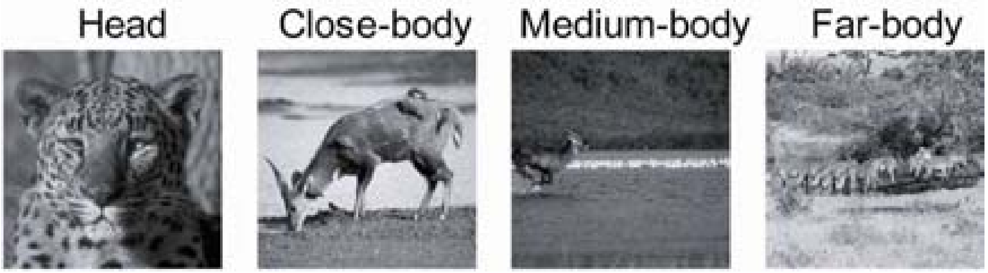


Figure 1.2: Example of resolution selection. (Figure taken from the Scene Understanding Symposium notes, Poggio, 2007).

particular class, and even with the same target at different times; the fact that both shape and appearance vary as the agent performs a particular motion must as well be taken into account;

3. their dynamics are highly non-linear, a-priori unknown, and they are always subject to sudden and unforeseeable changes; in this case, the agent's intentionality plays an important role;
4. in open-world applications, the number of agents within the scene may vary over time; they might also carry, leave or remove objects from the scene, thereby actively modifying the background;
5. in unconstrained and dynamic environments, the illumination and background-clutter distracters are uncontrolled, affecting the observed appearance as time goes by; this depends on issues such as the targets' position in the local background, or their orientation to different —and maybe time-varying— illumination sources;
6. finally, agents tend to interact among themselves, grouping and splitting, causing partial or complete occlusions, and thereby changing their observed appearance and shape at any time.

Summarising, both background and target appearances are extremely difficult to model, and they vary over time in an uncontrolled way. Further, target movements and interactions are considerably hard to predict. Therefore, there is still much ground to cover before reaching a point where it can be said that the unconstrained people-tracking problem has been solved, what makes the task specially appealing. Some examples of images with the above mentioned difficulties are shown in Fig. 1.1.

1.5 Assumptions over Considered Scenarios for HSE

One should also keep in mind that several scenarios may be considered depending on the desired application and where it is carried out. Different scenarios may imply rather different approaches. The following criteria can be used to distinguish among the different scenarios in order to decide the most suitable approach:

- Time-scale selection, in which the change ratio for the different features is set. Attributes can change abruptly —like the motion pattern, or a goal-directed behaviour— slowly —appearance— or be quasi-permanent —face shape, gait.

- Spatial-scale selection, in which the resolution is chosen. High resolution would be necessary to analyse gestures or facial expressions, whereas pose analysis requires a medium resolution and trajectory tracking the lowest one, see Fig. 1.2. Minimum resolution is set depending not only on the current application but also on the chosen approach.

- Information-channel selection, in which it is decided whether facial, hand, the whole body information, or several of these are used. This would depend on what the focus is placed on: expressions, gestures, pose, location, presence, etc. More than one channel could be considered in order to improve the system robustness by using redundancy to disambiguate unsettled situations.

- Application requirements, in which accuracy, speed and robustness are taken into account depending on the application purpose. For example, some applications will require real-time processing, while in others off-line processing will be enough.

- Model necessity and availability, in which the possibility of considering for instance an articulated —and perhaps even elastic— body-structure, or a simpler body model, is evaluated. Models for other information channels, such as faces, or for the scene can be also taken into account.

- Active or passive devices may be taken into account. The formers relies on radiating some signal from a transmitter attached to the subject; whereas the latters use natural signal sources such as light. In this second case, markers can be used. However, both methods can be considered as intrusive[5]. Obtaining robust and accurate tracking performances, whilst using non-intrusive technology, is frequently mandatory. This is what had led to vision-based systems. Therefore, a technology which does not depend on devices attached to cooperative subjects is desired, that is, a *Markerless Motion Capture*.

- Context, in which several premises are often assumed:

  - Camera assumptions relative to single or multiple cameras, fixed or mobile, monocular or stereo, monochrome, colour or infrared, active vision, or the use of particular camera models, optics, etc.
  - Background assumptions relative to whether outdoor or indoor scenes are considered, static or in-motion background, potential illumination changes, shadows, presence of clutter, a-priori known objects in the scene, or even a detailed scene model.
  - Foreground assumptions. Two kind of premises can here be considered. In the first place, those ones related to movement, such as possibility of

[5] An exception to this generalisation is given by thermal imagery, where the signal is radiated by the targets themselves and no marker is required.

> occlusions; agents and objects entries and exits from the scene; smooth, restricted or already-known dynamics; whether the camera is faced or not, or whether attitudes and intentions are known. Secondly, those related to the structure, such as whether the subject and the start pose is known or not; whether a single person, multiple or groups of people can be found in the scene; presence of special clothes or markers, or whether objects can be carried.

Some of these criteria have been also used by Pentland in his *Looking at people* domain taxonomy [69], by Moeslund et al. while considering *Motion Capture Assumptions* and *Application Performances* [63], or by Gavrila while describing possible taxonomies according to sensor properties [23].

The aim of this work is to develop a general approach able to cope with unconstrained tracking in trajectory-analysis HSE applications. Thus, among the above-presented common premises, only the following ones are assumed (which usually hold in most of this kind of applications):

1. All sources of noise are considered to be uncorrelated, and to cause White Additive Gaussian Noise (WAGN).
2. The background slowly changes with respect to the motion of the targets within the scene.
3. Changes in the target's dynamics and appearances are smooth at the current frame rate. This assumptions allow us to introduce the following simplifications in the dynamic, appearance and shape models:
   (a) Since their long-run dynamics are hardly predictable, a first-order dynamic model is adopted. Thus, the considered dynamic models are given by a constant-speed approaches where the acceleration is modelled as WAGN. The latter is supposed to be constant during the sampling period, and independent between periods.
   (b) Appearances are supposed to evolve smoothly in short-time scales. This allows to set a time continuity, and to avoid appearance updatings under certain conditions. A robust appearance model can be built, allowing target matching among frames which are close enough.
   (c) Target interactions cannot abruptly change between frames. Thus, for instance, targets cannot change from grouped to single without ever being splitting.
4. The size of the targets in the image is assumed to be big enough in order to build a representative statistical appearance model, but small enough w.r.t the scene size.
5. Humans will essentially remain in upright posture. This along with the chosen resolution permits to select a coarse blob representation as information channel. No body model is used. Blob orientation is considered to undergo just minor variations.

6. The sequence of input images come from a stationary single monocular colour camera.

Therefore, no assumption is taken relative to the following issues:

1. The number of targets within the scene, which may vary as time evolves.
2. Their trajectories and dynamics, which are completely unknown beforehand, and presumably highly non-linear.
3. The scene conditions, which could be uncontrolled. No knowledge is a-priori available about illumination conditions, complex clutter distracters, or regions in motion. These may also evolve over time depending on the lighting, weather, moved objects, etc.
4. The target appearances and shapes, which are unknown. No markers are placed on the targets. Heavy appearance and shape changes can be expected due to the deformable and articulate nature of the targets, and potential variable illumination conditions.

1.6 The Ultimate Goal

In this work, the aim is to achieve a robust and accurate MTT. This implies the inference of the state of each target within the scene. Therefore, tracking is the result of the conjunction of detection, estimation and adaptation tasks. Firstly, targets need to be detected within the scene. This allows the system to initialise a tracker over each target. Then, coherent relations must be established between detected targets over time by means of prediction and validation in accordance with new measurements. Thus, estimation reduces the search area and may cope with expected difficulties such as occlusions. In addition, different hypotheses can be considered simultaneously, improving the system performance in terms of robustness. Finally, the models themselves must be adaptive in order to handle unforeseeable alterations.

Therefore, the ultimate goal is to conceive a principled image-based tracking architecture which makes a step forward in dealing with the aforementioned difficulties. This system will be implemented and experimentally verified using real image sequences. It should be able to simultaneously perform a reliable tracking of multiple targets in unconstrained and dynamic open-world scenarios, in the above stated conditions. At the present stage, this will be done using as system input the output of a single, monocular, static colour camera.

As a result, target trajectories will be obtained, as well as quantitative information about the target state at any time, such as their speed and size, and qualitative one, such as whether they are being occluded, grouping or splitting, and entering or exiting the scene. Target trajectories, and the interactions among them will be analysed from a coarse representation without making use of a-priori models. Consequently, in this work no attention is placed on target postures and actions, or facial expressions and emotions. These are in a level of detail which is out of the scope of this document[6].

[6] Nevertheless, these topics are within the scope of the HSE framework that steers the

1.7 Approaches and Contributions

In this thesis, two different tracking approaches are presented and confronted. A probabilistic framework is commonly used as a way to perform this task [80]. Classical approaches, such as the *Kalman Filter* [48], rely on strong assumptions about the linearity and Gaussianity of the involved distributions, which cannot be applied in complex scenes. The first approach developed in this document focuses on tracking by means of *Particle Filtering* (PF) in the conditions described above. This approach has been widely explored by several previous algorithms [38, 91, 67, 16]. Although some results have been achieved, many undesirable effects still remain. These misbehaviours are here highlighted, and an algorithm which deals with some of them is proposed.

Thus, tracking is first performed by enhancing the particle filtering framework. The main contributions of the presented approach are the following:

- Previous state estimations are used in the dynamics updating process to feed back the sample state. All state variables are regularised. Both actions attempt to reduce the number of samples required to carry out the tracking task while attenuating the trajectory jitter.

- Target appearance is modelled by means of grey-scale templates. The effects of the position and size errors on the likelihood function are explored. Undesirable effects are tackled by making use of an appropriate likelihood mapping. Subsequently, in order to cope with clutter distracters, colour-based histograms are used instead to model target appearances. Likelihoods are computed from histogram similarity. Colour information relative to the target surroundings is used to tune the colour histograms.

- One of main particle-filter drawbacks is *sampling impoverishment*. This problem becomes critical in a multiple-target tracking scenario. By modifying the sample-weight normalisation —taking into account the number of detected targets— the loss of any of the targets due to the lack of samples is avoided.

- Model drift is precluded by careful updating, based on likelihood measures.

- Occlusions are handled considering the predicted trajectories of all targets within the scene and the history of likelihood measurements. Thus, target tracking and updating is faced according to their occlusion status. Likelihood measures are taken to infer when the appearance model can be reliably updated.

Despite the great efforts spent on this approach, it still lacks from a robust and accurate enough performance due to the important drawbacks of a PF framework, together with the inherent complexity involved in non-supervised multiple-human tracking. As a second approach, a principled framework is here proposed to accomplish this task. The main features of this approach are the following:

efforts of our lab. Cooperation with HSE Cognitive Levels, and body and face information channels is intended. See `http://iselab.cvc.uab.es/` for further details on these issues, and on the *Image Sequence Evaluation* research lab.

- It consists of a modular and hierarchically-organised architecture. It aims to deal with such a complex task by taking advantage of a general and structured framework. A set of co-operating modules, distributed in three levels, work following both bottom-up and top-down paradigms, thereby maximising potential synergies.

- The approach follows the natural paradigm, where visual-stimuli analysis is performed by the combination of pre-attentive and attentive processes. Further, it makes use of first-order and second-order motion perception.

- Levels are defined according to the different tasks to be performed, namely target detection, low-level and high-level tracking. Thus, a remarkable characteristic of this architecture is that the tracking task is split into two levels: a lower level based on a short-term blob tracker, and a long-term high-level target tracker. While the former permits tracking without the need of detailed knowledge, the latter automatically builds and tunes multiple appearance models, manages the events in which the target is involved, and selects the most appropriate tracking approach according to these.

- Every module has a specific functionality which is performed by a particular algorithm. However, being the architecture modular, these are subject to be substituted by any enhanced method developed in the future. New functionalities can also be easily added. Further, stress has been laid on designing robust high-level tracking algorithms to tackle such a complicated task.

- Two operation modes are implemented, namely Motion-based Tracking (MBT) and Appearance-based Tracking (ABT). These are independent and automatically selected according to each target particular conditions.

- A complex event management is performed. Multiple-target interaction events, and a proper scheme for tracker instantiation and removal according to scene events, are considered. This allows the system to switch between the two considered operation modes. Further, open-world applications can be tackled.

- The current proposal is fully automated, and thereby no human interaction is required. Further, no a-priori knowledge about either the scene or the targets, based on extensive off-line training or learning periods, is used. However, the expected future use of this high-level information can do nothing but enhance the current system performances. Hence, in the present approach the scenario could be completely unknown beforehand, and no a-priori knowledge is available about potential targets. The method is auto-adaptive in issues such as the scene model, the number of targets being tracked, or their most convenient appearance representation.

- The proposed system deals with multiple targets simultaneously. It is scalable with the number of targets, avoiding the curse of dimensionality present in most other systems.

- It copes with clutter distracters by selecting the most convenient colour-related features. A set of appearance models is continuously conformed, smoothed and updated. Thus, multiple targets are represented using several models for each of them, while they are simultaneously being tracked. Further, colour information relative to the target surroundings such as the background and other close targets is used to tune the appearance models.

- Model drift is precluded by a careful updating of high-level appearance colour models, thereby ensuring proper tracking despite noisy measures, estimate errors, partial or complete occlusions, and changes in the illuminant and camera viewpoint.

Summarising, the aim of the proposed system is to work as a stand-alone application in a non-friendly, complex and dynamic open-world scenario, which could be completely unknown beforehand. Thus, possible scenarios could include an indeterminate number of non-white light sources, heavy background clutter, huge target variability, and complex target interactions.

1.8 Document Outline

The remainder of this document is organised as follows. Chapter 2 covers the state of the art: some previous surveys and taxonomies related to the analysis of human motion are here described; subsequently, the most recent and relevant approaches which tackle target detection and tracking are reviewed. The advantages of the different methods are explained and their drawbacks exposed.

In Chapter 3, the HSE framework —in which the tracking proposal is inscribed— is depicted, evolved, and confronted to previous taxonomies. In this research, we aim to develop a system that can be seen as a part of a more complex one which performs a HSE, which is also the aim of the EU HERMES project[7] in which the author was actively involved at the time this thesis was developed.

Chapter 4 develops the first approach presented in this document. The necessary probabilistic framework to accomplish this work is described. Bayesian filters are explained. Particle filters are revisited and their misbehaviours exposed. A particle filter algorithm for multiple-target tracking is implemented and tested. Subsequently, it is enhanced by incorporating colour-based appearance models and likelihood functions.

The second proposal is described in chapter 5. First, the tracking architecture is outlined in section 5.1, and *justified* by pointing out the similarities with a Natural Vision System in section 5.2. Then it is fully described in the next three sections: section 5.3 details how the segmentation is carried out, and the chosen data representation for the detected foreground blobs; section 5.4 discusses the low-level tracking tier; and section 5.5 presents a high-level appearance tracker, with on-line feature selection, operation mode switching, and complex event management.

Chapter 6 is related to experimental results. First, some considerations on tracking performances are given in section 6.1. Then, the next sections shows an extensive set of experimental results of both approaches using own and public well-known

[7] EC grant IST-027110, see http://www.hermes-project.eu/

databases. Finally, chapter 7 presents a brief summary of the contributions, summarises the conclusions, and proposes some future-work lines which can extend the system capabilities.

1.9 Resum

Els éssers humans, així com una gran diversitat d'espècies animals, han desenvolupat una capacitat sorprenent de processament complex de estímuls visuals, variables i continus. Milions d'anys d'evolució han desenvolupat sistemes visuals altament eficaços que mostren, d'una manera aparentment fàcil, rendiments increïbles.

L'habilitat de la detecció de moviment s'ha d'esmentar indubtablement entre les facultats més potents dels sistemes de Visió Natural [68]. Aquesta habilitat està implicada crucialment en nombrosos assumptes crítics per a la supervivència de l'espècie, com el seguiment objectius que es mouen, malgrat oclusions parcials i canvis d'il·luminació dràstics; la extracció de l'estructura de profunditat del món aprofitant la paral·laxi del moviment; la detecció objectes que es camuflen en un fons de color i textura similars; o en el reconeixement d'objectes des del moviment relatiu de les seves parts del cos. A més, aquesta facultat s'implica en unes quantes respostes motors associades, com l'estabilització de la mirada, o el control de les extremitats.

Així, l'habilitat de percebre el moviment de predadors potencials i de preses ha estat inevitablement connectat amb les capacitats d'automoviment [54]. Aquestes suposen la necessitat d'un sistema nerviós i sensorial. El sistema visual és el sistema sensorial més important en organismes del nivell més alt de l'escala phylogenetic. En particular, el *Còrtex Visual* és el sistema més massiu al cervell humà.

Un domini nou interdisciplinari que tracta d'emular algunes d'aquestes capacitats s'ha alçat dins de les Ciències de la Computació en les últimes tres dècades [63]. Comprèn tècniques de Processament i Anàlisi d'Imatges, Reconeixement de Formes, Intel·ligència Artificial, Gràfics per Computador, i Robòtica, entre altres. Aquest domini nou analitza i avalua seqüències d'imatges d'escenes poblades amb humans. Els desenvolupaments impressionants també han estat possibles gràcies a un número gran d'avenços tecnològics en el camp del hardware. Les capacitats emergents han conduït a una àmplia gamma de contribucions científiques, i, posteriorment, a aplicacions noves de software.

El propòsit últim d'aquest camp nou és interpretar el comportament de la gent. Aquest objectiu exigeix detectar i seguir objectes que es mouen, i identificar gent entre ells. L'anàlisi del moviment humà s'està estudiant actualment minuciosament, i les taxonomies noves reemplacen a les antigues mentre l'estat de l'art fa progrés. Així, les taxonomies s'han convertit des de classificacions simples, segons criteris diversos com ara la dimensionalitat espacial o el tipus del sensor utilitzat [9, 1, 23, 69], a més complexos basats en funcionalitats de sistema organitzats jeràrquicament [8, 93, 62]. Mentre en treballs antics els algoritmes aspiraven calcular els paràmetres quantitatius que descriuen *quan* i *on* era detectat el moviment, en els més actuals s'incorporen processos d'alt nivell per analitzar *quina* classe de moviment està sent observat, i *com* s'està executant.

Així, al 2000 i segons Nagel [65], un sistema per l'*Avaluació de Seqüències d'Imat-*

ges (ASI) es va definir per transformar dades de seqüències d'imatges en descripcions semàntiques; posteriorment, aquestes descripcions es processen, i el sistema reacciona en termes de senyal o conceptes.

El 2004, Gonzàlez [25] proposà el terme *Avaluació de Seqüències amb Humans* (ASH) per definir l'anàlisi de moviment humana per aconseguir la comprensió del comportament humà observat, és a dir, l'explicació i raonament sobre el *per què* el moviment es observat. A més, un sistema AHS proporciona descripcions de l'escena en llenguatge natural, així com la generació de seqüències sintètiques de l'entorn per visualitzar comportaments reconeguts i simular situacions crítiques difícilment observables en el món real. Per això, ASH es pot definir com un Sistema de Visió Cognitiva, on hi han transformacions de valors d'imatge a descripcions semàntiques sobre el comportament humà, així com representacions visuals sintètiques. Per això, AHS constitueix un desafiament enorme en el qual el propòsit és emular els rendiments fascinants d'un Sistema de Visió Natural més les habilitats de raonament i comunicació d'observadors humans.

En aquest treball, el focus es posa en una de les tasques de ASH principals: el seguiment. Entenent el comportament d'éssers humans exigeix que aquests es detectin i se segueixin. El seguiment pot ser definit com la detecció i el seu posterior manteniment en qualsevol objecte d'interès. Conseqüentment, l'iterés es posa en objectes movent-se en escenes genèriques poblades amb humans. El problema es tracta sense posar cap classe de restriccions sobre la natura de l'escena. La proposta també hauria de ser independent del número d'objectes que se segueixen, que a-priori és desconegut.

Chapter 2

Related Work

In spite of being a relatively new research area, a massive number of contributions related to HSE have been published in the last years[63, 62]. Undoubtedly, it represents an ambitious challenge, which is further raising important amounts of private and public funds due to the increasing number of attractive commercial applications.

The growing number of contributions in recent years has motivated the publication of multiple surveys [1, 23, 93, 62]. These review the state of the art, while proposing new domain taxonomies. Nevertheless, this field still lacks from a widely accepted taxonomy which arrange in a systematic way the different works. Thus, it would be interesting to show the relations between these, while including a hierarchical classification.

In this chapter, the most relevant surveys are revisited, thereby putting into context the work here proposed. Further, a new taxonomy is also proposed. Subsequently, the focus is placed on detection and tracking methods. Thus, some of the most significant algorithms are discussed. The advantages of the different methods are explained and their drawbacks exposed.

2.1 A Review of Most Relevant Surveys and Taxonomies on HSE

The increasing number of papers —first related to people detection and tracking, then also to the analysis and understanding of human motion— in the last years has led to the publication of several surveys. Each of them has presented a taxonomy which arrange the most significant previous works according to different criteria.

Aggarwal and Cai presented a series of reviews in different workshops. Finally, this work resulted in what is probably the first relevant survey [1]. It reviews proposed approaches from 1980 to 1998, and 51 papers are referenced. Their taxonomy considers three main areas: (i) *body structure analysis*, (ii) *tracking moving humans*, and (iii) *recognition*, see Fig. 2.1.

The first area concerns the structure of human-body parts. It is subdivided in two kind of approaches, depending on whether they rely on a-priori human shape

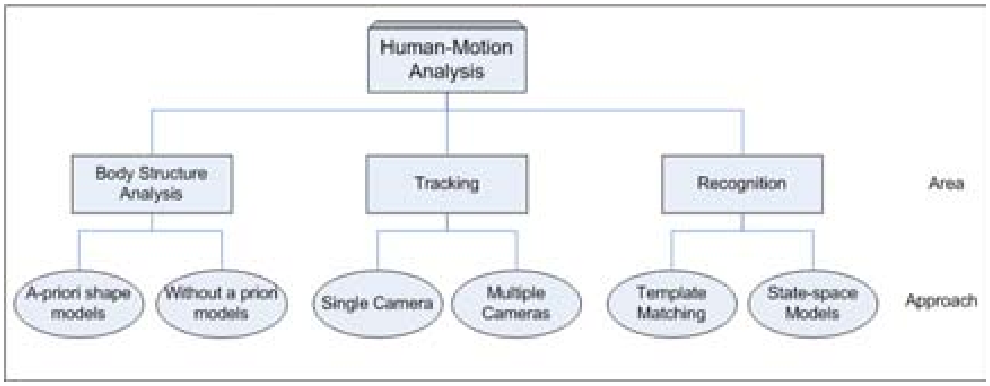


Figure 2.1: Taxonomy presented by Aggarwal and Cai in [1].

models or not. Approaches from both categories can be grouped according to the representation used, namely, stick figures —the supporting bones— 2-D contours —the projection of the human figure— or volumetric models —modelling the flesh.

The second proposed area involves human tracking without considering its articulated configuration. Another subdivision is made based on whether a single camera or multiple perspectives are used. Papers from both approaches are also grouped depending on the representation, namely, points, 2-D blobs —that is, regions with similar properties— or 3-D volumes. The considered features are related to motion information (position, velocity), intensity values, etc.

The final area addresses human-activity recognition. Papers are grouped depending on whether they use *template-matching techniques* or *state-space models*. The former uses representations based on points, lines and blobs, while the latter uses point and meshes.

Another survey covering the time period from 1973 to 1997 —which references 81 papers— was presented by Gavrila [23]. Here, the classification is based on two criteria: the type of model, and the space dimensionality. Thus, this survey distinguishes three categories: (i) *2-D approaches without an explicit shape model*, (ii) *2-D approaches with explicit shape models*, and (iii) *3-D approaches*, see Fig. 2.2.

The first kind of approach relies on statistical descriptions based on low-level features and heuristics such as image moments, orientation histograms, and skin colour. The second one assumes a known point of view and a defined motion model. Representations are based on sticks and 2-D blobs. The third kind of approaches are mainly based on stick figures which model the skeleton, and 2-D surfaces or volumes which model the flesh. Features such as joint angles are considered. The three categories aim to provide results for all the required functionalities at the moment, that is, detection, tracking and recognition.

In addition, Gavrila provided an application classification altogether with the system required capabilities. Six fields are considered: virtual reality, smart surveillance, advanced user interfaces, motion analysis, and model-based coding. Among the capabilities, presence detection, identification, tracking, action recognition, and gesture

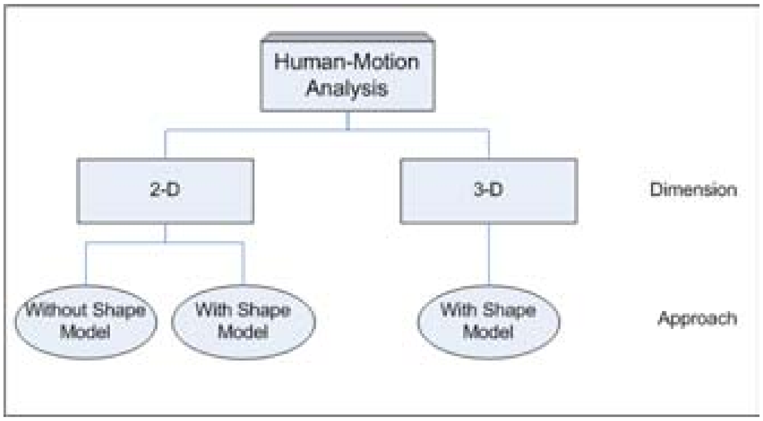


Figure 2.2: Taxonomy presented by Gavrila in [23].

or expression recognition can be found.

Moeslund and Granum [63] gave the most comprehensive survey, covering the years between 1980 and 2000 and citing 154 papers. Further, some previous surveys are discussed and compared. The covered period is later extended in [62], where contributions from 2000 to 2006 are included, and 337 papers are referenced.

In their work, a novel taxonomy based on functionalities is proposed: (i) *initialisation*, (ii) *tracking*, (iii) *pose estimation* and (iv) *recognition*, see Fig. 2.3. However, facial expression and hand gestures are not covered.

The first considered task concerns the camera, scene and target model initialisation, that is to say, calibration, manual or automatic parameter tuning, target initial pose, etc.

Then, tracking is addressed. The process is divided in three main tasks, i.e., target *segmentation*, *representation* and *tracking*. The former is divided in *temporal* and *spatial* approaches. According to the authors, on the one hand, temporal approaches can be subdivided into subtraction —which includes frame differencing and background subtraction— and optical flow techniques. On the other hand, spatial approaches may rely on thresholding, or on statistical methods.

Secondly, the representation of segmented entities is reviewed. Two categories are given, namely, *object-based* —points, boxes, silhouettes or active contours, and blobs— and *image-based* —spatial, spatio-temporal, edges, and features such as length, area, etc. Finally, the tracking task is discussed considering *model-based approaches* opposed to *probabilistic learnt models*; and single camera against multiple-camera approaches.

The third main functionality concerns the pose estimation. It is here considered as either a tracking post-processing, or as an active part of it. Three categories are then given: *model-free*, *indirect model* and *direct model*. The former builds a representation without the use of an a-priori model. It can be based on a point, box or stick representation. The second category considers approaches which use a model as a guide to interpret the given data. The latter includes those approaches which

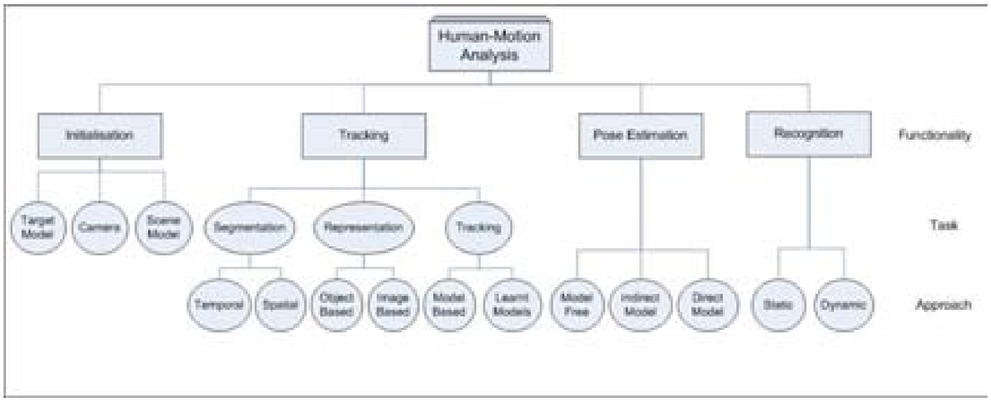


Figure 2.3: Taxonomy presented by Moeslund and Granum in [63].

use a direct model, that is, a detailed a-priori human model.

This last category is discussed in a comprehensive way. A large number of papers are classified according to their *abstraction level* —edges, silhouettes, sticks and joints, blobs, depth, texture, movement— the *dimension* —2-D, $2\frac{1}{2}-\text{D}$, 3-D— or the *model type* —cylinders, stick figures, patches, cones, ellipsoids, scaled prisms, CAD model, boxes, etc.

The way in which the results are evaluated is also taken into account: quantitative such as ground truth or manually segmented data, and qualitative such as visual inspection or animation.

Subsequently, the recognition task is addressed. Two distinction are made: *static* and *dynamic* recognition. Among the former, techniques such as template matching, normalised silhouettes or postures can be found in the literature. The latter includes low-level methods, such as spatio-temporal templates or motion templates, and high level ones such as Hidden Markov Models (HMM) or Neural Networks (NN).

Finally, a classification of applications is also proposed by considering three main areas: *surveillance*, *control* and *analysis*. A taxonomy relative to the assumptions made in the field is as well given, which consists of movement, environment and subject assumptions.

In 2003, Wang et al. presented an extensive and one of the most interesting surveys [93]. The time period from 1992 to 2001 is covered by citing 164 papers. Applications are classified under three categories, namely, *visual surveillance*, *advanced user interfaces,* and *motion-based diagnosis and identification.* Previous surveys are also revisited. This review presented a taxonomy based on functionalities organised in a hierarchical manner. The proposed framework consist of three levels corresponding to low-level vision, intermediate-level vision and high-level vision. Each level is focused on one of the following task: *detection*, *tracking* and *behaviour understanding*, see Fig. 2.4.

The detection level aims to segment and group moving pixels corresponding to people. It is divided in two sub-processes: (i) *motion segmentation* and (ii) *object classification.* The former includes several approaches which are organised under four

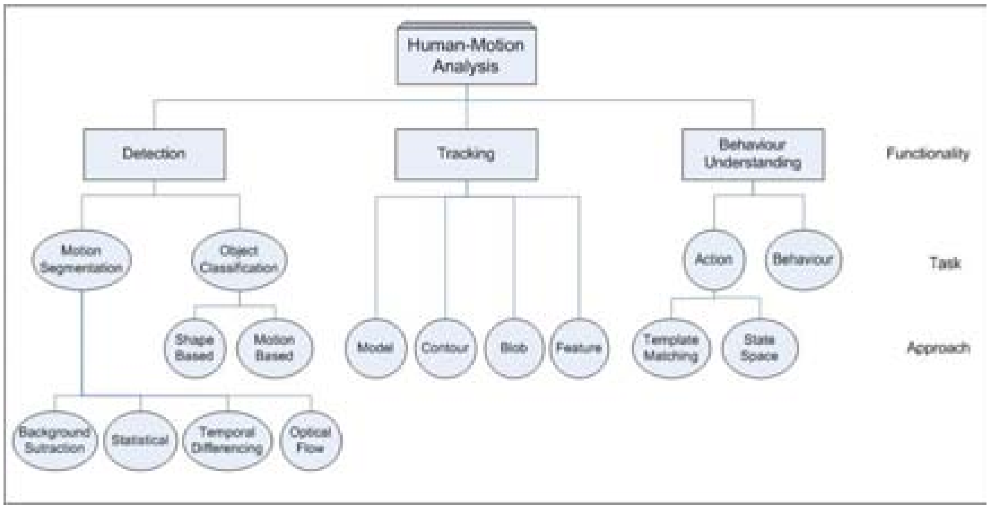


Figure 2.4: Taxonomy presented by Wang et al. in [93].

categories, namely, *background subtraction*, *statistical methods*, *temporal differencing* and *optical flow*. The latter is subdivided into two categories, which are *shape-based classification* and *motion-based classification*.

The goal of the tracking level is to establish coherent relations of image features between frames. Present-day approaches are classified according to whether they are *model-based*, *contour-based*, *region-based* or *feature-based*. With respect to the former, human-body models can be represented by stick figures, 2-D contours or volumetric models. The second and third kind of approaches aim to track detected contours and blobs, respectively. Finally, the last one aims to track sub-features as points or lines.

The highest level involves action recognition and description, and the analysis and understanding of human behaviours. The usual techniques are dynamic time warping, hidden Markov models or neural networks. The recognition is carried out under two groups of approaches, namely, *template matching* and *state-space methods*. Semantic descriptions are also receiving increasing attention from the community, as is stated by the authors.

Finally, Pentland [69] presented a paper which, without aiming to classify explicitly the up-to-time approaches, touches a diversity of human-motion analysis methods and applications. This domain was called in the paper "Looking at People", and this term have been subsequently widely used[1]. A review of related mathematical techniques, and a domain taxonomy based in channels, scales and intentionality is provided. The state-of art of face recognition, surveillance, 3-D methods and perceptual user interfaces is revisited.

As has been aforementioned, this thesis is focused on target detection and tracking. Further, a taxonomy of these two functionalities based on the information flow,

[1] As an example, the search of the terms "looking at people" plus "tracking" through the Internet yields more than 24000 hits.

and a structured framework which encloses Human-Motion Analysis functionalities are presented in section 2.2, and chapter 3, respectively.

In order to put the presented work into context, it is worth to locate it within the taxonomies above revisited. Thus, it lies within the *tracking* area, and the *single-camera* approach category of the taxonomy proposed by Aggarwal and Cai [1]; within the *2D* area, and *without-shape-model* approach category of the one proposed by Gavrila in [23]; in the taxonomy proposed by Moeslund and Granum in [63], it lies within the *tracking* functionality, covering all *segmentation*, *representation*, and *tracking* tasks, and following *temporal* segmentation approaches, *object-based* representation, and *probabilistic learnt models*; finally, it the taxonomy presented by Wang et al. in [93], our work is covers both *detection* and *tracking* functionalities, and it addresses *motion-segmentation* and *tracking* tasks by following *statistical* approaches for the former, and *blob* ones for the latter.

2.2 State of the Art of Target Detection and Tracking

In this section, a review of the most relevant papers published in recent times relative to segmentation, detection and tracking approaches is presented. The different proposals are here outlined, and their advantages and drawbacks discussed. However, despite the huge efforts made, and the fact that achieving robust and accurate tracking is the first basic task to HSE, the problem is still open.

From the author point of view, target segmentation and tracking tasks are so linked that they should be considered together. Thus, a proper segmentation is, at least, essential for tracking initialisation and error recovery. And without applying a tracking scheme, it is not possible to keep a temporal consistency on detected targets. Further, it is really unusual to find a relevant paper specific to just segmentation or tracking. Papers are here inscribed in one of the following categories or another according to their main contribution, albeit they usually cover several tasks

This review implicitly presents a taxonomy according to the information flow. Thus, tracking is usually carried out using either bottom-up or top-down approaches. The formers rely on foreground segmentation, and a subsequent target association, which is usually followed by a state filtering; on the contrary, the latters are based on a prior complex motion, shape and/or appearance modelling, and a posterior state prediction. Thus, bottom-up approaches generate hypothesis according to the results of image processing, whereas top-down ones specify a-priori generated hypotheses according to current image data.

In this taxonomy, each of the bottom-up tasks is subsequently divided according to the different techniques used —which in some cases coincide with the ones stated by the aforementioned surveys.

Top-down approaches are split taking into account the tracking technique used, although it is subsequently detailed the feature in which the particular proposal rely. A sketch of this taxonomy is shown in Fig. 2.5.

Finally, some research groups have developed structured architectures which aim not to be restricted to a particular task, but to perform a global scene analysis [46, 81].

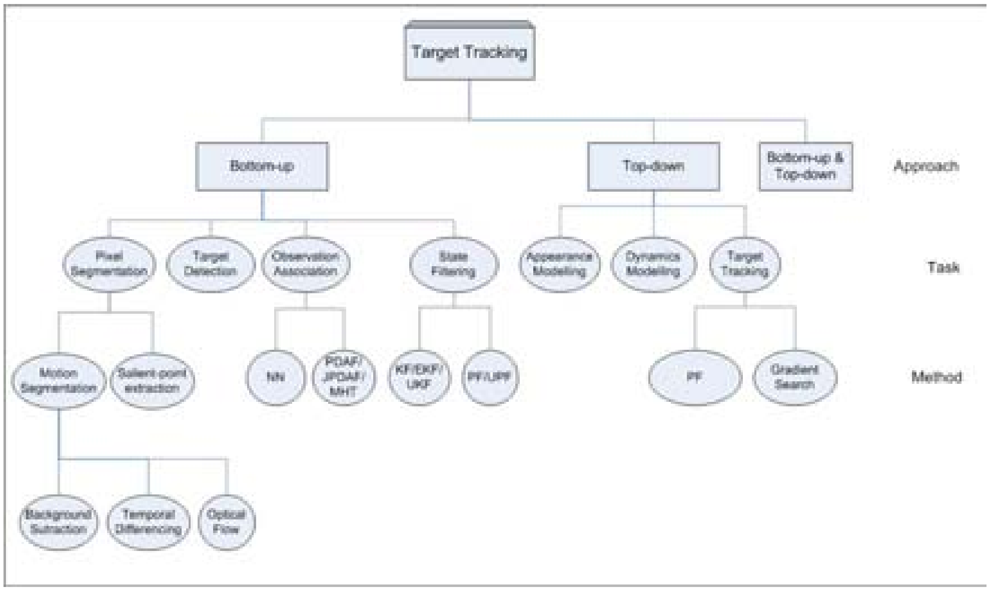


Figure 2.5: Proposed tracking taxonomy. Tracking approaches are classified in bottom-up and top-down methods. Bottom-up ones usually perform target segmentation, observation association, and state filtering tasks. Top-down approaches require an off-line appearance and dynamic modelling, and then perform target tracking according to the chosen methods.

These contributions usually combine several techniques.

2.2.1 Bottom-up Tracking

Bottom-up tracking approaches are usually based on motion segmentation in order to extract foreground entities from the background [94, 61, 83]. This can be performed by means of *background subtraction*, *frame differencing*, a combination of both, or *optical flow*.

Alternatively, detection can be achieved by means of detection of salient features [32, 55, 6]. In this case, regions with high curvature in space-scale images —blobs— regions with large gradients —corners—- and other significant image characteristics are extracted. However, by using this kind of approaches, any salient background point is selected as a potential target.

2.2.1.1 Pixel Segmentation

This task involves separating image regions that do not belong to the background, and extracting them. Although this issue is closely related to movement, foreground objects could remain static for an unknown number of frames while the background

Drawbacks of common approaches	Intrinsic difficulties
Bootstrapping	Illumination changes
Foreground aperture	Camouflage
Ghost	Clutter in motion
Stopped Objects	Camera motion

Table 2.1: Motion-segmentation difficulties.

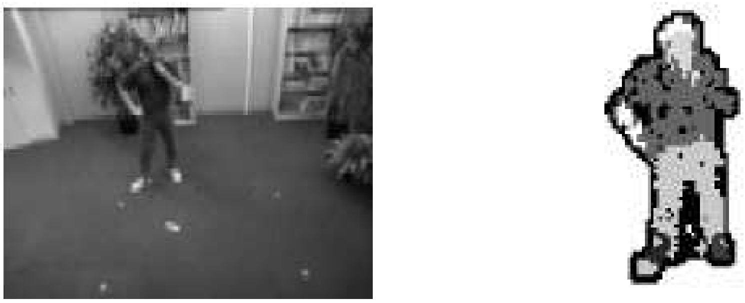
may be in motion[2].

Motion segmentation algorithms face multiple difficulties. These can be classified into two categories, since some of them are intrinsic to the problem domain, whereas others may be seen as drawbacks of the approach used, see Table 2.1. Thus, the main difficulties are the following:

- **Bootstrapping**. It refers to the problems that arise when the method requires and initialisation period, and a scene free of moving objects cannot be assured.
- **Foreground aperture**. In this case, homogeneous object in motion cause that the inner part is not segmented.
- **Ghosts**. The relocation of a background object implies changes in both the old and the new location. However, only the latter should be identified as foreground region.
- **Stopped object**. Some motion segmentation methods requires significant changes between frames to segment any pixel. Thus, if a target stop motion, the segmentation fails.
- **Illumination changes**. These completely alter the pixels characteristics, thereby resulting in a drastic increase of pixel segmentation. They may be global —thereby yielding a general highlight or shadow— or local —which are mainly caused by target shadows. Further, they can also be sudden —such as those due to changes in weather conditions, or by turning on/off a light— or gradual.
- **Camouflage**. In this case, some of the pixel features between the background and the foreground are too similar to disambiguate them.
- **Clutter in motion**. Any approach that relies on motion to perform segmentation is liable to consider as foreground any moving background pixel.
- **Camera motion.** In this case, the whole scene seems to be in motion.

In the following, papers are classified according to the approach used, and how the different difficulties are addressed is explained.

[2] Think about a person stopped momentarily at a traffic light. He or she must still be considered as foreground and, therefore detected and tracked. On the other hand, waving branches and leaves or flowing water must not be segmented, although they are in motion.



(a) Sample frame (b) Obtained segmentation

Figure 2.6: Sample frame using the approach published in [94] by Wren et al.

Background Subtraction Background subtraction is one of the most commonly used approaches for motion segmentation [71, 50]. Pixels in motion are segmented by comparing the current image and a reference one, namely, the background model. In the early days, simple methods consisted in differencing each image and a reference one, and subsequently compare the result with an a-priori set threshold [33]:

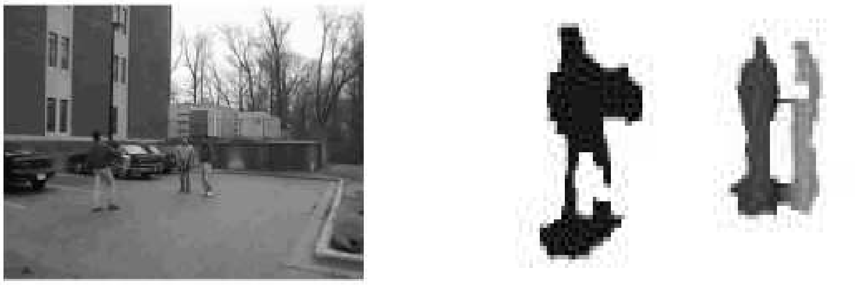
$$|\mathbf{B}_t - \mathbf{I}_t| \quad > \quad \tau, \tag{2.1}$$

where $\mathbf{B}_t$ is the reference background at time t, $\mathbf{I}_t$ the current frame, and τ a pre-set threshold. The model could be subsequently updated following a Infinite-Impulse Response filter (IIR) :

$$\mathbf{B}_{t+1} \quad = \quad (1-\alpha)\,\mathbf{B}_t + \alpha\mathbf{I}_t, \tag{2.2}$$

being α the adaptation rate that weights the current model versus the new observation. However, this method was extremely sensitive to changes in the background conditions such as lightning or due to background in motion, as well as to the camera noise. More recent approaches model either each pixel or group of pixels statistically. This allows building adaptive background models while providing robustness to the above-stated background conditions. Usually, model statistics are continuously updated in order to provide an adaptive approach.

Among the background-subtraction approaches, Wren et al. developed the *Pfinder* algorithm [94]. Each scene pixel is modelled using a Gaussian colour distribution. Thus, outliers are assumed to be foreground pixels, and are therefore segmented. Visible pixels are updated using a single adaptive filter. Segmented pixels are grouped into blobs and each blob is modelled using spatial and colour components. Blobs are associated with body parts using a log likelihood measure and tracked by means of



(a) Sample frame (b) Obtained segmentation

Figure 2.7: Sample frame using the approach published in [30] by Haritaoglu et al.

Kalman Filters (KF). However, it just attempts to detect and track one person, in upright posture, in indoor scenes. A sample frame is shown in Fig. 2.6.

Haritaoglu et al. presented the *W4* method [31, 30]. Unlike Pfinder, it aims to detect and track people, isolated or in groups, in outdoor scenes, and considering several poses. Each pixel is modelled with a range of intensity values given by minimum and maximum intensity values, and the maximum intensity difference between frames during a training period. Pixels whose values are placed outside the interval which is given by the minimum value minus a multiple of the maximum difference and the maximum value plus a multiple of the maximum difference are considered as foreground pixels. A sample frame is shown in Fig. 2.7.

The model is periodically updated considering both pixel-based and object-based methods: the former updates the values of the pixels classified as background, and the latter replaces the model parameters for those pixels classified as static foreground. Neighbour pixels are grouped and blobs are classified using heuristics. Poses are identified by means of projection histograms. KFs and textural temporal templates are used to track detected targets. However, this approach is rather sensitive to shadows and lighting changes, since the only cue is the pixel intensity.

Horprasert et al. [34, 35] implemented an statistical colour background algorithm, which models each pixel based on both brightness and colour distortion. It still needs a static background scene, but it's able to handle strong shadows and highlights. The proposed algorithm is able to classify the image pixels into four categories, namely, original, shadowed and highlighted background, and moving foreground. A sample result is shown in Fig. 2.8.

McKenna et al. [61] combined colour and gradient information in their adaptive background subtraction approach. Each pixel chrominance —given by the normalised red and green channels— is modelled using two Gaussians, one on each channel. The Gaussian parameters are updated using an adaptive filter. If one of the current chrominance values is farther from the mean more than three times the standard deviation, the pixel is marked as foreground. Using chrominance instead of RGB values, shadow detection is avoided, but it cannot cope with foregrounds of the same

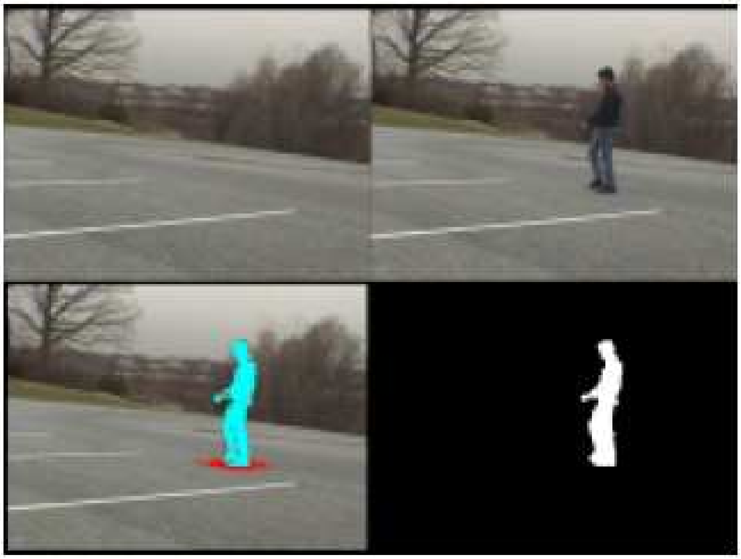


Figure 2.8: Sample frame using the approach published in [35] by Horprasert et al.

chrominance as the background. Thus, they also modelled the background pixels using the spatial RGB gradients, and pixels are also flagged as foreground if the gradient of any of the channels is out of the scope of the corresponding Gaussian. As a result, albeit foreground pixels with the same chrominance as the background can now be segmented, hard-edge shadows are also segmented. Tracking is done by means of data association.

Three levels of representation are used, namely *regions* —stable connected components— *people* —groups of regions that satisfy conditions relative to overlapping and area— and *groups* —people that share regions. People appearance is modelled using colour histograms. Visibility indexes —obtained from the probabilities that the pixels correspond to unoccluded people— are used to disambiguate occlusions. However, problems arise when several people and the background have a similar appearance. It is also assumed that the target appearance do not significantly change while the targets are grouped.

Still, shadow removal has not be properly addressed yet within a target detection framework, where shadows are considered to yield just changes in intensity, but not in chrominance. Last advances in the field —such as those contributions of Finlayson et al. [21]— need to be incorporated.

Nevertheless, none of these models can cope with background in motion. Stauffer and Grimson presented in [86] an approach focused on this issue. A colour background model is built using a Mixture of Gaussians (MoG) to represent each pixel. Thus, each Gaussian models the pixel colour distribution for one of the possible backgrounds learnt in a training period. Pixels which do not match any of the distributions are considered as foreground. The distribution weights are periodically updated according to the one that has matched the current pixel value. The least probable distribution is replaced in case none of them match the value, thereby, including long-term still foregrounds. The adaptive scheme apparently also copes with lighting and scenes changes, as well as motion from clutter. Tracking is performed by implementing a set of KFs.

Javed et al. [43] presented a method that aimed to solve most of the common segmentation difficulties: bootstrapping, ghosts, quick illumination changes, background in motion, and camouflage. It uses both colour and gradient cues. A hierarchical system is build based on three levels: *pixel*, *region* and *frame*.

At the pixel level, statistical models of pixel colour and gradients based on mixture of Gaussians are independently used to classify each pixel as potential background or foreground. At the region level, foreground pixels obtained from the colour model are grouped into regions, and the gradient model is then used to eliminate regions corresponding to highlights or ghosts. Pixel-based models are updated based on decisions made at the this level. Finally, the frame level ignores the colour-based segmentation if more than 50 percent of the image pixels are considered foreground. In this case, a global illumination change in considered, and segmentation is performed according to gradient information. Nevertheless, the ghosts are not eliminated if the background contains a high number of edges.

Frame Differencing and Hybrid Algorithms A typical temporal differencing approach segments motion by subtracting the current image from the previous one pixel by pixel. Then, pixels are segmented if the result is over a pre-defined threshold:

$$|\mathbf{I}_t - \mathbf{I}_{t-1}| \quad > \quad \tau. \tag{2.3}$$

It can also be done by considering several consecutive frames. For example, Collins et al. [11, 13] implemented an hybrid algorithm for target detection that combines an adaptive background subtraction and a three-frame differencing approach. Background subtraction techniques can provide good segmentation results, but they are extremely sensitive to scene changes due to dynamic background, lighting or extraneous events. In addition, *ghost* are usually detected when long-term stationary objects start moving —albeit statistical models eventually adapt to this situation. On the other hand, temporal differencing is very adaptive to dynamic environments and do not generate false alarms caused by ghosts, but it cannot segment all relevant pixels, and it may be rather sensitive to camera noise.

In that work, pixel intensity is taken as the representing feature. Thus, pixels whose intensity varies significantly from both the last frame and the next-to-last one are marked as moving. These pixels are clustered and a background subtraction method is applied to the inner region. Both background model and threshold are updated over time for non-moving pixels.

The approach is adapted to pan-tilt camera platforms by collecting a set of background references for known camera settings and registering the images according to selective pixel integration. They also introduced a *layered detection* algorithm: pixels are classified as stationary, transient or background according to two measures, namely, a motion trigger and a stability measure. These point out if the pixel belongs to a moving object, a stopped object or the “motion” is due to lightning changes. Foreground pixels are clustered into regions and classified as moving or stationary ones. Stationary regions constitute layers which are used to determine occlusions and motion resuming. Tracking is done by predicting next positions according to the estimated dynamic model, and convolving the object templates with candidate regions. Several scenarios are described according to the results of the two previous stages and hypotheses are launched accordingly. Finally, clutter in motion is rejected if the cumulative object displacement indicates changes in direction.

Thus, this system use a network of cooperative active cameras to detect and track people and vehicles in cluttered environments. Targets are classified into semantic categories and their activities are monitored. Once the geo-locations are extracted, symbolic data are inserted into a synthetic scene visualisation.

The algorithm proposed in [83] is also a good example of hybrid algorithms which combines frame differencing and background subtraction techniques to achieve motion segmentation. Segmentation is performed in two sequential steps. First, a fuzzy classification is carried out by according to current pixel motion on each RGB channel. Then, results are enhanced taken into account the previous segmentation result, and a background model. Finally, HSI colour space is used to eliminate shadows.

In addition to frame differencing and background subtraction, optical flow techniques have also been used to perform motion segmentation. These describe coherent feature motion between frames. These techniques independently segment moving objects, even in presence of camera motion. However, this approach is rather sensitive to noise and background in-motion, and it requires huge computational resources.

Optical Flow These methods look for coherent motion of points or features between frames. Bregler [7] presented a human-dynamics recognising method where motion is segmented according to optical flow results. An affine motion model is used for this purpose. Blobs are extracted by means of the *Expectation-Maximisation* (EM) algorithm, where the likelihood of each pixel of belonging to a particular blob depends on the coherent affine motion, HSV colour values, and spatial proximity. In order to incorporate past estimates, a bank of KFs provides priors for the EM initialisation, resulting in a MoG propagation.

Summarising, multiple techniques have been developed to tackle motion segmentation. They usually address a limited of the numerous difficulties expected. The way of solution may come from a smart combination of techniques. The different algorithms here described are summed up in Table 2.2, while pointing out the difficulties addressed.

2.2.1.2 Target Detection and Observation Association

Segmented pixels are grouped into blobs, which could be considered as an entity of interest. This is usually done according to a connected component analysis, and a

Addressed difficulty	References
Sudden illumination changes	[94, 35, 61, 86, 83, 43]
Gradual illumination changes	[30, 35, 61, 86, 13]
Camouflage	[61, 43]
Clutter in motion	[86, 13, 43]
Camera motion	[7, 13]
Bootstrapping	[86, 30, 43]
Stopped Objects	[86, 30, 13, 43]
Ghosts	[86, 13, 30, 83, 43]

Table 2.2: Motion-segmentation methods.

subsequent spatial filtering process. Then, some features can be extracted to represent a target observation, thereby classifying the target, and concluding its detection.

However, as it has been above stated, in some cases this process is enhanced by taking into account the probability of a given pixel of belonging to the target according to some statistical model.

In general, once detection has been performed, several approaches arise to keep track of the targets. New observations can be just associated to previous ones. This process can be done taking into account different cues like spatial proximity or appearance similarity. The latter may consist of a template matching between newly detected targets and the models of the previous ones. In both cases several problems must be expected due to detection failures. These mainly occur because of segmentation errors —such as those due to background clutter which mimics the target appearance, and illumination changes— and target occlusions or merging.

Depending on whether several targets and measurements are expected, the association is accomplished using nearest-neighbour techniques, or by means of Data Association Filters —such as the *Probabilistic Data Association Filter* (PDAF), the *Joint Probabilistic Data Association Filter* (JPDAF), or the *Multiple Hypotheses Tracking* (MHT) [4].

2.2.1.3 State Filtering

Usually, a prediction stage is also incorporated after associating the observation, thereby providing better chances of tracking success. Filters such as the KF [48], or subsequent extensions and improvements such as the *Extended Kalman Filter* (EKF) [2] or *Unscented Kalman Filter* (UKF) [45, 92] are commonly used.

The KF is a linear recursive estimator which predicts the next state according to a dynamic model, and updates this result in agreement with the obtained measurement. Although it has been widely used, it presents important drawbacks:

1. it requires strong assumptions about the linearity and Gaussianity of the transition model and the likelihood function;

2. it cannot cope with multiple targets and measurements;

3. and, it relies on a previous segmentation in order to provide the measurement.

These requisites are often not feasible in MTT scenarios, specially during target grouping and occlusions, or in cluttered backgrounds. Therefore, several approaches have been implemented in order to avoid these restrictions. The EKF linearises both transition and likelihood models using Taylor series expansions. The system Jacobian is computed for the predicted states, and the results are used in the updating stage. However, the EKF keeps several drawbacks:

1. posterior densities are still modelled as Gaussians;
2. the series approximation can lead to poor representations of the posterior distribution —this is specially the case on highly non-linear systems, because only the mean is propagated through the non-linearity;
3. and, although the models do not need to be linear, they still must be differentiable.

The UKF aims to propagate high-order moments through non-linear functions. A set of deterministic sample points —called *sigma points*— are selected around the mean and subsequently propagated. It can be analytically proved that it yields better approximations of the mean and covariance than the EKF. Further, there is no need to compute expensive, computationally speaking, Jacobians. However, it cannot be applied to general non-Gaussian distributions.

More general dynamics and measurement functions can be dealt with by means of *Particle Filters* (PF) [18, 3]—which are also known as *Sequential Importance Re-sampling* (SIR)— and further evolutions, such as the *Unscented Particle Filter* (UPF) [89]. These address the filtering problem when no assumption about linearity or Gaussianity is made on almost all involved probability density functions. Since the seminal paper by Gordon et al. [27], PFs have been widely used to perform stochastic estimation. The algorithm is based on Bayesian filters. Therefore, they compute a posterior probability density function (pdf) which undergoes a diffusion-reinforcement process making use of Monte Carlo simulation techniques. The reinforcement stage is accomplished by means of factored sampling. Thus, the PF approach provides a complete representation of the posterior pdf. Therefore, any statistical estimate can be computed despite non-linearities and non-Gaussianity of the involved distributions. Multiple hypotheses can simultaneously be considered, and they can be propagated even when no evidence is obtained from the current image. However, the search region is reduced, which may increase the processing speed, but the robustness could as well be cut down.

Although the asymptotic correctness of the algorithm is proved, it has several drawbacks [52]:

1. there is no information about the number of samples required for a requested precision, specially for undefined times lengths;
2. it suffers from several intrinsic problems such as *sample degeneration* or *sampling impoverishment*, depending on the whether re-sampling is used or not;

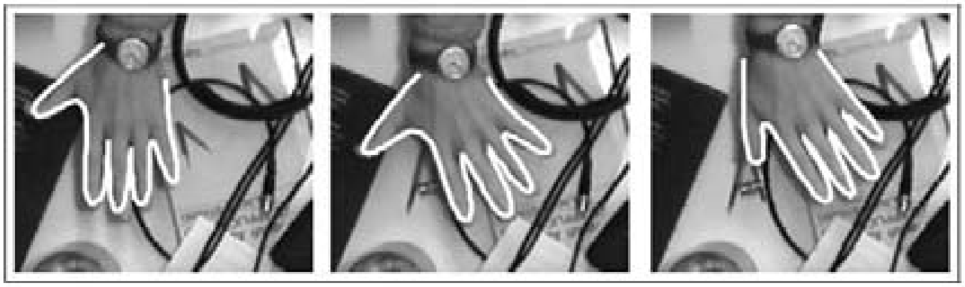


Figure 2.9: Sample frame using the approach published in [38] by Isard and Blake.

3. and finally, PFs were initially designed to keep multiple hypotheses but only for a single target; further extensions which combine information about all targets in every sample usually cause the curse of dimensionality.

In every PF approach, samples are drawn from a proposal distribution. Usually, the transition model is used as such proposal. However, problems may arise if the samples are placed in the tail of the temporal prior or if the likelihood is very peaked. De Freitas et al. [15] used the results provided by EKF as a proposal distribution. More recently, given that the UKF outperforms the EKF, this filter has been used to generate the prior samples [89].

2.2.2 Top-down Tracking

Despite these efforts, there are many situations where segmentation-from-motion, and the subsequent observation-tracker correspondence, is not possible, like in target grouping or target occlusion. Top-down approaches incorporate a-priori knowledge about the targets and the context in order to tackle these situations. Thus, these methods rely on accurate target modelling. Hence, complex templates, which should cope with an important degree of deformation, are predefined. Further, high-level motion patterns are a-priori learnt, and used to reduce the state-space search region in agreement to some state prediction.

Further, targets can be localised following an appearance segmentation, instead of a motion segmentation. This relies on feature extraction, and a subsequent exhaustive search of some feature patterns learnt during a classifier training process.

Nevertheless, model-based high-level tracking is not feasible in case this information is not available there is not enough a-priori knowledge about either the scene or the targets. Also, an accurate initialisation is often not possible. The need of adaptation when target appearances considerably evolve over time usually leads to the phenomenon known as *model drift*. In those cases, motion-based tracking usually outperforms model-based appearance or shape tracking.

Notwithstanding, numerous proposals have been presented to perform model-based tracking, while trying to overcome these drawbacks.

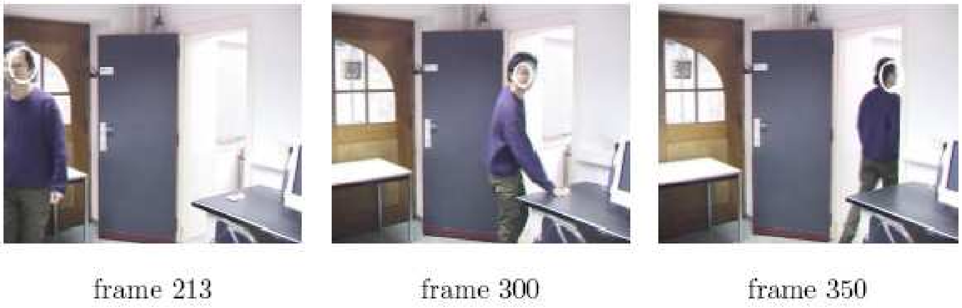


Figure 2.10: Sample frame using the approach published in [67] by Nummiaro et at.

2.2.2.1 Particle Filtering

The aforementioned PF techniques —together with complex dynamic and appearance models— have constituted a common approach [38, 56, 58, 53, 87, 16]. These techniques were introduced in the Computer-Vision field in Condensation [38, 40] by Isard and Blake, albeit they were already known in some other areas, such as Automatic Control or Artificial Intelligence. This algorithm is based on a PF framework combined with edge-based image features. Subsequently, contour tracking have been widely researched within this framework [39, 57], although this may not be the best approach in crowded scenarios because of the potential multiple occlusions. A sample performance is shown Fig. 2.9.

Nummiaro et al. [67] applied PFs using colour distributions as image features. These are approximated using histograms, which are supposed to be less sensitive to partial occlusions and rotations in depth than other appearance models such as templates. They used the HSV colour space since they claimed that it can provide robustness to changes in lightning conditions. Histograms are calculated inside an elliptic region, once the pixels have been weighted according to a kernel. A similarity function is implemented using the Bhattacharyya Coefficient (BC) [5]. Samples are represented using the centroid position in image coordinates, its speed, the length of the ellipsis axes, and a scale change. The tracker is initialised placing samples —assuming a known target model— at strategic positions. Models are only updated when the likelihood of the estimated state is over a pre-defined threshold. However, no MTT is considered —which implies that no event such as target grouping or occlusion can be analysed— and it lacks from an independent observation process, since samples are evaluated according to the histograms of the predicted image region. A sample frame is shown in Fig. 2.10.

Perez et al. [74] proposed also a PF based on a colour-histogram likelihood. They introduced interesting extensions in multiple-part modelling, incorporation of background information, and MTT. Nevertheless, it may require an extremely large number of samples, since one sample contains information about the state of all targets, dramatically increasing the state dimensionality. Further, no appearance model updating is performed, what leads to target loss in dynamic scenes.

Deutscher and Reid [16] presented an attractive approach called *Annealing Particle Filter* to recover full body motion. It aims to reduce the required number of samples. A series of weighting functions is designed from the original one by raising to a series of decreasing exponents, thereby defining a series of layers. One annealing run is performed at each time slice. The run started using the broader weighting function. At each layer, N particles are weighted, re-sampled with replacement, and used to yield a particle set for the next layer by applying Gaussian diffusion. As a result, all particles are spread around the global maximum. This final set is used to initialise the broader layer at the next time slice. Thus, the number or required samples is considerably reduced. However, pruning hypotheses with lower likelihood may lead to a single hypothesis, and therefore it could be inappropriate in cluttered environments.

The weighting function is built taken into account two image features: edges and silhouettes. Edges are obtained using a gradient-based mask over the entire image. Silhouettes are produced using a background-subtraction algorithm. Pixel weight maps are built taken into account both the proximity to an edge, and its enclosing into an extracted silhouette. In addition, two enhancements are introduced. Firstly, a soft-partition sampling is implementing by adding an amount of randomness to each parameter proportional to the variance of that parameter. In this way, samples are not wasted and the effort is concentrated on those parameters whose uncertainty is bigger. Secondly, a cross-over operator is used by combining selected particles, and thereby, tracking in parallel different sections of the search space. As they focus on motion analysis, multiple targets and unconstrained environments are not explored.

BraMBLe [42] is an appealing approach to multiple-blob tracking which models both background and foreground using MoG. However, no model updating is performed, there is a common foreground model for all targets, and it suffers from the curse of dimensionality —as all PF-based methods which tackle MTT combining information about all targets in every sample.

Occlusion events present particular difficulties which should be explicitly addressed. Wu et al. [95] address these issues using a PF by implementing a Dynamic Bayesian Network (DBN) with an extra hidden process for occlusion handling.

2.2.2.2 Gradient-descent Search

Target localisation following a gradient-descent search —*Mean-shift* tracking— has also been commonly used [10, 14, 12]. The search is performed in the basin of attraction of a spatially-smooth similarity function given by a weighted image region. Thus, in this case the search is deterministic. This is usually done according to a measure of histogram similarity between both model and candidate distributions related to the BC.

However, these methods do not work in unconstrained situations. The main drawbacks of the algorithm consist of the assumptions that the target candidate do not drastically change its appearance between time steps, and that its new location is in the basin of attraction of the similarity function, which is defined by the kernel size. Further, it is assumed that the similarity function presents a unique local maximum within the basin of attraction. In addition, only one hypothesis is considered, thereby

Figure 2.11: Sample frame using the approach published in [14] by Comaniciu et al.

limiting its effectiveness in case of occlusions or heavy cluttered backgrounds.

For instance, Comaniciu et al. [14] represented a target by an elliptic regions defined at given location, and a target model. This is obtained from the features of the normalised-to-unit-circle pixels locations, once applied an isotropic kernel. Colour is selected as image feature, and the target model pdf is approximated by means of histograms. However, it tracks just one target, initialised by hand, and the appearance model is never updated. A sample performance is shown in Fig 2.11.

Collins et al. [12] presented an appealing tracker, based also on the mean-shift algorithm, with on-line feature selection of discriminative features. It aims to maximise the distinction between the target appearance and its surroundings. Still, it tracks just one target, and may suffer from model drift, although models are anchored to the first frame, which is manually segmented. It still tracks rigid targets (or rigid regions of them), appearance changes are limited, and since MTT is not considered, interaction events are not studied. These facts cannot be seen as minor issues in real applications such as video-surveillance.

2.2.3 Bottom-up and Top-down Tracking

Algorithms which combine both bottom-up and top-down approaches have also been proposed [41, 81]. Most appealing approaches rely on the combination of several techniques. Senior et al. presented a two-level tracking system with template-based appearance models [81]. These are used in conjunction with probability masks to infer depth ordering and detect occlusions. Nonetheless, appearance ambiguities among grouped targets have not been addressed.

In [41], the probabilistic top-down tracking framework developed for CONDENSATION [40] is extended by means of *importance sampling* in order to generate samples according to a bottom-up process.

Yang et al. [97] proposed a system which specifically tackles grouping situations, albeit no filtering is carried out, and grouped targets are not independently tracked. Thus, during grouping events, just a coarse localisation can be obtained by considering that the targets are inside the group region. Therefore, grouped targets are not accurately tracked, and no complex situation can satisfactorily be faced —for instance, those in which a group of more-than-two members merge and split, see Fig. 2.12.

Kahn et al. [46, 47] developed a system called Perseus. It is a visual purposive architecture which aims to recognise gestures. The way in which the structure is

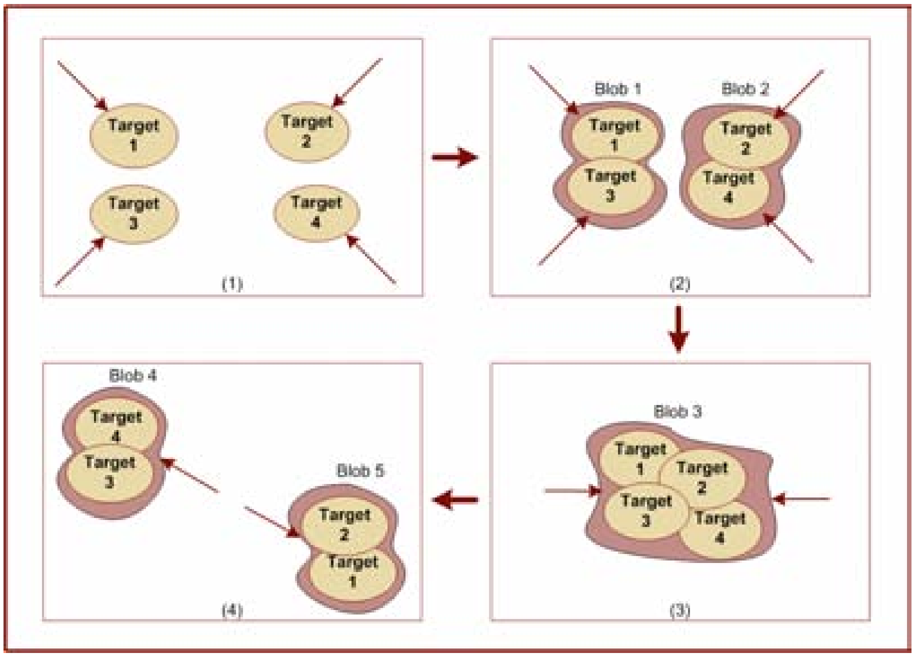


Figure 2.12: Target interaction. Keeping the identity of multiple targets which cannot be independently segmented is a challenging task. Notice the different group membership of targets in blob 1 and 4.

modularised was surprisingly novel, allowing the system to use knowledge about context and task at every stage and providing it with redundancy and independence of assumptions. It also provides an interface to higher-level systems. It consisted of six components: a *planner* is located at the higher level. It called *visual routines* which aim to detect and track selected objects. *Object representations (OR)* —background objects, light, people, objects, etc.— can be instantiated, which involves registering it at the *long term visual memory*. The object methods, such as *segment*, keep a *global segmentation map* using the image features maps located at the lower level. The considered features are intensity, edges, disparity, colour and motion. All higher levels made use of these maps to carry out their functionalities. Features parameters can be tuned according to the task and context. All object representations are also associated to *markers* which track the segmented objects.

Alternatively, several approaches take advantage of 3D information by making use of a known camera model and assuming that agents move on a known ground plane. These and other assumptions relative to a known Sun position or constrained standing postures allow the system presented in [98] to initialise trackers on people who do not enter the scene isolated.

2.3 Discussion

Summarising, an evolution in the perception of the analysis of the human motion task can certainly be noticed. Taxonomies have being refined from mere classifications according to the aim of the task, or even to criteria such as the model dimension or the sensor used, to hierarchical structures which cope which all the required functionalities. These are spread through different levels which are task-oriented.

However, this area is sill in a transition step between Image Processing and Pattern Recognition, and a more advanced view in which Cognitive Sciences provide a global understanding of the scene. The latter supplies also interactive capabilities, such as a natural language communication between a user and the system, or synthetic scene visualisations.

With respect to segmentation, it can be concluded that although remarkable advances have been achieved by presenting a wide set of different approaches, the segmentation task is still an open problem. These techniques must be enhanced to cope successfully with the numerous difficulties expected, specially in outdoor scenes. Among these difficulties, we can include lighting changes, different weather conditions, background in motion, or camouflage. Further, it is still not clear how to deal with background objects which unexpectedly move at a given moment, with the *ghost* they leave, or with foreground objects which stop momentarily. The solution may come from the combination and development of some of the existing approaches, thereby providing the system with redundancy. Taking advantage of context knowledge and making use of high-level information may also be a way of solution.

With respect to tracking, numerous approaches have been proposed to perform this task. Data-association techniques on their own are not reliable enough, since they completely depend on a proper segmentation. Prediction-updating approaches should be flexible and general enough to cope with complex environments. The combination of several of the aforementioned techniques may lead to a way of solution. Thus, for instance, EKF/UKF approaches may enhance system predictions; mean shift techniques could adjust final estimates; and several segmentation methods may be combined with prediction-updating techniques in order to provide the system with error recovery capabilities.

In our opinion, it is clear that some sort of structured architecture with cooperative levels is needed in order to cope with a such a complex problem as the analysis of human motion.

2.4 Resum

Resumint, un es pot naturalment adonar d'una evolució en la percepció de l'anàlisi del moviment humana. Les taxonomies s'han refinat de meres classificacions segons el propòsit de la tasca, o fins i tot a criteris com la dimensió dels models o el sensor utilitzat, a estructures jeràrquiques que descriuen totes les funcionalitats exigides. Aquestes s'estenen a través de nivells diferents que estan orientats a tasques particulars i diferents.

Tanmateix, aquesta àrea està encara en un pas de transició entre el Processament d'Imatges i el Reconeixement de Formes, i un punt de vista més avançat en el qual les Ciències Cognitives proporcionen una comprensió e interpretació globals de l'escena. Les Ciències Cognitives subministren també capacitats interactives, com la comunicació en llenguatge natural entre un usuari i el sistema, o les visualitzacions sintètiques d'escenes.

Respecte a la segmentació, es pot concloure que encara que els avenços notables s'han aconseguit presentant un conjunt ample d'enfocaments diferents, la tasca de la segmentació és encara un problema obert. Aquestes tècniques s'han de millorar per afrontar reeixidament dificultats esperades i nombroses, de manera especial en escenes a l'aire lliure. Entre aquestes dificultats, podem incloure canvis d'il·luminació, l'estat del temps, el fons en moviment, o el camuflament. A més, no està encara clar com tractar amb objectes del fons que inesperadament es mouen en un moment donat, o amb objectes en primer pla que s'aturen momentàniament. La solució pot arribar de la combinació i el desenvolupament d'algunes tècniques ja existents, aportant així redundància al sistema. Aprofitar el coneixement de context i l'ús que es fa de la informació d'alt nivell també poden ser el camí de solució.

Respecte al seguiment, nombroses aproximacions s'han proposat per realitzar aquesta tasca. Les tècniques d'associació de dades no són prou fiables en si mateixes, ja que depenen completament d'una segmentació correcta. Els enfocaments d'actualització de les prediccions haurien de ser més flexibles i prou generals per afrontar ambients complexos. La combinació d'unes quantes de les susdites tècniques pot conduir a una via de solució. Així, per exemple, les aproximacions d'EKF/UKF poden millorar les prediccions de sistema; les tècniques de *mean-shift* podrien refinar les prediccions finals; i diferents mètodes de segmentació es podrien combinar amb tècniques d'actualització de la predicció per dotar al sistema de capacitats per la recuperació d'errors.

En la nostra opinió, està clar que es necessita alguna classe d'arquitectura estructurada amb nivells cooperatius per afrontar un problema tan complex com la comprensió del moviment humà observat en seqüències d'imatges.

Chapter 3

A Framework to Human-Sequence Evaluation

Accomplishing HSE involves such a complexity that a structured framework is required. This is not only related to Human-Motion Analysis (HMA) —as were most taxonomies described in the previous chapter— but also to behaviour understanding. Therefore, the proposed framework must include the different required system functionalities, while making use of cognitive processes.

In this chapter, the HSE framework presented in [25] is reviewed and enhanced, see Fig. 3.1. This framework steers the efforts of our lab, are therefore its implementation constitutes the aim of the research projects in which its members are involved.

HSE defines a complete Cognitive Vision System which transforms image values into semantic descriptions of human behaviour by performing multiple bottom-up and top-down processes. Thus, its aim goes far beyond detecting, tracking and identifying the actions being performed: its goal is to apply cognition methodologies to understand human behaviour in image sequences.

Therefore, this proposal is not restricted to Image Processing and Analysis, or Pattern Recognition techniques, but it also comprehend topics related to Artificial Intelligence, Computational Linguistics, Computer Animation, and Automatic Control. For instance, Computer Animation techniques are taken into account in order to provide to a user graphical information and simulations about the situation which is taking place, as well as predictions about potential future ones; Automatic Control can come into scene to allow machine responses to recognised behaviours in human-inhabited environments, and to operate PTZ cameras.

Mainly, the implementation of HSE involves three co-operating tasks: (i) the obtention of a dynamic description of the observed human motion; (ii) the transformation of these quantitative parameters into logic predicates; and (iii) the communication of the obtained results to an human user.

Hence, multiple issues are demanded in order to accomplish HSE. At the very least, these include (i) active video camera control, (ii) target segmentation, (iii) robust and accurate MTT, (iv) target classification, (v) posture and action recognition, (vi) facial expression analysis, (vii) behaviour understanding, and (viii) communication of those inferred conceptual interpretations to human operators. The last task can be achieved by means of NL text generation —by applying syntax rules to those instantiated conceptual primitives— and by the synthesisation of virtual environments from this conceptual information.

The computational knowledge of the three different *channels* of human motion, namely the motion of *agents* (trajectories), *bodies* (postures and actions), and *faces* (expressions and emotions), is linked together in the same discourse domain.

Unfortunately, adversities common to other Computer Vision areas could cause system failures, for instance due to acquisition conditions, uncontrolled illumination, shadows, cluttered backgrounds —possibly in motion— etc. In addition, dealing with people entails numerous special difficulties such as posture changes, huge appearance variability, or unforeseeable motion changes. Hence, location, orientation and linear or angular speeds may not be enough to describe human motion, since great structural changes should be expected. At least, these changes are restricted: a basic structure is preserved by maintaining a logical body part order —depending on the pose— and a relative aspect ratio between body parts. Bounded positions and speeds should also be expected.

Alternatively, understanding people involves, as an essential aspect, intentionality. Relations between agents, and between an agent and its environment, must be taken into account in order to explain some situations. Moreover, conceptual interpretations of motion include a degree of uncertainty due to the inaccuracy of the semantic terms used to explain human behaviour. Therefore, considering the context will be a determining factor.

Due to this complexity, an HSE system is here presented as a highly modularised and hierarchically organised framework. Thus, multiple co-operating modules are defined through the different levels. They work following both top-down and bottom-up approaches in a closed loop, thereby defining the interactions of different Computer Vision algorithms with other components, such as human behaviour modelling and NL text generation. This is done while taking into account the uncertainty generated during motion naming, i.e. the textual explanation of perceived motion. HSE requires intermediate models of human motion to associate geometric knowledge with conceptual statements. Thus, each level exploits the a-priori knowledge provided by models and context.

Levels are defined according to main functionalities. The whole structure is highly interconnected, and each level receives inputs from higher and lower ones, providing the system with redundancy. The inter-level communication can be seen in three different ways: first of all, a data stream is provided to the higher levels by lower ones including all the results obtained in the bottom-up process; secondly, higher levels feed back the lower ones in a top-down process, so that the whole procedure can be enhanced; at the same time, higher levels can act on the lower ones by tuning the parameters, and selecting different operation modes, models or approaches depending on what is known about the current scene, and what goals are pursued.

Information is processed according to the following flows: on the one hand, visual sensors provide evidences about the real world to the system at the *Active-Sensor Level* (ASL). Then, the next three levels process and analyse the image sequence. At the *Image-Signal Level* (ISL), the sequence of image data is processed by segmenting potential targets. The resulting foreground regions are the basis for the following level: the *Picture-Domain Level* (PDL). Possible segmentation errors generated at the ISL are handled here by means of representation, classification, and tracking techniques. At the *Scene-Domain Level* (SDL), the 3D configuration of the scene is used to compute the parameters of each agent within its 3D environment.

Results obtained at either the PDL or the SDL are forwarded to the two higher levels which perform the description of the obtained quantitative results, and finally carry out a principled reasoning over them. Hence, the *Conceptual-Integration Level* (CIL) instantiate semantic predicates for a given agent and time step. These qualitative descriptions are used to generate interpretations of its motion, as well as conceptual relationships of the agent and its environment. Instantiated predicates are fed forward to the *Behaviour-Interpretation Level* (BIL), where the expected temporal evolution of descriptions are a-priori modelled in order to generate coherent spatio-temporal interpretations.

On the other hand, a top-down process closes the loop by feeding back the lower levels with the results obtained at the higher levels. For example, the behaviour interpretation generated at the BIL is used at the CIL to avoid an exponential explosion of situation hypotheses; the current inferred situation permits to disambiguate tracking scenarios at the SDL; an scene analysis provided by the SDL allows the PDL to cope with the effect of the view point; the ISL can enhance the segmentation by taking into account the presence of tracked targets.

Finally, the *User-Interface Level* (UIL) provide NL descriptions of situations and behaviours that occur within the scene. Further, an interactive Graphical User Interface (GUI) allows a single human operator to monitor a significant area of interest. An example of an HSE into operation is shown in Fig. 3.2.

3.1 Machine Interface Levels

These levels, that can be seen as the lowest and highest ones in the hierarchical architecture, constitute the interface between the Human-Sequence Evaluation system and, on one side, the real world, and on the other, the user.

3.1.1 Active Sensor Level (ASL)

This level acquires raw video sequences and information about camera parameters. Pieces of reality can be captured by the cameras according to the kind of sensor used and the visual field. Thus, this level includes hardware devices, such as the camera itself and the acquisition cards, and models to deal with these devices[1]. Such models consist mainly of three modules which define the camera, the digitiser and the encoder/decoder, respectively.

The first module deals with the optic parameters —the focal distance, the optic centre, the diaphragm aperture, the exposure time— and the camera position and orientation. The second module defines the transformation from camera signals to image values, namely, the image resolution, the pixel depth, the number of frames

[1] Such as a pin-hole camera model, a stereo camera model, or a model for PTZ camera.

which are acquired per second, and the acquisition mode, colour or grey-scale. The latter is used when the image sequence must be encoded for transmission or security reasons.

Multiple cameras can be used and combined to produce a single scene mosaic. A panorama can also be obtained by making use of pan-tilt cameras.

Finally, being the sensors active, the system is allowed to modify the camera parameters depending on the task and environment conditions. Thus, the camera module could modify the focal length —zooming in or out allowing active vision— or the aperture depending on the light conditions; and the viewpoint —panning and tilting. The digitiser module could change the image resolution and pixel depth when a higher accuracy is required; the frame rate could be also adjusted to the scene dynamics.

3.1.2 User Interaction Level (UIL)

At this level, human-computer communication is carried out. Multiple modules can be included in order to bring new interaction capabilities, such as natural language, visual descriptions, or audio interactions.

3.1.2.1 Natural Language (NL)

One of the main tasks of the UIL is to provide a natural-language description of what is actually happening within the scene. The quantitative information generated at lower levels is associated with qualitative semantic terms such as verbs, nouns, adverbs and adjectives, and it is used to generate natural sentences by means of syntactical, morphological, and orthographic rules.

The first step involved is the elaboration of a corpus made by native speakers. Then, a technique is required to facilitate the conversion of conceptual information into linguistic outputs. At the lexicalisation step, the logical predicates imported from the BIL are clustered into appropriated lemmas by means of an ordered set of language-dependent rules.

After that, Text Generation Rules (TGRs) are specified in order to infer the syntactical order of the input lemmas. Subsequently, morphological rules are applied over the set of lemmas to properly inflect the linguistic elements (number, gender, tense...). Lastly, orthography provides punctuation symbols to the sequence of words to be delivered to the final user.

3.1.2.2 Graphical User Interface (GUI)

Keeping track of multiple people, vehicles, and their interactions among them and with other objects, within a complex scene is a difficult task. A GUI allows a single human operator to effectively monitor a significant area of interest. Thus, the GUI automatically places virtual agents representing people and vehicles into a synthetic view of the environment.

This approach has the benefit that visualisation of scene events is no longer tied to the original resolution and viewpoint of a single video sensor. Through this interface, the user can act on individual sensor units, modify the system parameters, select one particular approach, and ask for situation descriptions, behaviour explanations, and synthetic simulations.

An audio-based interactive environment can also be here considered to enhance the user interaction.

3.2 Image Analysis Levels

These levels perform image processing, and a subsequent data analysis according to 2D-picture or 3D-scene representation.

3.2.1 Image Signal Level (ISL)

At this stage, the task is to process the bit flow that represents the image sequence provided by the sensor modules. This is carried out, frame by frame, by the ISL, whose main goal is to segment foreground objects. The results obtained at this level involves two main issues, namely *foreground segmentation* and *data representation*. Several pre-processing tasks such as noise filtering are also carried out at this level.

As a result of the current level, a compact image representation of the foreground objects is given to the PDL.

This level receives also feed-back from the PDL Level. Thus, information about models and context can enhance the level performance in both accuracy and robustness senses. The ISL can act on active sensors in order to modify the camera parameters. Thus, a better segmentation could be obtained.

3.2.1.1 Image Feature Selection

A set of image features are extracted from each frame. Several cues can be used depending on the application aim, which assumptions and heuristics are considered, and the methods chosen to achieve the goals from each task. The different image features to be used are selected by the higher levels and extracted at this level from the image sequence. They are used to carry out several tasks including *segmentation*, *classification*, *tracking* or *identification* through the different levels. Different cues can be taken into account to carry out the same task in order to provide the whole process with redundancy, and thus with robustness. However, not all the selected cues will be used to perform the same task. Thus, different subsets of cues could be more appropriate to the different tasks.

This feature set could include intensity values, colour, gradients, disparity, motion, texture, curvature, lines, edges, shape and depth.

It will be desirable to allow higher levels to tune —according to the current scenario— the cue value range of interest, resolution, thresholds, colour spaces, motion sensitivity, texture patterns, and other parameters of interest.

3.2.1.2 Foreground Segmentation

This task involves separating image regions that do not belong to the background, and extracting them. Targets can be segmented following an appearance segmentation —which requires high-level information— or by means of motion segmentation. This module may implement also several methods to perform the latter, such as temporal differencing, optical flow, background subtraction, or a combination of these.

3.2.1.3 Image Data Representation

This task may be seen as placed in the interface between the ISL and the PDL. Features are here manipulated to obtain representations which can be handled by the PDL. In addition, segmented objects are represented in a compact way in order to reduce the complexity of the search space and remove confusing elements.

This representation can be foreground-oriented or image-oriented. Among the former, points —centroid, median coordinate, contour points, axis points— bounding boxes, blobs, contours or more elaborate structures made of segments or blobs can be used. Among the latter, spatial and spatio-temporal transformations (Fourier, PCA, Wavelets, DCT, histograms), and features points representations can be taken into account.

3.2.2 Picture Domain Level (PDL)

The purpose of this level is to carry out an image analysis in order to perform the following tasks: a *classification* of the targets already segmented by the ISL, and the *tracking* of them through the sequence of frames. As a result, the tracked labelled targets are supplied to the PDL Level.

This level receives also feed-back knowledge from the PDL level —such as projected 3D model and information about the scene— and from the CIL —concerning analysed situations. Besides, the level is acting on the ISL. Thus, it would be possible to choose which cues to extract, the segmentation method or the representation approach, as well as to perform background updating according to high-level information, threshold tuning, etc.

3.2.2.1 Target Classification

Targets can be classified according to whether they are agents or objects, that is, by taken into account if they are targets with intentional capabilities, or not. Depending on the chosen resolution, regions such as body parts can be also classified.

Again, multiple approaches can be taken into account, in this case depending on whether a shape model is used or not. If no shape model is used, the target can still be classified according to its shape, appearance features, or movement. The former may be based on representations such as projection histograms, or on structural relation like key points order, curvature, symmetry, or aspect relations such as compactness. Features classifiers are based on (skin) colour, texture, intensity, or salient points. Therefore, some heuristics should be used in the classification process. The proper heuristics are selected by higher levels depending on the available information about the scene. In any of these two kind of approaches, the classification can be done after the representation of the segmented entities is achieved.

On the contrary, movement classifiers analyse the periodic nature of the movement, or whether there are any kinematic restrictions. This requires that the targets have been tracked for a time period. Both kind of classifications can be combined.

On the other hand, an a-priori shape model can be used to perform the classification. Two options can here be considered: using a 2D model or a 3D model projection. The former would be located at the PDL. Several object representations can be chosen allowing the comparison between segmented target and the model. The second option would consist in using the projection of a 3D model. This model would be located at the SDL and its projection should be given to the PDL in order to compare it with the current image representation. Features used to perform the comparison include edges, contours, blobs texture, colour or intensity, segments and joints, depth, or movement. In this case, structural models could provide possible human configurations.

Once a target has been segmented and classified, it is considered that the detection has been performed.

3.2.2.2 Target Tracking

This phase involves matching targets in consecutive frames, thereby establishing coherent target relations over time. The process is based on predicting the target's next state and evaluating the results according to what is found in the current image. The state could include information about spatial position, speed, shape or appearance.

Hence, *transition models* are required. They describe the target's motion, providing a set of equations. It is possible to distinguish between *dynamic* and *aspect* models. The former deals with global position changes, whereas the latter models the shape and appearance changes. These models can be locally located at the PDL, or provided by higher levels according to learn patterns, 3D projections, etc.

Several context restrictions can also be used in order to narrow the search. They are usually provided by higher-level feedback, although it is also possible to learnt them over time. These constrictions could include speed limits, forbidden areas given by collisions, allowed shapes, et cetera.

3.2.3 Scene Domain Level (SDL)

At this stage a higher-level tracking process is performed taking into account 3D knowledge. Thus, the results provided by the PDL are refined using a 3D human model[2]. A 3D scene model can also be used, thereby providing context restrictions as well as a set of heuristics. Using a *correspondence model*, the level knowledge is fed-back to the PDL. This model may be placed in the interface between both PDL and SDL.

[2] It is interesting to remark that the information flow can be very flexible. For example, a segmentation based on depth cues represents a direct collaboration between the ISL and the SDL.

Several kinds of shape models can here be used to represent the human body. Thus, the model can be made of sticks, polygons, 3D surfaces (such as patches) and volumes. Sticks are used to model the skeleton while the other representations are used to model the flesh. Both components, skeleton and flesh, can be used simultaneously. Among the volume representations, it is possible to use several geometric primitives such as cylinders, cones, spheres or prisms. This last representation could be used to take into consideration not only structural and kinematic properties, but also dynamic ones.

Again, this level is acting on the lowest levels by selecting proper models, parameters and approaches according to the current 3D knowledge about the current scene.

3.3 Cognitive Levels

The following two levels carry out first a description of the current scene situation, and subsequently perform spatio-temporal reasonings over the inferred descriptions, thereby explaining agent potential behaviours.

3.3.1 Conceptual Integration Level (CIL)

This level aims to describe conceptual situations according to the data given by both PDL and SDL[3]. Thus, all the conceptual knowledge used for HSE is implemented at the CIL as a set of logic predicates. This level should cope with the temporal and uncertainty aspects inherent in the integration of numerical values into conceptual terms. This include dynamic occurrences, uncertainties of the state estimation process, and intrinsic vagueness of conceptual terms.

Two source of knowledge are established. Firstly, the quantitative knowledge embedded in the numerical state vector such as position, speed or orientation values. Since the state vector is determined by the nature of the parameters used for tracking, semantic terms will refer to dynamical, positional and postural properties of the human agent. These quantitative parameters are associated to semantic concepts like *moving*, *slow*, *small*, and *crawling* or *lying*, along with a fuzzy degree of validity characterising how good a concept matches the numerical parameter value.

Secondly, spacial relationships of each agent w.r.t. its environment are derived by considering the positions of the agents and other static objects in the scene. This is implemented by applying a distance function between the positions of the different agents and objects in the scene. Subsequently, a discretisation of the resulting distance value is obtained by using fuzzy logic, thus allowing to instantiate logic predicates, such as the presence or proximity of other agents or objects in terms such as *left* or *near*, and events such as *grouping* or *splitting* and *occluded*. Other spatial relationships are derived by considering the semantics of the scene, so a conceptual scene model is required to identify specific locations within the environment, or events such as entering or exiting.

[3] Again, another example of flexible collaboration is given by the fact that the CIL can infer conceptual situation from 2D results provided by the PDL.

All the aforementioned conceptual knowledge generated at the CIL at each time step is called a *situation*. As a result of this stage, conceptual descriptions or situations are given to the Behaviour Interpretation Level. Further, the context information provided by the BIL is used to prune the number of potential situation hypotheses.

3.3.2 Behaviour Interpretation Level (BIL)

By means of spatio-temporal reasonings —based on semantic terms— this level aims to explain behaviours and intentions. Then, the inferred information could be used to predict future situations. Due to the impossibility of modeling all possible human situations, the expected evolution of situations to be described are modelled a-priori for improving spatio-temporal interpretation. That means, the BIL selects those situations to be instantiated at the CIL, thus allowing to interpret the intentions of the agent in a goal-oriented manner.

There exists a data flow in two directions, top-down and bottom-up, which may restrict the combinatorial explosion of data and the reproduction of errors. On the one hand, top-down data flow is generated for hypothesis verification. This information may be forwarded to the lower levels of the architecture to assist segmentation and tracking procedures, thereby constraining the uncertainty in lower levels. On the other hand, bottom-up data flow corresponds to potential semantic descriptions —hypotheses made at the CIL based on estimations— derived from motion analysis processes carried out at the PDL and SDL.

3.4 Discussion

HSE is focused on the transformation of image data into semantic descriptions in natural language, and vice-versa. This transformation process implements motion understanding in the Computer Vision domain. HSE involves different topics such as acquisition; detection and tracking; recognition; interpretation; human behaviour modeling; and NL textual generation and synthetic visual representation. These main steps are organised within an architecture based on a set of cooperating modules, each one devoted to a specific task.

The proposed HSE architecture embeds three goals. Firstly, the estimation of spatio-temporal descriptions of human motion in terms of quantitative knowledge —this is done at the ASL, ISL, PDL and SDL. Secondly, the association of geometric parameters with semantic predicates —what is done at the CIL and BIL. Thirdly, the generation of NL texts explaining the meaning of observed human motion patterns, and the synthetic visualisation of them —performed at the UIL.

Three information channels can be considered depending on the image resolution and camera views. Thus, a trajectory analysis can be considered for the whole scene, thereby detecting and tracking the agents within it. By making use of closer cameras or using active camera zooms, their body posture can be evaluated. Finally, with a higher resolution, their face can be resolved sufficiently well, and facial emotions can be analysed. It is also interesting to integrate these three modes into a single application environment.

This architecture must be considered as a framework to perform HSE, that is, a way to organise the different tasks that can be carried out by a Cognitive Vision System. However, it must not be seen as a fixed structure, but a rather flexible one dependent on the goal of the current application. That means that non-relevant tasks can be avoided and the implementation does not have to strictly follow this structure. For instance, a *counting people* or a video-surveillance application may not need any SDL functionality, and perhaps only a zenithal 2D view is required. A fixed segmentation method and cues can be selected avoiding having the need to select different ones.

The ISE Lab aims to design a Cognitive Vision System for human motion and behavior understanding, followed by the communication of the system results to end-users, based on two main goals: the first goal is to determine which interpretations are feasible to be inferred from three different categories of human motion, i.e. the motion of agent, body and face. The second objective is set to establish how these three types of interpretations can be linked together (i) to coherently evaluate the human motion as a whole in image sequences, and (ii) to communicate inferred interpretations using natural-language texts or virtual environments as a visual language.

The rest of this work is focused on the ISL, PDL and CIL within the HSE framework. The mail goal is to perform a robust MTT. Therefore, detection, estimation and adaptation tasks are here addressed. This requires target segmentation, representation and tracking. Further, model adaptation, target interactions, and extraneous events demand situation description.

It is worth to say that the proposed system is also prepared to be integrated in the near future in a complex HSE architecture. Obtained results are currently being forwarded to further conceptual and behaviour interpretation. High-level information about the context and current situations provided by cognitive levels of the HSE framework will enhance tracking performances. Make future use of multiple active cameras from several point of views is also feasible, and will solve problems derived from the use a fixed point of view.

3.5 Resum

ASH se centra en la transformació de les dades d'una imatge a descripcions semàntiques en llenguatge natural, i viceversa. Aquest procés de transformació comporta la comprensió del moviment en el camp de la Visió per Computador. ASH implica temes diferents com l'adquisició; la detecció i el seguiment; el reconeixement; la interpretació; el modelatge del comportament humà; i la generació de textos en llenguatge natural així com la representació visual sintètica. Aquests passos principals s'organitzen dins d'una arquitectura basada en un conjunt de mòduls que cooperen, però on cada un està dedicat a una tasca específica.

L'arquitectura d'ASH proposada s'arrela en tres objectius. En primer lloc, l'estimació de descripcions espaitemporals del moviment humà en termes de coneixement quantitatiu —això es fa a l'ASL, ISL, PDL i SDL. En segon lloc, l'associació de paràmetres geomètrics amb predicats semàntics —que es fa al CIL i BIL. Finalment, la generació de texts en llenguatge natural explicant el significat dels patrons de moviment observats, així com la visualització automàtica d'animacions virtuals– realitzats a l'UIL.

Es poden considerar tres canals d'informació depenent de les vistes de la imatge, així com la resolució de la càmera. Així, es pot considerar una anàlisi de trajectòries per a l'escena sencera, detectant i seguint així els agents dins d'aquesta. Fent ús de càmeres més properes o utilitzant càmeres actives amb zoom, la postura del cos es pot avaluar. Finalment, amb una resolució molt més alta, la cara es pot analitzar suficientment bé, i es poden analitzar emocions facials. És també interessant integrar aquests tres modes en un domini d'aplicació únic.

Aquesta arquitectura s'ha de considerar com un marc per realitzar ASH, és a dir de, una manera d'organitzar les diferents tasques que poden ser fetes per un Sistema de Visió Cognitiu. Tanmateix, no s'ha de veure com una estructura fixa, sinó flexible i dependent de l'objectiu de l'aplicació a desenvolupar. Això significa que es poden evitar tasques no pertinents ja que l'aplicació no ha de seguir estrictament aquesta estructura. Per exemple, un *comptador de persones* o una aplicació de vigilància de vídeo pot no necessitar funcionalitat de SDL, i potser només és demanat un punt de vista zenital 2-D. A més, es pot seleccionar un mètode de segmentació fix evitant tenir la necessitat de seleccionar-ne diferents.

Els propòsits de Laboratori d'ISE per dissenyar un Sistema de Visió Cognitiu per a la comprensió del moviment i comportament humans, seguida per la comunicació dels resultats a usuaris finals, estan basats en dos objectius principals: el primer objectiu és determinar quines interpretacions són factibles per ser inferides en les tres categories diferents de moviment humà, i.e. d'agent, cos i cara. El segon objectiu és establir com aquests tres tipus d'interpretacions poden ser connectats junts (i) per avaluar coherent i globalment el moviment humà en la imatge , i (ii) comunicar les interpretacions inferides mitjançant texts en llenguatge natural o en entorns virtuals com a llengua visual.

La resta d'aquest treball se centra en l'ISL, PDL i CIL dins de l'estructura ASH. L'objectiu és realitzar un MTT robust. Per això, les tasques de detecció, representació i adaptació són encarats tot seguit. Això exigeix la segmentació d'objectes, la seva representació i el seu seguint. A més, per a la descripció de situacions es requereix l'adaptació dels models, les interaccions entre objectes, i l'anàlisi esdeveniments externs.

Cal dir que el sistema proposat també es prepara per ser integrat en el pròxim futur en una arquitectura ASH més complexa. Els resultats obtinguts s'estan enviant actualment per promoure la interpretació conceptual del comportament. La informació d'alt nivell sobre el context i les situacions actuals proporcionades per nivells cognitius realçaran els rendiments aconseguits. L'ús futur de càmeres actives múltiples des d'uns quants punt de vistes també és factible, i resoldrà els actuals problemes obtinguts per l'ús un punt de vista fix.

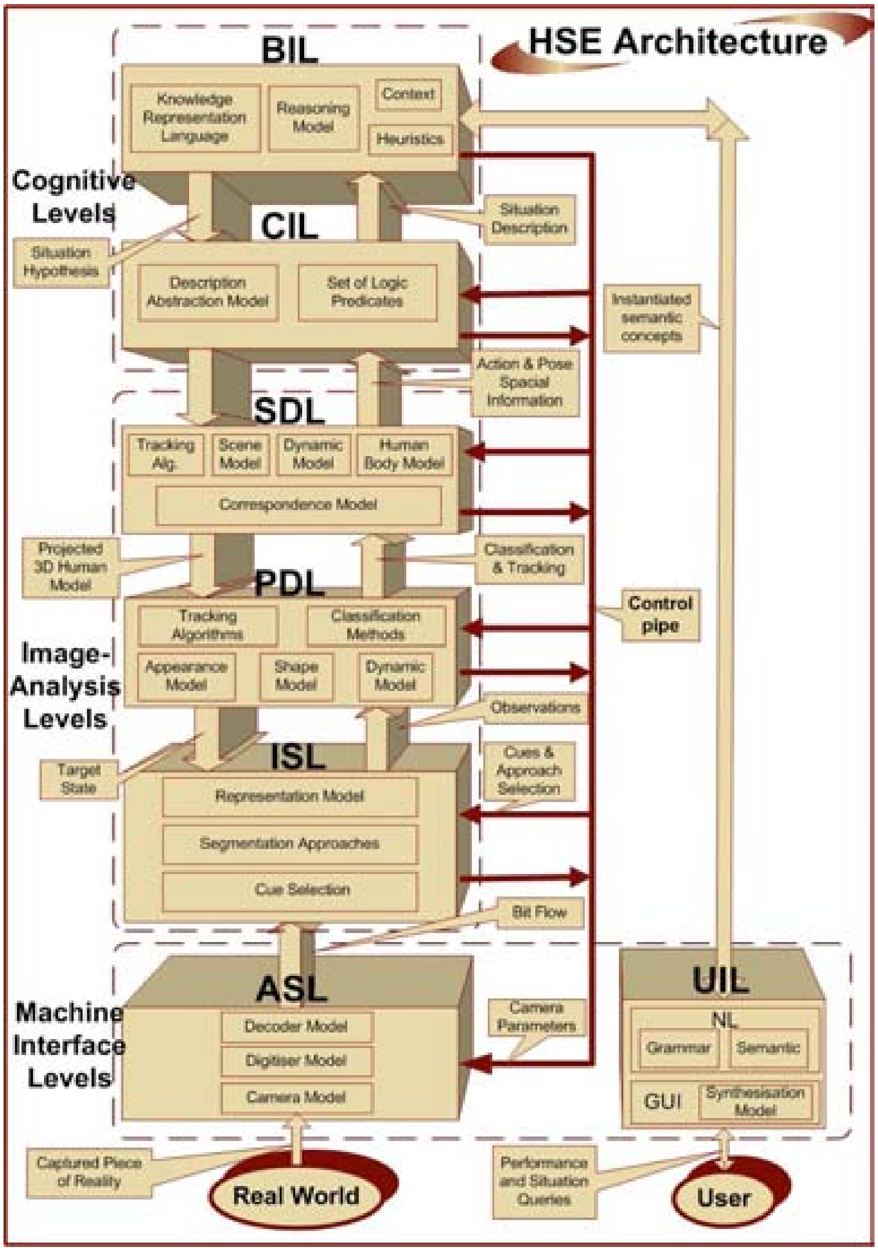
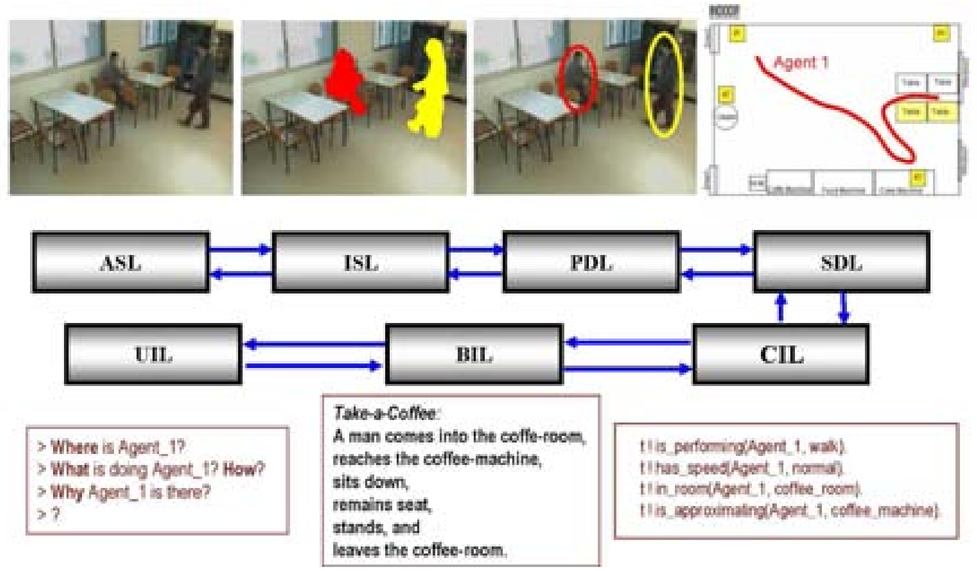


Figure 3.1: HSE framework evolved from [25]. Levels are defined according to main functionalities. Thus, each level performs some general task such as providing a machine interface —ASL, UIL— processing and analysing the image sequence —ISL, PDL, SDL— and describing and reasoning over the obtained quantitative results —CIL, BIL.



> Where is Agent_1?
> What is doing Agent_1? How?
> Why Agent_1 is there?
> ?

Take-a-Coffee:
A man comes into the coffe-room,
reaches the coffee-machine,
sits down,
remains seat,
stands, and
leaves the coffee-room.

t ! is_performing(Agent_1, walk).
t ! has_speed(Agent_1, normal).
t ! in_room(Agent_1, coffee_room).
t ! is_approximating(Agent_1, coffee_machine).

Figure 3.2: Example of an HSE system into operation in an indoor scene.

Chapter 4

Multiple-Target Tracking based on Particle Filtering

In this chapter, the first proposal to tackle multiple-target tracking is developed. Here, tracking is performed by enhancing the particle filtering framework. This approach has been widely explored by several previous algorithms, as discussed before. Despite this effort, many undesirable effects still remain. These are here highlighted, and some proposals are presented in order to cope with them.

4.1 Framework Outline

A probabilistic framework is commonly used as a way to perform tracking in order to deal with uncertainty over time [80]. Classical approaches, such as the *Kalman Filter* [48], rely on linearity and Gaussianity assumptions about the involved distributions, see Appendix D.

More recent works make use of *Bayesian filters* combined with *Monte Carlo Simulation* methods in order to deal with nonlinear and non-Gaussian transition models and non-Gaussian likelihood functions [77, 59]. Subsequent developments have introduced a re-sampling phase in the sequential simulation-based Bayesian filter algorithms [27]. These approaches are known as *particle filtering* within the control field or *survival of the fittest* in Artificial Intelligence.

Such methods were first introduced in the computer-vision research area by Isard and Blake, and renamed as *Condensation* [38, 40]. They have been widely used in recent years [41, 15, 91, 57, 89, 58, 42, 74, 67, 95, 16]. Excellent reviews have been presented by Doucet [18], and by Arulampalam et al. [3]. Further, comprehensive treatments are given in [19, 76]. However, several important drawbacks remain, as stated by King and Forsyth [52]. Despite the great number of improvements that have been already introduced, many open issues prevent from stating that particle filters are able to solve unconstrained tracking problems.

In order to perform the following analysis, a strong probabilistic background is required. Basic statistics are summed up at Appendix C. For further proofs and explanations, see [51, 80]. Simulation techniques are covered in [77, 59].

4.2 Probabilistic Framework

From a probabilistic point of view, the tracking problem involves dealing with stochastic processes. These are series of time-slices describing the state of all entities within the scene. Each time-slice consists of a set of random variables[1]. Two kind of variables can be distinguished, namely unobservable state variables at time t, denoted as $\mathbf{S}_t$, and observable evidence variables, denoted as $\mathbf{E}_t$. The interval between time-slices depends on the frame rate[2].

In order to specify the dependencies among the different variables, these are ordered following a temporal criterion, i.e, taking causality into account. This means that the variables from previous time-slices cause the values of subsequent time-slice variables. Thus, it should be possible to specify conditional probability density functions for all variables given their predecessors, from now on called *parents* [80]. On the order hand, variable conditional independence within a time-slice could be established given a set of parents.

However, since every time-slice must be considered, several problems arise:

1. There is an unbounded set of conditional probability density functions.

   This problem can be overcome making the *homogeneous process assumption*:

   > *The process is governed by laws that do not change themselves over time.*

   Hence, there is no need to specify all conditional pdf but only those within a representative time-slice.

2. There is an unbounded set of parents.

   Let us consider separately the effect of the parents on the state variables $\mathbf{S}_t$ and on evidence variables $\mathbf{E}_t$. Considering the *Markov assumption* on both states and evidences, it is possible to get over this problem:

   (a) *The current state $\mathbf{S}_t$ depends only on a finite history of previous states, $\mathbf{S}_{t-\tau:t-1}$.*

   Therefore, the *state* could be defined as the information needed to make the future independent from the past given the present. In *first-order*

[1] The following notation is here used: related to variables, non-bold lowercase denotes scalars, whereas bold lowercase denotes vectors, and matrices are given by bold uppercase. In a probabilistic context, uppercase denotes probability density functions (pdf) and random variables; lowercase denotes probabilities and variable instances. $\mathbf{X}_{t_1:t_2}$ denotes a variable set from time $t = t_1$ to $t = t_2$.

[2] This parameter is set considering the possible dynamics of the targets that could appear in the scene.

Markov processes the current state only depends on the immediately previous one. Here, this kind of Markov processes is considered, since it is always possible to reformulate a non first-order Markov process as a first-order one by increasing the state variable set [80].
Thus, the state variables are conditional independent of all other previous variables given the previous state:

$$P(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}) = P(\mathbf{S}_t \mid \mathbf{S}_{t-1}) \,. \tag{4.1}$$

The latter conditional pdf is called the *transition model*. In the tracking problem here presented, the transition model will be split into a *dynamic model*, which considers the target's motion, and an *aspect model*, which captures the target's shape and appearance.

(b) *The evidence variables at time t* $\mathbf{E}_t$ *depend only on the current state* $\mathbf{S}_t$*.*

Hence, the evidence variables are conditional independent from all other variables given the state:

$$P(\mathbf{E}_t \mid \mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}) = P(\mathbf{E}_t \mid \mathbf{S}_t) \,. \tag{4.2}$$

In this case, the latter conditional pdf is called the *observation* or *sensor model*. It is also called the *likelihood* function since it forecasts how likely an observation is, once the state is given. It models a causal relation: it is the current state which causes the obtained evidence.

Thus, the developments within the scene can be modelled as a Hidden Markov Model (HMM) where $\mathbf{S}_t$ constitutes the unobservable or hidden state variables and $\mathbf{E}_t$ the observable evidence variables at time t. The HMM is described by:

- an initial prior state density function, $P(\mathbf{S}_0)$;
- the transition model[3], $P(\mathbf{S}_t \mid \mathbf{S}_{t-1})$ for $t \geq 1$;
- the likelihood function, $P(\mathbf{E}_t \mid \mathbf{S}_t)$ for $t \geq 1$;
- both assumptions on variable conditional independence stated in Eqs. (4.1) and (4.2):
  - the state variables, $\{\mathbf{S}_t; t\in \mathbb{N}\}, \mathbf{S}_t \in \mathbb{R}^{n_s}$, given the immediately previous state $\mathbf{S}_{t-1}$; $n_\mathbf{s}$ denotes the state-space dimension;
  - the evidence variables, $\{\mathbf{E}_t; t\in \mathbb{N}\}, \mathbf{E}_t \in \mathbb{R}^{n_e}$, given the corresponding state variable; $n_\mathbf{e}$ denotes the evidence-space dimension.

[3] A sequence of random variables $\mathbf{S}_t$ satisfying the Markov assumption is called a *Markov chain*. If the conditional probability density functions $P(\mathbf{S}_t \mid \mathbf{S}_{t-1})$ are time independent, the Markov chain is called *homogeneous*. However, it does not mean that the probability density functions of consecutive states are the same, $P(\mathbf{S}_t) = P(\mathbf{S}_{t-1})$, a fact that is called *stationarity*.

Given both models and assumptions, it is possible to specify the complete joint density function:

$$\begin{aligned}
P\left(\mathbf{S}_{0:t},\mathbf{E}_{1:t}\right) &= P\left(\mathbf{E}_t \mid \mathbf{S}_{0:t},\mathbf{E}_{1:t-1}\right)P\left(\mathbf{S}_{0:t},\mathbf{E}_{1:t-1}\right) && \text{(cond. prob.)}\\
&= P\left(\mathbf{E}_t \mid \mathbf{S}_t\right)P\left(\mathbf{S}_{0:t},\mathbf{E}_{1:t-1}\right) && \text{(Markov on ev.)}\\
&= P\left(\mathbf{E}_t \mid \mathbf{S}_t\right)P\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1},\mathbf{E}_{1:t-1}\right)P\left(\mathbf{S}_{0:t-1},\mathbf{E}_{1:t-1}\right) && \text{(cond. prob.)}\\
&= P\left(\mathbf{E}_t \mid \mathbf{S}_t\right)P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)P\left(\mathbf{S}_{0:t-1},\mathbf{E}_{1:t-1}\right) && \text{(Markov)}\\
&\dots\\
&= P\left(\mathbf{S}_0\right)\prod_{k=1}^{t}P\left(\mathbf{E}_k \mid \mathbf{S}_k\right)P\left(\mathbf{S}_k \mid \mathbf{S}_{k-1}\right), && (4.3)
\end{aligned}$$

which specifies the probability of every event within the scene and, therefore, can answer every probabilistic query about it. Unfortunately, it is usually too complex to be analytically computed.

4.3 Bayesian Filtering

Let us now consider the probabilistic inference problem in which the state variable set $\mathbf{S}_{1:t}$ is estimated from the observed evidence $\mathbf{e}_{1:\tau}$, finding out the *posterior probability density function* $P\left(\mathbf{S}_{1:t} \mid \mathbf{e}_{1:\tau}\right)$. Let us also focus in one of the posterior pdf marginals, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:\tau}\right)$.

The previous computation is called *smoothing* if $t < \tau$, *filtering* or *monitoring* if $t = \tau$, and *predicting* if $t > \tau$. The general term *estimating* comprises all three processes. This work is focused on filtering, the computation of the belief state $\mathbf{S}_t$ —or, even better, the posterior pdf over the current state $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$— given all evidence up to date $\mathbf{e}_{1:t}$.

In this case, instead of the causal relation given by the likelihood function which assigns probabilities to potential evidences given the state, the filtered pdf allows to make and inference about the state given the evidence.

This pdf can be calculated through *recursive estimation*, that is, computing the new posterior given the previous one and the new evidence [18, 80]:

$$\begin{aligned}
P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) &= P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t-1},\mathbf{e}_t\right) && (4.4)\\
&\propto P\left(\mathbf{e}_t \mid \mathbf{S}_t,\mathbf{e}_{1:t-1}\right)P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}\right) && \text{(Bayes')}\\
&= P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}\right) && \text{(Mark. on ev.)}\\
&= P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)\int P\left(\mathbf{S}_t \mid \mathbf{s}_{t-1},\mathbf{e}_{1:t-1}\right)P\left(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1}\right)d\mathbf{s}_{t-1} && \text{(cond.)}\\
&= \underbrace{\underbrace{P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)}_{\text{likelihood}}}_{\text{updating}}\underbrace{\int \underbrace{P\left(\mathbf{S}_t \mid \mathbf{s}_{t-1}\right)}_{\text{trans. model}}\underbrace{P\left(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1}\right)}_{\text{previous post.}}d\mathbf{s}_{t-1}.}_{\text{prediction}} && \text{(Markov)}
\end{aligned}$$

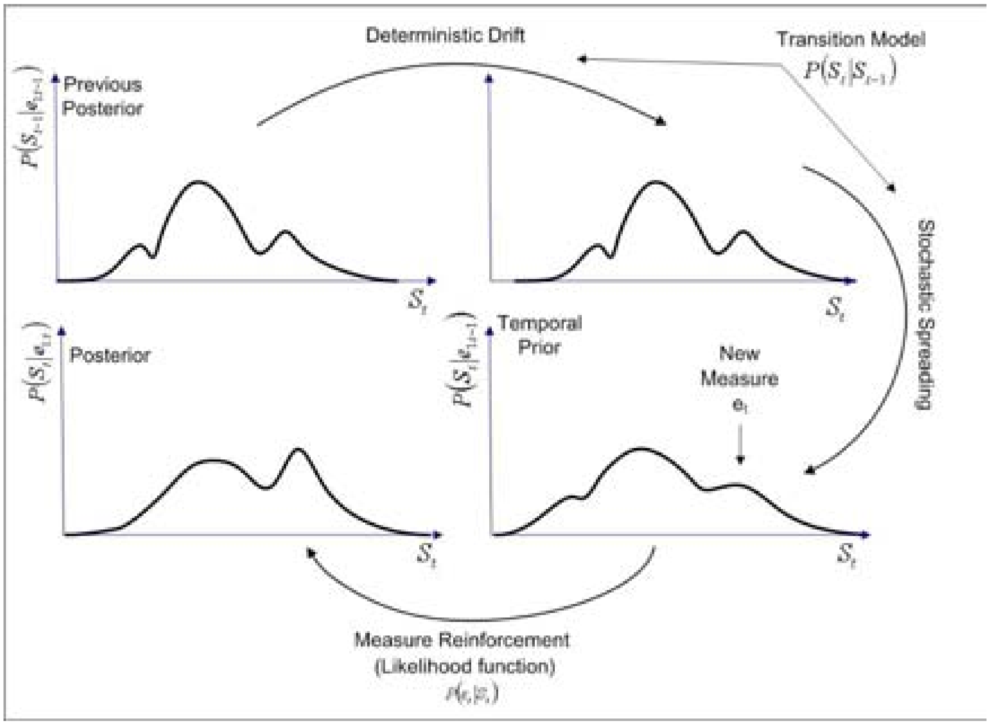


Figure 4.1: Temporal propagation of posterior density functions. A deterministic drift and a stochastic spreading given by the transition model yield the temporal prior. Then, the new posterior is obtained by using the correction given by likelihood function.

The pdf is projected forward according to the transition model, making a prediction. Then, it is updated in agreement with the new evidence, $\mathbf{e}_t$. The prediction term represents the density function after applying the transition model to the previous posterior density function. It leads to the so-called *prior density function*, $P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1})$. It is called prior because it is previous to the likelihood correction.

The temporal propagation of the posterior pdf marginal can be seen as a diffusion–reinforcement process, see Fig. 4.1. The transition model has a deterministic and a stochastic component. The former imposes a drift to the probability density function, while the latter causes the spreading of the pdf that increases the state uncertainty. Subsequently, the likelihood function reinforces the pdf in the vicinity of observations altering the peaks and reducing the uncertainty.

4.4 Monte-Carlo Simulation

Unfortunately, the recursive estimation given above leads to expressions that are impossible to evaluate analytically unless strong assumptions are made. For example, the Kalman Filter is a linear recursive estimator which assumes a linear Gaussian transition model, and a Gaussian likelihood function.

In a more general framework, this problem is overcome by making use of Monte-Carlo methods[4], where N independent-and-identically-distributed (i.i.d.) random samples, $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$, are generated from the posterior pdf, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$.

On the one hand, a simulated probability density function is given by the following expression:

$$\tilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\mathbf{S}_t - \mathbf{s}_t^i\right), \tag{4.5}$$

where $\delta\left(\cdot\right)$ denotes the Dirac delta function.

On the other hand, the posterior expectation is given by:

$$\mu \triangleq \mathbb{E}_{P\left(\mathbf{S}_t | \mathbf{e}_{1:t}\right)}\left[\mathbf{S}_t\right] = \int \mathbf{S}_t P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t, \tag{4.6}$$

and the posterior variance by:

$$\sigma^2 \triangleq \mathbb{E}_{P\left(\mathbf{S}_t | \mathbf{e}_{1:t}\right)}\left[\mathbf{S}_t^2\right] - \mathbb{E}^2_{P\left(\mathbf{S}_t | \mathbf{e}_{1:t}\right)}\left[\mathbf{S}_t\right]. \tag{4.7}$$

Let us now consider the following estimate:

$$\bar{\mathbf{S}}_N = \int \mathbf{S}_t \tilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}_t^i, \tag{4.8}$$

if both posterior expectation and variance are finite, it follows, due to the Central Limit Theorem, that when $N \rightarrow \infty$, $\bar{\mathbf{S}}_N$ has a distribution that is approximately normal, which mean is the posterior expectation μ and its variance is proportional to the posterior variance σ^2:

$$\bar{\mathbf{S}}_N - \mu \dot{\sim} \mathcal{N}\left(0, \frac{\sigma^2}{N}\right). \tag{4.9}$$

Therefore, the posterior expectation $\mathbb{E}_{P\left(\mathbf{S}_t | \mathbf{e}_{1:t}\right)}\left[\mathbf{S}_t\right]$ can be estimated and, in addition, the deviation from the true value follows a normal distribution. Moreover, the higher the number of samples is, the lower the estimate variance will be. These results are also applied for expectations of the form:

$$\mathbb{E}_{P\left(\mathbf{S}_t | \mathbf{e}_{1:t}\right)}\left[\phi\left(\mathbf{S}_t\right)\right] = \int \phi\left(\mathbf{S}_t\right) P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t \tag{4.10}$$

where $\phi\left(\cdot\right)$ is a general function of the state.

However, there are several drawbacks which prevent from using the method as it is presented above. The posterior pdf, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$, is usually complex enough,

[4] Stochastic simulation techniques are referred as Monte-Carlo methods for the Casinos of Monte Carlo, the capital city of gambles. Roulette wheels and dice rolls are simple random number generators.

multivariate, and only known up to a proportionality constant. These problems make impossible to sample directly from it. Thus, alternative solutions are required.

4.5 Sequential Importance Sampling (SIS)

It is possible to avoid the difficulty of sampling directly from the posterior density by sampling from an importance or proposal distribution, $Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)$. As it will be proved, the posterior density function can be approximated arbitrary well by drawing samples from a proposal distribution, and thereby, obtaining approximations of the expectations of interest. Without the lack of generality, results are here obtained for the first raw moment, i.e, the mean:

$$\begin{aligned}
\mu_{P(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} &= \int \mathbf{S}_{0:t} P\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t} && (4.11)\\
&= \int \mathbf{S}_{0:t} \frac{P\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)}{Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)} Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t} && \text{(proposal distr.)}\\
&= \int \mathbf{S}_{0:t} \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{P\left(\mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)} Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t}. && \text{(Bayes)}
\end{aligned}$$

By defining the *unnormalised importance weights* as:

$$\pi_t = \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)}, \qquad (4.12)$$

and conditioning over the evidence probability density function, it follows that:

$$\begin{aligned}
\mu_{P(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} &= \frac{1}{P\left(\mathbf{e}_{1:t}\right)} \int \mathbf{S}_{0:t} \pi_t Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t} && (4.13)\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t}}{\int P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right) d\mathbf{s}_{0:t}} && \text{(conditioning)}\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t}}{\int P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right) \frac{Q(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})}{Q(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} d\mathbf{s}_{0:t}} && \text{(prop. distr.)}\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t}}{\int \pi_t Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) d\mathbf{s}_{0:t}} && \text{(weight def.)}\\
&= \frac{\mathbb{E}_{Q(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\mathbf{S}_{0:t} \pi_t\right]}{\mathbb{E}_{Q(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\pi_t\right]}. && \text{(expect. def.)}
\end{aligned}$$

Both expectations can be approximated by sampling from the proposal distribution. Thus, the posterior distribution mean is thereby approximated using the following estimate:

$$\begin{aligned}\bar{\mathbf{S}}_N &= \frac{\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_{0:t}^i\pi_t^i}{\frac{1}{N}\sum_{i=1}^{N}\pi_t^i} \\ &= \sum_{i=1}^{N}\mathbf{s}_{0:t}^i\overline{\pi}_t^i, \end{aligned} \tag{4.14}$$

where:

$$\overline{\pi}_t^i = \frac{\pi_t^i}{\sum_{j=1}^{N}\pi_t^j}, \tag{4.15}$$

denotes the *normalised importance weights*. The posterior density function can then be approximated in the following way:

$$\begin{aligned} P\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) &\approx \tilde{P}\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) \\ &\approx \sum_{i=1}^{N}\overline{\pi}_t^i\delta\left(\mathbf{S}_{0:t} - \mathbf{s}_{0:t}^i\right), \end{aligned} \tag{4.16}$$

what results from comparing Eq. (4.8) and Eq. (4.14).

Considering a filtering scenario, that is, assuming that current states will not be modified by future observations, the proposal distribution can be decomposed as:

$$\begin{aligned} Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) &= Q\left(\mathbf{S}_{0:t-1}, \mathbf{S}_t \mid \mathbf{e}_{1:t}\right) && (4.17) \\ &= Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t}\right) && \text{(cond. prob.)} \\ &= Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right) && \text{(Mark. on ev.)} \end{aligned}$$

This allows us to obtain a recursive expression for the importance weights:

$$\begin{aligned}
\pi_t &= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)} && (4.18)\\
&= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right)} && \text{(proposal decomp.)}\\
&= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right)} \frac{\pi_{t-1}}{\frac{P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)}{Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right)}} && \text{(weight def.)}\\
&= \pi_{t-1} \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)}\\
&= \pi_{t-1} \frac{P\left(\mathbf{e}_t \mid \mathbf{S}_{1:t}, \mathbf{e}_{1:t-1}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)} && \text{(cond. prob)}\\
&= \pi_{t-1} \frac{P\left(\mathbf{e}_t \mid \mathbf{S}_t\right) P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right)} && \text{(Markov),}
\end{aligned}$$

where

- $P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)$ is the likelihood function;
- $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ is the transition model;
- and, $Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right)$ is the proposal distribution.

A common and easy choice for the proposal distribution —for instance, the one taken in [40]— is:

$$Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) \approx P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right). \tag{4.19}$$

In this case, the importance weights are given by:

$$\pi_t = \pi_{t-1} P\left(\mathbf{e}_t \mid \mathbf{S}_t\right), \tag{4.20}$$

and the normalised importance weights are given by:

$$\overline{\pi}_t^i = \frac{\pi_{t-1}^i p\left(\mathbf{e}_t \mid \mathbf{s}_t^i\right)}{\sum_{j=1}^{N} \pi_{t-1}^j p\left(\mathbf{e}_t \mid \mathbf{s}_t^j\right)}. \tag{4.21}$$

However, this choice has several drawbacks derived from the fact that not incorporating the observations introduces errors in the prediction. Thus, it may be the case that only a few particles have significant weights after being evaluated, specially when the likelihood function is much narrower than the temporal prior.

4.5.1 Degeneracy Problem

The SIS algorithm have an intrinsic problem which prevents from using it as it is. As it is proved in [18], the variance of the importance weights increase over time. This result has devastating consequences on the simulation performance, since the majority of the normalised importance weights tend to zero after few iterations. This samples being numerically insignificant, they are not taken into account in the pdf approximation. This result implies a sample wastage and a poor representation of the posterior distribution.

4.6 Sequential Importance Re-sampling (SIR)

Under this approach, a re-sampling stage is used to prune those particles with negligible importance weights, and multiply those with higher ones. Thus, samples are re-sampled with replacement using the importance weights as probabilities.

This idea is based on the *factored sampling* algorithm [28] designed for stationary pdf's. It works as follows: A posterior representation is given by the Bayes' theorem:

$$P\left(\mathbf{S} \mid \mathbf{e}\right) \propto P\left(\mathbf{e} \mid \mathbf{S}\right) P\left(\mathbf{S}\right), \tag{4.22}$$

but the likelihood function is complex enough to prevent the posterior being evaluated in closed form. Thus, sampling techniques are proposed to generate random variates from a distribution $\tilde{P}\left(\mathbf{s}\right)$ that approximates the posterior $P\left(\mathbf{S} \mid \mathbf{e}\right)$. A sample set of N i.i.d. random samples, $\left\{\hat{\mathbf{s}}^i; i = 1 : N\right\}$, is simulated from the initial prior density function, $P\left(\mathbf{S}\right)$. The algorithm assigns normalised weights $\overline{\pi}^i$ to each sample in the set according to the likelihood function:

$$\overline{\pi}^i = \frac{p\left(\mathbf{e} \mid \widehat{\mathbf{s}}^i\right)}{\sum\limits_{j=1}^{N} p\left(\mathbf{e} \mid \widehat{\mathbf{s}}^j\right)}. \tag{4.23}$$

Subsequently, the samples are selected —or re-sampled— from the sample set with probability $\overline{\pi}^i$. Therefore, the new sample set, $\left\{\mathbf{s}^i; i = 1 : N\right\}$, represents the posterior density function, $P\left(\mathbf{S} \mid \mathbf{e}\right)$, accurately as $N \to \infty$. Obviously, some particles may be chosen several times, especially those with higher weights. Thus, some samples in the new set could be identical. On the other hand, samples with lower weights could be not chosen at all.

This weighted particle representation is shown in Fig. 4.2, where the posterior density function is represented by blobs whose centres are the sample set $\left\{\mathbf{s}^i; i = 1 : N\right\}$ and their area is proportional to the observation value given by the weights $\overline{\pi}^i$.

This idea was introduced by Gordon et al. [27] within a Bayesian filtering framework, thereby leading to *Sequential Importance Re-sampling* (SIR) filters. Here, a posterior probability density function represented by samples is iteratively computed. The pdf undergoes a diffusion-reinforcement process, and the reinforcement stage is

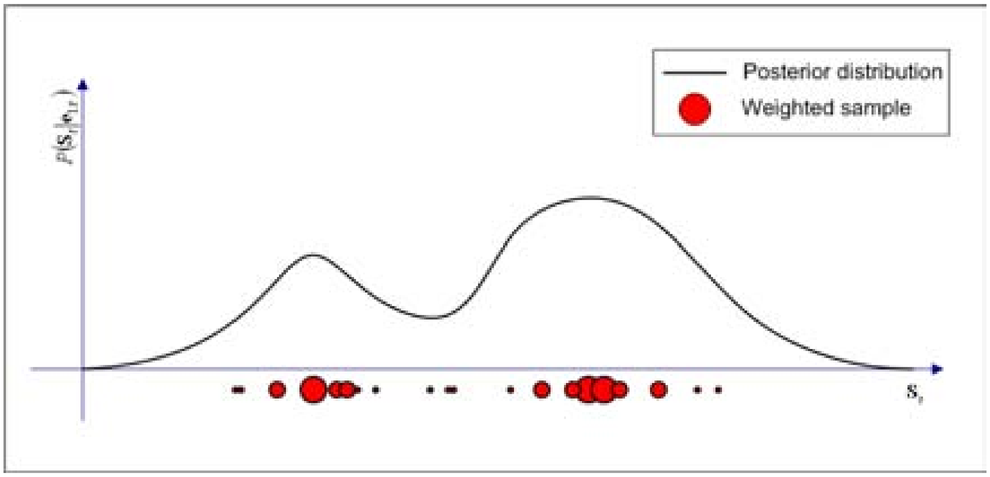


Figure 4.2: Posterior pdf representation as set of weighted particles. See text for details.

followed by a run of the factored sampling algorithm presented above. Thus, the factored sampling is extended by applying it iteratively to successive time-slices.

Subsequently, this techniques were introduced in the Computer Vision field, as well as in other areas such as Artificial Intelligence, or Automatic Control. Therefore, these methods are also variously called: *particle filtering* —after the use of samples or particles as the way of propagating the probability density function— *survival of the fittest* —after the re-sampling stage— *bootstrap filtering*[5], etc. In Computer Vision they are widely used under the name of CONDENSATION, after the paper presented in [38].

4.6.1 The CONDENSATION Algorithm

The CONDENSATION algorithm was presented by Isard and Blake in short form at the European Conference on Computer Vision in 1996 [38]. Later on, it was fully developed in [40]. This intended to track a human contour, which moves in cluttered background, given a raw video signal as data.

CONDENSATION addresses the filtering problem when no assumption about linearity or Gaussianity is made on almost all involved probability density functions. The algorithm is based on Bayesian filters. Therefore, it computes a posterior probability density function $P(\mathbf{S}_t \mid \mathbf{e}_{1:t})$ which undergoes the diffusion-reinforcement process described above. Because of the analytical problems already exposed, it makes use of Monte-Carlo simulation techniques.

[5] The use of the term bootstrap derives from the phrase *"to pull oneself up by one's bootstrap"*, widely thought to be based on one of the eighteenth century *Adventures of Baron Munchausen*, by Rudolph Erich Raspe. In the context of this thesis, it means that the algorithm starts up and recovers by itself: fittest old samples give rise to many new ones.

It follows the aforementioned SIR approach. Thus, the posterior pdf at time $t-1$, $P\left(\mathbf{S}_{t-1} \mid \mathbf{e}_{1:t-1}\right)$, is given by a set of tuples, each of them consisting in one sample and its weight, $\left\{\hat{\mathbf{s}}_{t-1}^i, \overline{\pi}_{t-1}^i; i=1:N\right\}$ or, after applying the factored sampling algorithm, by the re-sampled sample set $\left\{\mathbf{s}_{t-1}^i, \frac{1}{N}; i=1:N\right\}$. In this case, since all particles are evenly weighted, weights are not displayed and the notation is reduced to $\left\{\mathbf{s}_{t-1}^i; i=1:N\right\}$.

Summarising, the four density functions involved in a Bayesian filter are:

1. the initial prior density function, $P\left(\mathbf{S}_0\right)$;
2. the transition model, $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ for $t \geq 1$;
3. the likelihood function, $P\left(\mathbf{E}_t \mid \mathbf{S}_t\right)$ for $t \geq 1$;
4. the posterior state density function, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$ for $t \geq 1$.

The initial prior density function is now the only one supposed to be Gaussian. Therefore, the initial sampling is straightforward. Samples are propagated using the approach described above, that is, by sampling them from the transition model. Thus, there is no need to sample from the previous posterior in subsequently iterations. This fact avoids one of the main problems of the approach based on Monte Carlo Simulation, i.e., sampling from a complex, multivariate and only known up to a proportionality constant posterior pdf.

This algorithm works as follows: each iteration starts with the prediction stage where the temporal prior $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}\right)$ is obtained by applying the transition model $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ to the previous posterior. Computationally, this is done in two steps. In the first place, a deterministic drift is applied to each sample of the previous posterior,$\left\{\mathbf{s}_{t-1}^i; i=1:N\right\}$. Obviously, those samples which were identical will undergo the same drift. Then, the random component, i.e. the diffusion, is applied causing identical samples to split. As a result of this stage, the sample set represents the prior density function at time t, $\left\{\hat{\mathbf{s}}_t^i; i=1:N\right\}$.

The second stage consists in the likelihood correction where the sample weights are calculated according to:

$$\pi_t^i = p\left(\mathbf{e}_t^i \mid \hat{\mathbf{s}}_t^i\right). \tag{4.24}$$

It is worth to notice that there is no need to recursively propagate the weights —as done in Eq. (4.21)— since all previous weights are even and equal to $\frac{1}{N}$ after the re-sampling stage. Once all samples have been propagated and measured, the final stage applies the factored sampling to carry out the re-sampling phase. Thus, weights are normalised:

$$\overline{\pi}_t^i = \frac{\pi_t^i}{\sum\limits_{i=j}^{N} \pi_t^j}, \tag{4.25}$$

where $\overline{\pi}_t^i$ denotes the i-th sample normalised weight at time t.

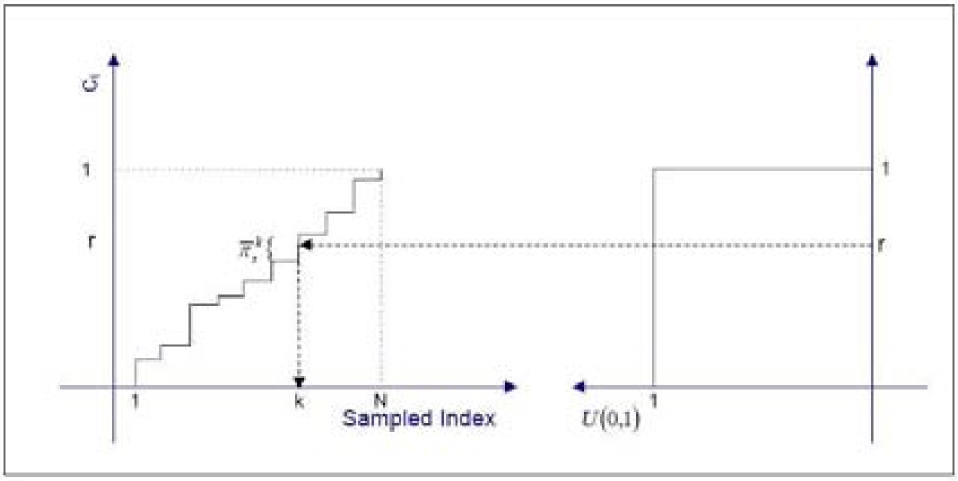


Figure 4.3: Cumulative distribution.

```
Algorithm 1 Re-sampling stage.
  • For each sample $\mathbf{s}_t^i$:
      1. a random number is generated from a Uniform distribution, $r \in [0, 1]$.
      2. the smallest k index for which $c_t^k \geq r$ is found.
      3. the corresponding sample is selected, $\mathbf{s}_t^i = \hat{\mathbf{s}}_t^k$.
  • end for $i$
```

Sampling from the discrete set $\left\{\hat{\mathbf{s}}_t^i; i = 1 : N\right\}$ with probabilities $\overline{\pi}_t^i$ can be accomplished by sampling from a discrete uniform distribution, projecting the index onto the sample cumulative distribution range and then onto the distribution domain [18], see Fig. 4.3.

The cumulative probability distribution is constructed according to:

$$\begin{aligned} c_t^0 &= 0, \\ c_t^i &= c_t^{i-1} + \overline{\pi}_t^i, \quad i = 1 : N. \end{aligned} \tag{4.26}$$

Then, the new sample set, $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$ is calculated by generating a random number, and selecting the sample whose corresponding cumulative probability exceed this number. This process is summarised in Algorithm 1.

Finally, the sample set represents the posterior pdf at time t, $P\left(\mathbf{s}_t, \mathbf{e}_{1:t}\right)$. The

Algorithm 2 Condensation.

```
Propagation
  • for each sample in the set $\left\{\mathbf{s}_{t-1}^{i}; i = 1 : N\right\}$ do
      1. predict the sample values $\hat{\mathbf{s}}_t^i$ using the transition model $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$;
      2. measure the sample weights $\pi_t^i$, Eq. (4.24);
  • end for $i$
State Estimation
  • Estimate the state according to Eq. (4.27);
Re-sampling
  • Normalise the weights, Eq. (4.25);
  • Compute the cumulative probabilities as in Eq.(4.26);
  • Call the algorithm in Algorithm 1.
```

sample set size N is kept constant over time for all iterations. The expected value at time t can be approximated as:

$$\mathbb{E}_{P(\mathbf{S}_t|e_{1:t})}\left[\mathbf{S}_t\right] \approx \sum_{i=1}^{N} \varpi_t^i \hat{\mathbf{s}}_t^i \tag{4.27}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_t^i. \tag{4.28}$$

It is interesting to remark that the accuracy of any estimate —such as the mean and covariance— of the posterior distribution can only decrease as a result of the re-sampling stage. Thus, if these quantities are to be used or displayed, then these should be computed prior to re-sampling, as in Eq. (4.27), instead of using the posterior expression in Eq. (4.28).

The algorithm is graphically depicted in Fig. 4.4, and summed up in Algorithm 2.

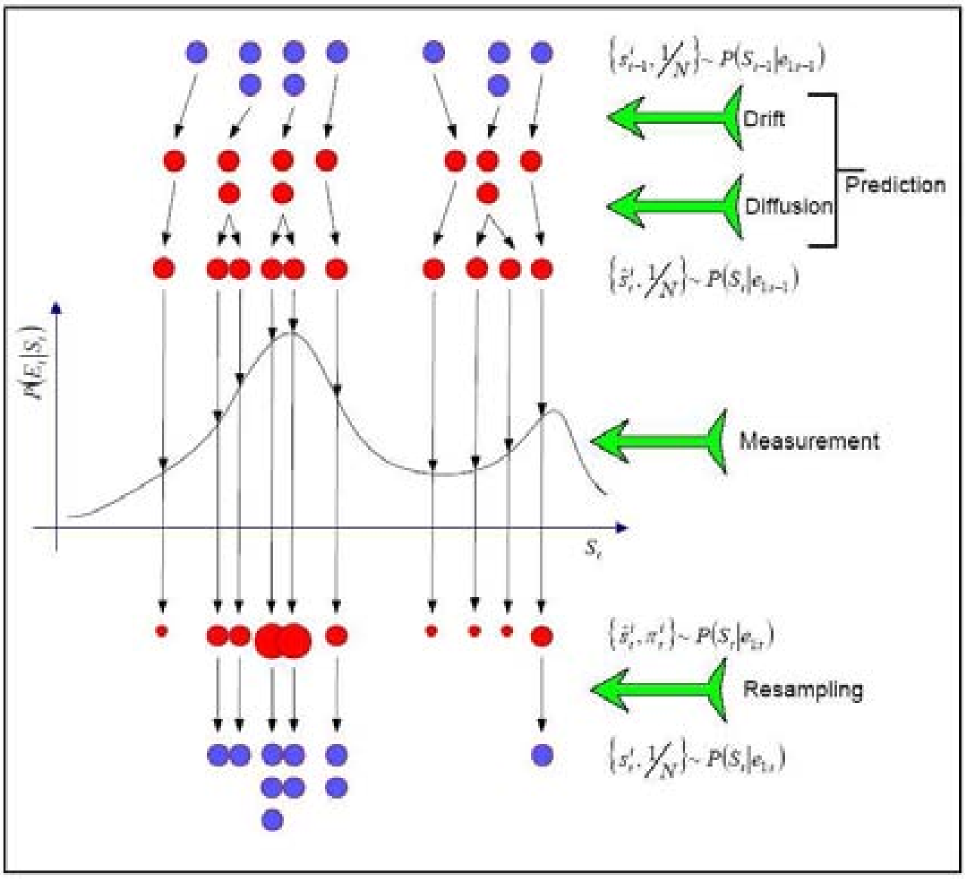

$\{s^i_{t-1}, 1/N\} \sim P(S_{t-1}|e_{1:t-1})$
Drift
Diffusion
Prediction
$\{\tilde{s}^i_t, 1/N\} \sim P(S_t|e_{1:t-1})$
$P(E_t|S_t)$
Measurement
S_t
$\{\tilde{s}^i_t, \pi^i_t\} \sim P(S_t|e_{1:t})$
Resampling
$\{s^i_t, 1/N\} \sim P(S_t|e_{1:t})$

Figure 4.4: Condensation algorithm: a graphical representation of one iteration. See text for details.

4.6.2 The Drawbacks of the Condensation Algorithm

Condensation has certainly been widely applied between 1999 and 2003. According to Cite-Seer[6], it has a peak of over 35 citations in 2001 and 271 hits within the Cite-Seer database. It has been considered fast and efficient due to its two main advantages:

1. first of all, it can represent multi-modal density functions. This fact allows us to consider multiple hypotheses, which is essential in scenes where background clutter or other moving objects[7] could mimic the target. Thus, it is possible to propagate multiple hypotheses which are pruned or reinforced in each iteration depending on their likelihood.

[6] http://citeseer.ist.psu.edu/

[7] Which does not mean that several targets can be tracked at the same time using the algorithm as it is.

2. The second advantage is that, maintaining the sample set size fixed, it was supposed to be able to run with bounded computational resources in near real time[8].

Isard and Blake proved in [40] the asymptotic correctness of the algorithm by showing that the sample set representation of the posterior density function has weak and uniform convergence as $N \to \infty$. Thus, it is stated that each sample at time t of the sample set $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$ is drawn from a probability density function $\widetilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$ such that $\widetilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) \to P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$, where $\to$ denotes weak, uniform convergence[9].

However, they already warned that the convergence was proved for $N \to \infty$ *given a fixed t.* Therefore, the sampled representation approximates the true distribution with a desired accuracy but only for a fixed number of frames T. Nothing is said about the limit $T \to \infty$. Thus, *at later times larger values of N may be required.*

They also stated that *there is no information about how large N should be* for a requested precision and, therefore, it is heuristically determined. These and other undesirable Condensation side-effects were thoroughly discussed by King and Forsyth [52]. They are briefly presented in the next paragraphs.

One of the main drawbacks of the re-sampling algorithms is a phenomenon called *sampling impoverishment*. Let us consider that the samples are spread around several *modes*[10]. King and Forsyth demonstrated that, with probability one —what is called an *almost sure* event[11]—, all samples will end up in one of those modes. Moreover, the probability that one mode absorbs all samples is proportional to the number of samples that started in it. Therefore, spurious modes have a non-zero probability of usurping all samples, causing the true mode to be lost.

Although sampling impoverishment is well studied and proved in [52], it can also be informally explained as a result of what is called *genetic drift*: consider a finite population and one particular gene. The frequency of the gene will not be exactly reproduced in the offspring due to sampling errors. This sampling error is propagated over time. The initial frequency is lost because there is not any kind of genetic memory. Eventually, this random process leads to a population where this gene is either lost or is present in every individual. In both cases, no further changes are

[8] However, as will be shown later, having a fixed sample set size has several drawbacks. Further, the number of samples required to ensured acceptable performances in high dimensional spaces prevent from a real-time use in most applications. An on-line sample-set size adaptation was explore was Fox [22] by evaluating the approximation error using the Kullback-Leibler distance; this was kept bounded by modifying the sample set size.

[9] **Weak convergence:** for every Q defined in a probability space, $\left\langle \widetilde{P}\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right), Q\right\rangle \to \left\langle P\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right), Q\right\rangle$ where $\langle\rangle$ denotes the inner product.

Uniform convergence: for every $\varepsilon > 0$, there exists a natural number N such that for all $\mathbf{s}_t$ and all $n > N$, $\left|\widetilde{p}\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right) - p\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right)\right| < \varepsilon$.

[10] The term mode here refers to each local maximum of the distribution.

[11] There is a subtle difference between an event being *sure* and *almost sure*. On the one hand, a sure event will always happen, and no other event can ever happen. On the other, if an event is almost sure, other event are allowed to occur, but they happen almost never. Thus, for instance, infinite sequences of events, or a continuum of outcomes, allow events with zero-probability to occur —like hitting with a dart a particular point.

possible. Thus, one mode has disappeared and it cannot be recovered. The Markov chain that modelled the process has reached an *absorbing state*, and its distribution is known as a *stationary distribution* which means that $P(\mathbf{S}_{t+1}) = P(\mathbf{S}_t)$.

Condensation uses factored sampling. This process involves a loss of information. The probability for one sample of being selected is given by its weight. Consider now that several samples could be identical and similar samples form modes that can be far enough one from the other. The probability of propagating one mode is proportional to the number of samples that constitute it. Sample impoverishment means that all but one of these modes could disappear, and this fact has a non-negligible probability of happening in finite time.

Considering a real-time tracking application —whose frame rate can be set for instance at 30 frames per second, which means 30 generations per second— it is obvious that many modes could disappear in less than seconds. How many seconds will be needed is only a matter of how many samples are used.

Moreover, lost modes have a very low probability of being recovered. The diffusion process could preserve diversity, as mutation does in genetics. However, the distance between modes is usually bigger than the diffusion. One sample will need several iterations in order to move from one mode to another. But the likelihood in the region between modes is small, thereby making such a journey highly improbable.

Summarising, *there is a non-negligible probability of losing modes, a low probability of recovering them, and the remaining modes could be all spurious.*

There is also another interesting fact, albeit undesirable as well. Isolated populations, starting with identical gene frequency, can end up in different absorbing states. Thus, variation within populations is turned into variations between populations. Returning to the tracking problem, this fact means that different runs of the algorithm lead to different results. Therefore, *computed expectations may have high variance.* However, *computed expectations within the same algorithm run have low variance* making the tracker look stable.

A yet another remarkable phenomenon is caused by the tendency of Condensation towards clustering samples. *Even when the likelihood function gives no information at all*, i.e, there is nothing to track in the scene, *samples become quickly concentrated.* It strongly looks as if the tracker is following something, when actually it isn't. Of course, the peaks tracked differ from run to run.

Finally, Condensation was designed to keep multiple hypotheses but only for a single target. Thus, multiple-target tracking was not feasible. Further extensions and variations from other authors [74, 57] usually lead to the so-called *curse of dimensionality*[12].

King and Forsyth proposed two approaches to tackle sampling impoverishment. In the first place, they suggested using fewer re-sampling steps. Obviously, a well constrained dynamic model would be required, what is usually not feasible. The second suggestion implies generating new samples occasionally. This suggestions has

[12] This is a term coined by Richard Bellman in 1961 to refer to the problem caused by the exponential increase of an hyper-volume as a function of space dimensionality: adding extra dimensions causes an exponential growth of the number of required samples to densely populate the space.

been followed by Varona et al. in [91], and within the importance-sampling framework, by Isard and Blake [41].

4.7 An Approach to MTT by Particle Filtering

In this section, an proposal based on particle filters is developed in order to perform Multiple-Target Tracking. The approach was initially inspired in the *iTrack* algorithm —within the SIR framework— implemented by Varona in his PhD thesis [90]. Subsequently, the focus has been placed in coping with two main difficulties:

1. inherent drawbacks of SIR methods;
2. and, scenario-dependent problems.

On the one hand, serious computational problems arose due to the inability of managing particle sets which must be big enough to populate adequately the search space, thereby being able of representing arbitrary distributions. Thus, particles should be wisely steered and re-sampled, so as to reduce the number of required particles. Issues such as sample impoverishment, and the curse of dimensionality must be tackle in a principled way.

On the other hand, robust tracking requires to deal with expected difficulties, such as background clutter and target occlusion. The non-rigid nature of the targets, along with changing illumination conditions, make model updating unavoidable. However, model drift should be prevented at any cost to ensure tracking viability.

4.7.1 State Modelling

A first-order dynamic model in image coordinates is used to model the motion of the central point of a bounding box. This bounding box is considered the region within the scene which is thought to enclose the target.

Thus, the target's motion is characterised by its position at time t, $\mathbf{x}_t = (x_t, y_t)^T$, and its speed, $\mathbf{u}_t = (u_t, v_t)^T$. This dynamic model involves the assumption of constant speed —acceleration will be given by Gaussian noise—- which can be more o less realistic depending on the target's dynamics and the frame rate. It usually holds in trajectory-analysis applications at current common frame rates of 25-30 fps.

The aspect model is given by a bounding box and an appearance matrix. The former, denoted by $\mathbf{w}_t = (w_t, h_t)^T$, defines a rectangle whose size is given by its width, w_t, and its height, h_t. The latter, denoted by $\mathbf{A}_t$, stores the pixel intensity values within the bounding box. An indicator of the expected likelihood value is given by λ_t. This stores expected matching, taking into account that differences will be found due to sensor noise, changes in illumination, shape deformations, etc.

The occlusion status is inferred and store in ρ_t. This is a binary variable which points out whether the target is the nearer one in a group to the camera.

Finally, a label l associates a specific appearance model to the corresponding samples, allowing multiple-target tracking. Therefore, the $l-$target's state is defined as $\mathbf{s}_t^l = \left(\mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{w}_t^l, \mathbf{A}_t^l, \rho_t^l, \lambda_t^l\right)^T$.

4.7.2 Transition Model

Several independence relationships are assumed in order to determine the transition model. It is considered that both aspect and dynamic models are independent, that the position only depends on the previous position and speed, the speed on the previous one, and so does the bounding box on and the appearance. Therefore, the transition model can be split:

$$\begin{aligned} P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right) &= P\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_t, \mathbf{A}_t \mid \mathbf{X}_{t-1}, \mathbf{U}_{t-1}, \mathbf{W}_{t-1}, \mathbf{A}_{t-1}\right) \qquad (4.29)\\ &= P\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \mathbf{U}_{t-1}\right) P\left(\mathbf{U}_t \mid \mathbf{U}_{t-1}\right) P\left(\mathbf{W}_t \mid \mathbf{W}_{t-1}\right) P\left(\mathbf{A}_t \mid \mathbf{A}_{t-1}\right). \end{aligned}$$

Given the constant speed assumption, the dynamic model can be defined according to:

$$\begin{aligned} P\left(\mathbf{X}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}\right) &= \mathcal{N}\left(\mathbf{X}_t; \mathbf{x}_{t-1} + \mathbf{u}_{t-1}\Delta_t, \mathbf{\Sigma}_\mathbf{x}\right), \qquad (4.30)\\ P\left(\mathbf{U}_t \mid \mathbf{u}_{t-1}\right) &= \mathcal{N}\left(\mathbf{U}_t; \mathbf{u}_{t-1}, \mathbf{\Sigma}_\mathbf{u}\right). \qquad (4.31) \end{aligned}$$

Thus, the position state variable $\mathbf{X}_t$ evolves according to a linear Gaussian whose mean is a linear expression of its parents and the variance is fixed and heuristically determined. Δ_t is the sampling period. Time is considered discrete and measured in frames. Thus, Δ_t equals 1. Position is also discrete and measured in pixels. On the other hand, the speed state variable $\mathbf{U}_t$ evolves according to a Gaussian whose mean is its parent and the variance is again heuristically fixed according to the expected target acceleration. These two covariance matrices are denoted by $\mathbf{\Sigma}_\mathbf{x}$ and $\mathbf{\Sigma}_\mathbf{u}$.

In order to implement the aspect model, it is assumed that the shape evolves smoothly, and the appearance is fixed between consecutive frames according to:

$$\begin{aligned} P\left(\mathbf{W}_t \mid \mathbf{w}_{t-1}\right) &= \mathcal{N}\left(\mathbf{W}_t; \mathbf{w}_{t-1}, \mathbf{\Sigma}_\mathbf{w}\right), \qquad (4.32)\\ P\left(\mathbf{A}_t \mid \mathbf{A}_{t-1}\right) &= \delta\left(\mathbf{A}_t - \mathbf{A}_{t-1}\right). \qquad (4.33) \end{aligned}$$

where $\mathbf{\Sigma}_\mathbf{w}$ denotes the size covariance matrix.

Although the appearance is considered to be fixed when propagating the state, it will eventually be updated once the posterior expectation is computed.

Therefore, the position, speed, and size of each sample are predicted according to:

$$\begin{aligned} \hat{\mathbf{x}}_t^{i,l} &= \mathbf{x}_{t-1}^{i,l} + \mathbf{u}_{t-1}^{i,l}\Delta_t + \xi_\mathbf{x}^i,\\ \hat{\mathbf{u}}_t^{i,l} &= \mathbf{u}_{t-1}^{l} + \xi_\mathbf{u}^i,\\ \hat{\mathbf{w}}_t^{i,l} &= \mathbf{w}_{t-1}^{i,l} + \xi_\mathbf{w}^i, \qquad (4.34) \end{aligned}$$

where the random vectors $\xi_\mathbf{x}^i, \xi_\mathbf{u}^i, \xi_\mathbf{w}^i$, sampled from WAGN processes, provide the system with a diversity of hypotheses.

Sample likelihoods depend on sample position and size, but not on their speeds. Thus, if speeds were propagated considering the previous speed, they would be in quasi open loop[13]. Thus, their values could become completely different from the true values within a few frames, and an important proportion of samples would be wasted. In order to avoid this phenomenon, the estimated target speed $\mathbf{u}_{t-1}^{l}$ at time $t-1$ is fed back into the prediction of $\hat{\mathbf{x}}_t^{i,l}$.

After the initialisation, no sample is generated using detection, since it would mask tracking misbehaviours. Thus, just tracking performances are tested by means of propagating hypotheses and weighting them according to evidence. Clearly, by incorporating detection, the general performance will be enhanced, providing the system with error-recovery capabilities.

4.7.3 Template-based Likelihood Function

In a visual tracking context, the likelihood function gives the probability density function of image features given the state. The intensity is chosen as image feature. Features are considered pixel-oriented. Hence, the appearance is given by a matrix whose elements are the pixels' intensity values.

Let $\mathbf{I}_t$ be a matrix whose elements are the scene pixel intensity values at time t. Thus, evidence $\mathbf{e}_t$ is given by the input image sequence $\mathbf{I}_t$. Given the predicted position $\mathbf{X}_t$ and bounding-box size $\mathbf{W}_t$, the corresponding image sub-region is denoted by $\mathbf{I}_t^p$. The model appearance matrix must be scaled according to the sample size. Let $\mathbf{A}^s$ be the model scaled matrix. Thus, assuming that the likelihood function is independent of the speed component, it can be expressed as:

$$\begin{aligned} P\left(\mathbf{I}_t \mid \mathbf{S}_t\right) &= P\left(\mathbf{I}_t \mid \mathbf{X}_t, \mathbf{W}_t, \mathbf{A}_t\right) \\ &= P\left(\mathbf{I}_t^p \mid \mathbf{A}_t^s\right), \end{aligned} \tag{4.35}$$

and, once assumed constant appearance between frames and *White Additive Gaussian Noise*, the likelihood function can be defined as a similarity measure which averages the likelihood of all pixels within the bounding box[14]:

$$\begin{aligned} P\left(\mathbf{I}_t^p \mid \mathbf{A}_t^s\right) &= \frac{1}{M} \sum_{a,b \in \mathbf{A_t^s}} P\left(\mathbf{I}_t^p\left(a,b\right) \mid \mathbf{A}_t^s\left(a,b\right)\right) \\ &= \frac{1}{M} \sum_{a,b \in \mathbf{A_t^s}} \mathcal{N}\left(\mathbf{I}_t^p\left(a,b\right); \mathbf{A}_t^s\left(a,b\right), \sigma_n^2\right), \end{aligned} \tag{4.36}$$

[13] There would still be a weak relation, since speeds are used to predict positions, and position errors can be measured, but a considerable delay would be introduced, as it will be shown in the experimental results.

[14] This expression does not pretend to follow a probabilistic derivation. The likelihood function is usually defined in terms of a *distance*, and this distance is here computed from the likelihood of each pixel within the bounding box.

where M is the number of pixels of the appearance model, (a, b) defines a pixel position in the appearance matrix and σ_n^2 is the camera noise variance, which randomly influences the pixels' intensity values.

4.7.4 Weight Normalisation

In a multiple-target tracking scenario, those targets whose samples exhibit lower likelihood are more likely to be lost, since the probability of propagating one mode is proportional to the cumulative weights of its samples. In order to avoid one target absorbing other target samples, genetic drift must be prevented. Thus, a memory term, which takes into account the number of targets being tracked, is included. Weights are normalised according to:

$$\overline{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum\limits_{i=1,j=l}^{N} \pi_t^{i,j}} \frac{1}{L}, \tag{4.37}$$

where L is the number of tracked targets. Each weight is normalised according to the total weight of the target's samples. Thus, all targets have the same probability of being propagated, since the addition of the weights of each target samples sums $\frac{1}{L}$. This allows multiple-target tracking using a single PF framework, despite the differences between their likelihoods and the genetic drift phenomenon.

4.7.5 State Estimation

The l-target estimates are computed according to:

$$\begin{aligned}
\mathbf{x}_t^l &= (1 - \alpha_{\mathbf{x}}) \left(\mathbf{x}_{t-1}^l + \mathbf{u}_{t-1}^l \Delta_t\right) + \alpha_{\mathbf{x}} \left(L \sum_{i=1}^{N} \overline{\pi}_t^{i,l} \hat{\mathbf{x}}_t^{i,l} \right), \\
\mathbf{u}_t^l &= (1 - \alpha_{\mathbf{u}}) \, \mathbf{u}_{t-1}^l + \alpha_{\mathbf{u}} \left(\frac{\mathbf{x}_t^l - \mathbf{x}_{t-1}^l}{\Delta_t} \right), \\
\mathbf{w}_t^l &= (1 - \alpha_{\mathbf{w}}) \, \mathbf{w}_{t-1}^l + \alpha_{\mathbf{w}} \left(L \sum_{i=1}^{N} \overline{\pi}_t^{i,l} \widehat{\mathbf{w}}_t^{i,l} \right),
\end{aligned} \tag{4.38}$$

where $\alpha_{\mathbf{x}}, \alpha_{\mathbf{u}}, \alpha_{\mathbf{w}} \in [0, 1]$ denote the adaptation rates. Target speeds are not estimated according to sample speeds and their weights, since significant errors would be introduced: samples are chosen only because of sample weights, which do not directly depend on the current speed. This fact could imply a significant amount of jitter and many samples would be wasted. Therefore, target speeds are computed from successive position estimates. Further, both position and speed estimates are enhanced by regularising them according to their histories.

The target appearance must also be updated. However, this is a sensitive task which may lead to the well-known *model drift* phenomenon. Thus, models are then only updated when two conditions hold:

- the target is not occluded;
- and, the likelihood of the estimated target's state suggests that the estimate is sufficiently reliable.

In this case, they are updated using an adaptive filter:

$$\mathbf{A}_t^l = (1 - \alpha_{\mathbf{A}})\, \mathbf{A}_{t-1}^{l,s} + \alpha_{\mathbf{A}} \mathbf{I}_t^l, \tag{4.39}$$

where $\alpha_{\mathbf{A}} \in [0, 1]$ is the learning rate, and $\mathbf{I}_t^l$ is the image sub-region cropped given the target new estimate position and size $\mathbf{x}_t^l, \mathbf{w}_t^l$.

In order to determine when the estimate is reliable, the likelihood of the current estimate is computed, $p\left(\mathbf{e}_t \mid \mathbf{s}_t^l\right)$. The appearance is then updated when this value is higher than an indicator of the expected likelihood value, calculated following an adaptive rule:

$$\lambda_t^l = (1 - \alpha_l)\, \lambda_{t-1}^l + \alpha_l p\left(\mathbf{e}_t \mid \mathbf{s}_t^l\right). \tag{4.40}$$

4.7.6 Occlusion handling

Although the appearance model is not updated during occlusions, these still constitute a main cause of catastrophic failures. Partial occlusions may cause inaccurate size updating, according to the area that can be seen. In case of complete occlusions, sample likelihoods are meaningless, and the re-sampling phase randomly propagate them, quickly losing the target.

Hence, proper handling of occlusions is crucial. The state binary variable ρ_t^l tracks the occlusion status. Occlusions are predicted according to the learnt dynamics. When the predicted occlusion is significant, and the target likelihood is lower than the expected one given by λ_t^l, the target state changes into occluded. Then, the following changes are introduced:

- neither the size, nor the velocity or the likelihood-expectation indicator are updated; the position is just propagated
- those samples belonging to the occluded target are not re-sampled. As a result, samples are spread around the target because of the uncertainty predictions terms. The other targets' samples are re-sampled, but are not assigned to the occluded target, since otherwise this one would monopolise the whole sample set.

When the occlusion is no longer predicted, or a sample likelihood exceeds the value previous to the occlusion, ρ_t^l turns into zero, which immediately implies pruning those samples with lower weights. Furthermore, all estimates are again updated.

4.7.7 Extension of the Tracking Algorithm

Bounding-boxes and templates can hardly model the shape and appearance of non-rigid targets. The target region representation is changed into an ellipse in order to reduce the number of background pixels included in the model. Now, the motion of the central point of an elliptical region is modelled using first-order dynamics in image coordinates.

Further, the target appearance is represented by means of colour histograms. Histograms are broadly used to represented human appearance, since they are claimed to be less sensitive than colour templates to rotations in depth, the camera point of view, non-rigid targets, and partial occlusions. By using colour as image feature instead of intensity, a better target disambiguation can be achieved.

Thus, the $l-$model is given by:

$$\overline{\mathbf{p}}^l = \left\{p_k^l; k = 1 : K\right\}, \tag{4.41}$$

where K is the number of bins, and the probability of each feature is:

$$p_k^l = C^l \sum_{a=1}^{M} \delta\left(b\left(\mathbf{x}_a\right) - k\right), \tag{4.42}$$

where C^l is a normalisation constant required to ensure that $\sum_{k=1}^{K} p_k^l = 1$, δ the Kronecker delta, $\{\mathbf{x}_a; a = 1 : M\}$ the pixel locations, and $b\left(\mathbf{x}_a\right)$ a function that associates the given pixel to its corresponding histogram bin.

The l-labelled target's state is then defined as $\mathbf{s}_t^l = \left(\mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{w}_t^l, \overline{\mathbf{p}}^l, \rho_t^l, \lambda_t^l\right)^T$, where components are the ellipse position, velocity, both axes, the appearance model, the occlusion status, and the expected target likelihood.

4.7.7.1 A Colour-based Likelihood function

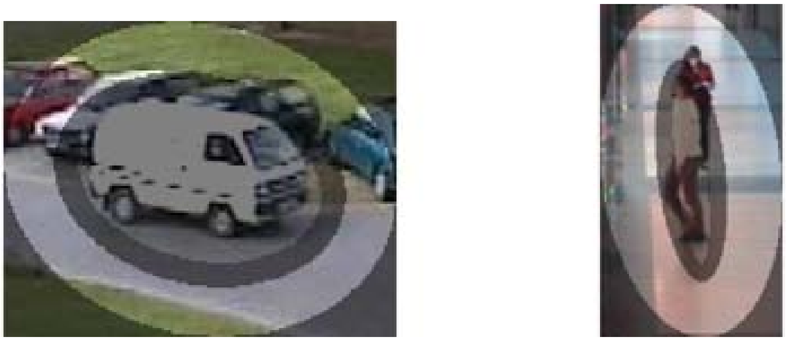
The target distribution at the predicted position $\hat{\mathbf{x}}_t^{i,l}$ and ellipse size $\hat{\mathbf{w}}_t^{i,l}$, is given by $\mathbf{p}_i^l$, which is calculated in the same way as the model. The similarity between two histograms can be computed using the following metric [14, 67]:

$$d_B = \sqrt{1 - \rho\left(\mathbf{p}, \overline{\mathbf{p}}^l\right)}, \tag{4.43}$$

where

$$\rho\left(\mathbf{p}, \overline{\mathbf{p}}^l\right) = \sum_{k=1}^{K} \sqrt{p_k \overline{p}_k^l}, \tag{4.44}$$

is known as the *Bhattacharyya coefficient.* Therefore, similar histograms have a high



(a) (b)

Figure 4.5: Examples of a centre-surround model with safety margin. (a) Tracked van from a traffic-monitoring sequence. (b) Tracked person from an indoor surveillance application in a shopping centre. Regions from centre to border: target estimation, safety margin, surrounding background, and non-local background.

Bhattacharyya coefficient, which should correspond to high sample weights. The computed metric can be mapped using a Gaussian distribution [67], and samples are thus weighted according to:

$$\pi_t^{i,l} = p\left(\mathbf{e}_t \mid \hat{\mathbf{s}}_t^{i,l}\right) = \mathcal{N}\left(d_B; \mu, \sigma^2\right). \tag{4.45}$$

So far no background information has been used. However, tracking success depends on how distinguishable the target is from a local environment. Thus, foreground features present also in its surroundings should be less important for target localisation. Here, an approach similar to [14] is adopted by using a *centre-surround* model to compute the local background histogram $\mathbf{q}^l$ according to the outer region which encloses the target, see Fig. 4.5.

The local background region is given by an ellipse which encloses the tracked one by defining two margins of dimension $\kappa_s * \max(h, w)$. The potential incorporation of own target pixels, specially if the target shape cannot be fairly represented by an ellipse is minimised by taking into account just the outer region to build the local background histogram. κ_s is usually equal to 0.1 for the inner margin and 0.3 for the outer one. Hence, the background histogram is used to compute a weight for each bin:

$$\omega_k^l = \left\{ \min\left(\frac{q_k^{l*}}{q_k^l}\right); k = 1 : K \right\}, \tag{4.46}$$

where q_k^{l*} is the minimum non-zero value. Thus, these weights are then applied to the target histogram to diminish the importance of those bins which represent the local background. Hence, the resulting Bhattacharyya coefficient is

Algorithm 3 MTT particle filtering

```
PROPAGATION
  • for $i = 1$ to $N$ do
      1. predict the sample values $\hat{\mathbf{s}}_t^{i,l}$ using the transition model in Eq. (4.34)
      2. measure the sample weights $\pi_t^i$ according to Eq. (4.45)
  • end for $i$
UPDATING
  • normalise the weights as in Eq. (4.37)
  • predict occlusion percentage according to target's dynamics models
  • for $l = 1$ to $L$ do
      1. evaluate occlusions according to target collision and likelihoods
      2. estimate the target state:
          (a) if target is occluded then set adaptation rates $\alpha_{\mathbf{x}}, \alpha_{\mathbf{u}}$ to zero
          (b) estimate target position and speed according to Eq. 4.38
          (c) if the target estimate is reliable
               i. update target's size
              ii. update the appearance models following Eqs. (4.38),(4.48)
             iii. update $\lambda_t^l$ as in Eq.(4.40)
  • end for $l$
RE-SAMPLING
  • Build the cumulative distribution as in Eq.(4.26)
  • for $i = 1$ to $N$ do
    if target $l$ is occluded then keep the sample: $\mathbf{s}_t^{i,l} = \hat{\mathbf{s}}_t^{i,l}$.
    else proceed with re-sampling as in Algorithm 1
  • end for i
```

$$\rho_w\left(\mathbf{p},\overline{\mathbf{p}}^l\right) = \sum_{k=1}^{K}\omega_k^l\sqrt{p_k\overline{p}_k^l}. \tag{4.47}$$

Finally, in the state-estimation stage, Eq.(4.39) is changed accordingly:

$$\overline{\mathbf{p}}_t^l = \left(1-\alpha_{\mathbf{q}}\right)\overline{\mathbf{p}}_{t-1}^l + \alpha_{\mathbf{q}}\mathbf{p}_t^l, \tag{4.48}$$

where $\alpha_{\mathbf{q}} \in [0,1]$ is the learning rate which weights the most recent values versus the historic ones. The complete algorithm is summarised in Algorithm 3.

4.8 Discussion

With this work we have attempted to take a step towards solving the numerous difficulties which appear in MTT applications by means of particle filtering[15].

Dynamics updating is modified by feeding back the estimated speed into the prediction stage. The target's speed is estimated from successive position estimates. Both position and speed estimates are now regularised. Thus, sample wastage is significantly reduced. In addition, trajectory jitter is considerably attenuated.

Different likelihood function have been explored in order to properly evaluate samples associated to targets which present a high appearance variability. Finally, the approach relies on the Bhattacharyya coefficient between colour histograms to perform this task.

Model updating is carried out with special care, in order to overcome the model drift phenomenon. A multiple-target tracking scenario causes several problems, including sampling impoverishment and mutual occlusions. These issues are tackled by redefining the weight normalisation, and predicting and handling occlusions. The proposed sample-weight normalisation avoids losing any of the targets due to the lack of samples.

Although significant advances have been obtained —see chapter on experimental results— the approach is far from being suitable to perform multiple target tracking in cluttered environments under uncontrolled conditions in long sequences. This is due to multiple facts:

- Monte-Carlo methods are usually not able to densely populate a high-dimension spaces. Estimations are performed from a limited number of samples. This results in poor state approximations when dealing with multi-modal pdf's.
- Top-down approaches require extremely constrained models, which is not feasible in generic applications. Errors in the estimation are propagated, thereby causing model drift.

[15] Experimental results obtained using the presented approach in both synthetic and real scenarios are shown in the corresponding chapter, see section 6.2 on page 147.

- An independent observation process from prediction is required to cope with estimation errors with a finite number of samples. This entails the necessity a bottom-up process.
- Likelihood functions are usually not discriminative enough.

Taking all these issues in mind, a novel approach which simultaneously takes advantage of both bottom-up and top-down paradigms is developed in next chapter. As stated by the English Franciscan Friar William of Ockham in the 14th century, "entia non sunt multiplicanda praeter necessitatem". This principle[16] suggests to select the theory that introduces the fewest assumptions and postulates the fewest entities, which is of course not the case of PF's in uncontrolled environments.

The hierarchical architecture presented in the following intends to make use of all available sources of information, while keeping the assumptions to a minimum, and avoiding the use of constrained models. Two trackers —motion-based and appearance-based— are embedded as modules in each pathway, i.e. bottom-up and top-down, respectively. In the proposed approach, these are implemented as a Kalman Filter and a Mean-shift tracker. Both functionalities can be carried out by a particle filter like the one above described —in case some conditions hold, like the existence of constrained models. Nevertheless, given the aforementioned reasons, the practical implementation has been left to the above stated filters.

4.9 Resum

Amb aquest treball hem intentat avançar cap a la resolució de les nombroses dificultats que apareixen en aplicacions de MTT per mitjà de resultats de filtratge. L'actualització de les dinàmiques es realitza alimentant-se de la velocitat aproximada a l'escenari durant la predicció. La velocitat dels objectes es calcula des de successius prediccions de posició. Es normalitzen ara les prediccions tant de posició com de velocitat. Així, el desaprofitament de les mostres utilitzades es redueix significativament. A més a més, les desviacions en la predicció de la trajectòria també s'atenuen considerablement.

Han estat explorats funcions de versemblança diferents per pròpiament avaluar que les mostres s'associaven a els objectes que presenten una variabilitat d'aspecte molt alta. Finalment, l'enfocament depèn del coeficient Bhattacharyya entre histogrames de color utilitzats per aquesta tasca.

L'actualització dels models es fa amb una cura especial, per vèncer el fenomen de la deriva del model. El seguiment de múltiples objectes provoca uns quants problemes, incloent-hi l'empobriment de mostreig i les oclusions mútues. Aquests cassos es tracten redefinint la normalització de pes, i pronosticant i manejant oclusions. La normalització proposada dels pesos de les mostres evita perdre qualsevol dels objectes a causa de la manca de mostres.

Encara que s'han obtingut avenços significatius – vegi el capítol sobre resultats experimentals – l'enfocament és lluny de ser l'adequat per realitzar el seguiment de

[16] It is usually referred as the 'Ockham´s razor'

múltiples objectes en ambients oberts sota condicions incontrolades i en seqüències llargues. Això és a causa de múltiples fets:

- Els mètodes de Monte-Carlo no són normalment capaços de poblar densament espais de dimensionalitat alta. Les prediccions es realitzen des d'un nombre limitat de mostres. Això ocasiona aproximacions pobres en tractar amb pdf's multimodals.
- Les aproximacions de Dalt-a-Baix exigeixen models extremadament restrictius, la qual cosa no és factible en aplicacions genèriques. Els errors en l'estimació es propaguen, provocant així una deriva dels models.
- Es requereix un procés d'observació independent de la predicció per afrontar errors en les prediccions amb un nombre finit de mostres. Això suposa la necessitat un procés de Baix-a-Dalt.
- Les Funcions de Versemblança normalment no són prou discriminatives.

Prenent tots aquests problemes, es desenvolupa en el pròxim capítol una nova aproximació que prengui avantatge simultàniament de paradigmes tant de Baix-a-dalt com de Dalt-a-baix. Com va manifestar el Franciscà anglès William d'Ockham al segle 14è, "entia non sunt multiplicanda praeter necessitatem". Aquest principi suggereix seleccionar la teoria que introdueixi les menors suposicions que pressuposi les menors entitats, que naturalment no és el cas dels filtres de partícules en entorns no controlats.

L'arquitectura jeràrquica presentada tot seguit pretén fer ús de totes les fonts d'informació disponibles, mantenint les suposicions a un mínim, i evitant l'ús de models massa restrictius. S'inclouen dos algorismes de seguiment —basat en moviment i en l'aparença– en cada sentit, i.e. de Baix-a-Dalt i de Dalt-a-Baix, respectivament. En l'aproximació proposada a continuació, aquests s'implementen com un Filtre Kalman i un algorisme de Mean-shift. Les dues funcionalitats poden ser assolides per un filtre de partícules com el descrit anteriorment —in el cas de que algunes condicions es mantinguem, com l'existència de models restrictius. No obstant això, atès les susdites raons, es presenta una aplicació pràctica amb els filtres anteriorment mencionats.

Chapter 5

A Principled Hierarchical Architecture to Multiple-Target Tracking

Non-supervised MTT involves such an inherent complexity that leads to propose a structured framework to accomplish such a task. First of all, reliable target segmentation is critical in every tracking system in order to achieve an accurate feature extraction without considering any prior knowledge about potential targets. This is even more crucial in dynamic open scenes. However, complex interacting agents who move through cluttered environments require high-level analysis.

5.1 Approach Outline

Our proposal combines in a principled architecture both bottom-up and top-down approaches. This is implemented as a modular and hierarchically-organised system. The resulting architecture is based on a set of co-operating modules which are distributed through three levels. Each level is defined according to the different tasks to be performed: Target Detection, Low-Level Tracking (LLT), and High-Level Tracking (HLT). A sketch of this system[1] is shown in Fig. 5.1.

The different modules take part in both bottom-up and top-down processes. On the one hand, the bottom-up process provides the system with capabilities for initialisation, error-recovering and simultaneous modelling and tracking. On the other hand, the top-down one builds the models according to a high-level event interpretation, and allows the system to switch between the two operation modes implemented: Motion-Based Tracking and Appearance-Based Tracking.

These concurrent processes are allowed due to the fact that in the proposed architecture the tracking task is split into two levels: the lower one, which is based on

[1] The notation used through this chapter is summed up and explained in detail in Appendix B. It may slightly differ from the one used in the previous chapter due to practical reasons derive from dealing with multiple approaches and algorithms.

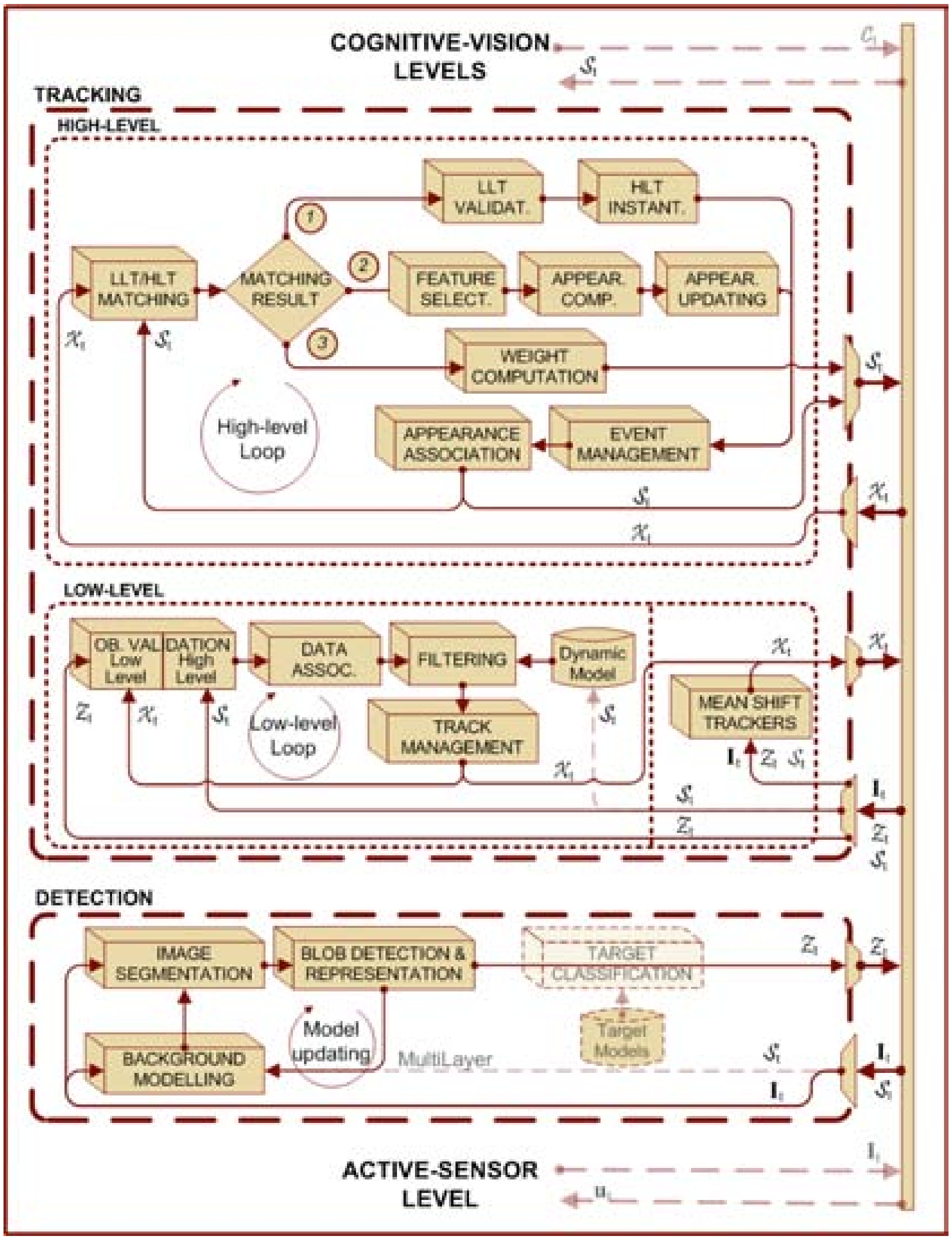


Figure 5.1: Tracking architecture. $\mathbf{I}_t$ represents the current frame; the observation, LLT and HLT data structures are denoted by $\mathcal{Z}_t$, $\mathcal{X}_t$ and $\mathcal{S}_t$ respectively; $\mathbf{u}_t$ represents a vector of potential system control signals, while $\mathcal{C}_t$ refers to high-level information. Matching results are explained in the text. Ongoing and future-planned modules are shown in transparent dash lines.

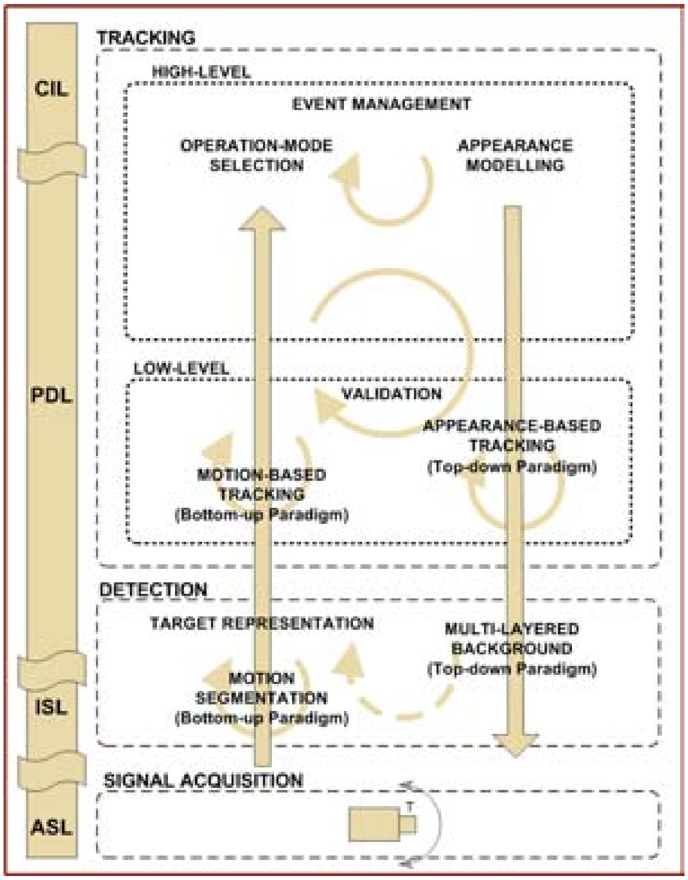

TRACKING
HIGH-LEVEL
EVENT MANAGEMENT
OPERATION-MODE SELECTION
APPEARANCE MODELLING
LOW-LEVEL
VALIDATION
APPEARANCE-BASED TRACKING (Top-down Paradigm)
MOTION-BASED TRACKING (Bottom-up Paradigm)
DETECTION
TARGET REPRESENTATION
MULTI-LAYERED BACKGROUND (Top-down Paradigm)
MOTION SEGMENTATION (Bottom-up Paradigm)
SIGNAL ACQUISITION
CIL
PDL
ISL
ASL
T

Figure 5.2: Relations between the HSE framework and the proposed tracking architecture. See text for details.

short-term blob trackers, and a higher one, based on long-term target trackers. The latter has a crucial importance: it automatically builds and tunes multiple appearance colour models, manages the events in which the target is involved, and selects the most appropriate tracking approach according to these. Therefore, the system can react to what is taking place, and switch accordingly to a most convenient operation mode [60].

It is interesting to remark that the tracking architecture presented in Fig. 5.1 is a part of the complex HSE framework shown in Fig. 3.1 on page 50. Thus, segmentation tasks within the Detection Level correspond to ISL; target detection and classification, as well as LLT, and appearance representation within the HLT belong to PDL; and event management, operation-mode selection, and other HLT tasks are assimilated to CIL[2].

[2] However, the HSE framework aims to be a conceptual abstraction of system functionalities, while the proposed architecture implements a real tracking system. Therefore, func-

Further, cognitive levels consequently require the global position, shape and appearance of all targets within the scene: this information is fed forward by the tracking system. In addition, this system can benefit in the future from the cognitive processes performed at the higher levels of the HSE framework. Finally, the scene could be recorded using active cameras. In this case, an image mosaic would be built, and the entire process would be transparent for the architecture here presented. The relations between both HSE framework and the implemented tracking architecture is shown in Fig 5.2.

The current system design considers no use of a-priori knowledge about either the scene or the targets, based on extensive off-line training or learning periods. The aim is to implement a system general enough to be independent of a particular scenario, and which can directly be used. However, the expected future use of this high-level information can do nothing but enhance the current system performances[3].

In the following, a comparison is presented between the proposal and a natural paradigm. Subsequently, each level shown in Fig. 5.1 is depicted in detail, as well as the relations among the different cooperating modules. Thus, it is the architecture itself what is considered as the main contribution: it introduces in the system the necessary synergies which permit to tackle such a inherently complex problem. However, contributions include not only the architecture itself, but also the development and improvement of the different modules. Notwithstanding, there the main focus is placed on the high-level tracking algorithms. Hence, contributions have been presented on diverse modules, levels ans tasks —such as on segmentation [37], low-level tracking [26], high-level tracking [79], and event management [78].

5.2 A Solution Inspired in a Natural Paradigm

The proposed architecture can be seen as a biological-inspired solution in many ways. In a natural paradigm, visual-stimuli processing can be divided into two categories [44, 68]: on the one hand, bottom-up or pre-attentive processes carry out raw data processing without high-level, a-priori learnt information —this is usually done quickly and apparently effortless in the whole visual field; on the other hand, top-down or attentive processes perform goal-oriented tasks by making use of context and domain knowledge. Nevertheless, these two kind of processes are strongly linked, and they occur simultaneously in a closed loop [68]. In this way, the latters are applied to solve those cases in which the formers fail, and to tune them in order to focus the attention on the object of interest. Further, the pre-attentive stage of vision performs the processing for different visual cues, such as motion or colour. This is done in a parallel and independent way. Subsequently, these results are fused in the attentive stage.

tionality correspondences have a degree of fuzziness, as represented in Fig 5.2.

[3] The system can be particularised to a defined scenario by introducing known context constraints in the different algorithm implemented in each module. Further, learning methods can be considered to tune the algorithm parameters. However, it is worth to say the current sensitivity to these is low enough to allow keep them fixed during the hundreds of processed frames of each of the multiple considered sequences in many different scenarios.

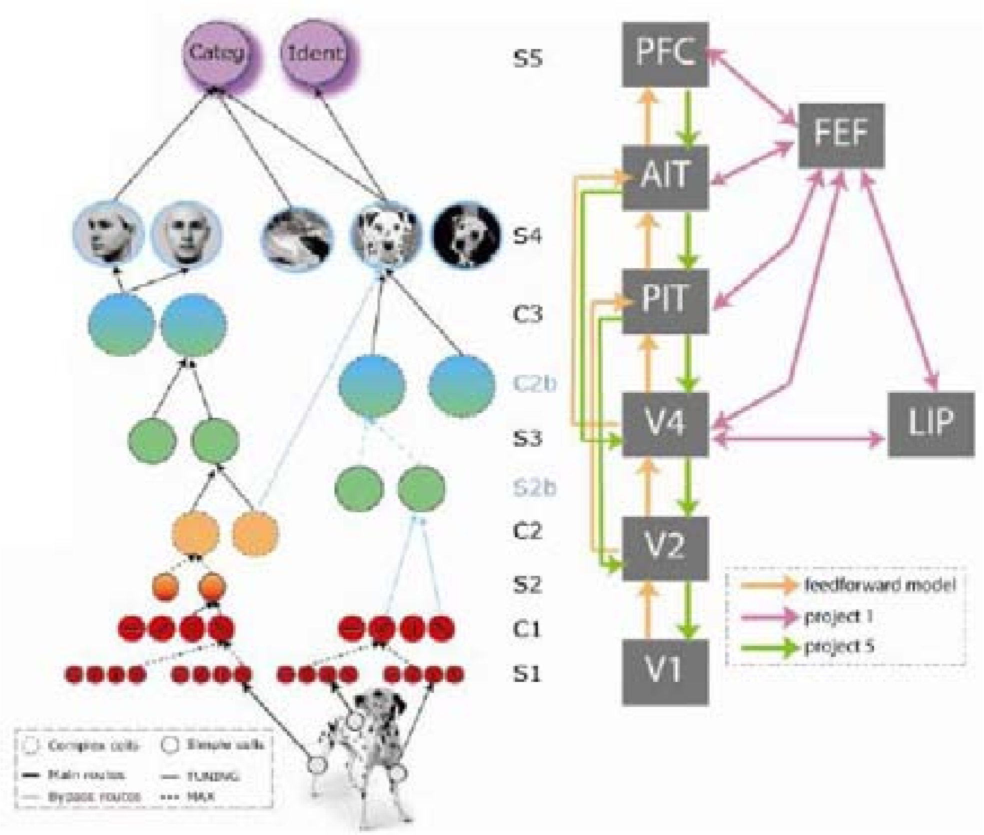


Figure 5.3: Biological foundation of the tracking architecture. (Figure from Scene Understanding Symposium, MIT, T. Poggio, 2007)

Hence, our proposed architecture follows this natural paradigm in several senses. As it has been stated, the approach is based on a two-level tracking system fed by a detection level. Thus, it combines the two stages of visual perception. Our pre-attentive stage provides a coarse localisation, while the attentive one performs an accurate tracking of those objects of interest, by means of a further analysis and hypothesis confirmation. This biological basis can also be found in other Computer Vision applications, such as medical imaging retrieval, and face tracking approaches [64, 20]. Further, the attention/back-projection information flow[4] is currently a challenging new line of research [82], see Fig. 5.3.

As intermediate objects, low-level trackers are created at a initial level of abstraction, by processing segmented image data. This step provides several advantages: (i) segmentation errors due to noise, camouflage, or the inclusion of shadows and reflections are reduced, thereby limiting potential spurious structural changes; (ii) the

[4] http://suns.mit.edu/SUnS07Slides/Poggio_SUnS07.pdf, T. Poggio, Scene Understanding Symposium, MIT, 2007.

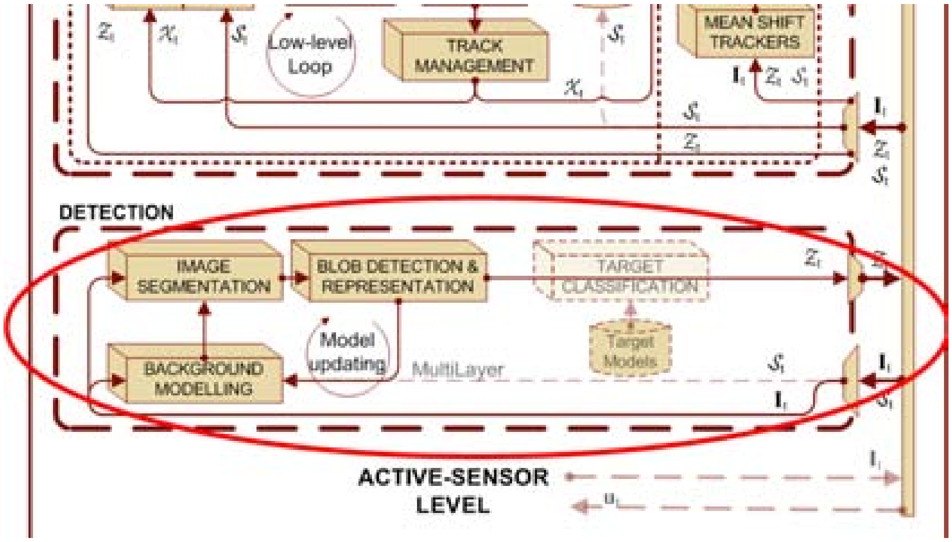


Figure 5.4: Detection level.

low-level target representation can be handled by high-level entities, thereby reducing the sensory gap between images and high-level abstractions; and (iii) the computational complexity is cut down by using a compact representation, which also removes confusing elements.

Thus, low-level motion trackers perform a rough tracking where detailed models are avoided. No appearance information is used, and events are not analysed. After this first stage of pre-attentive processing, and once the low-level trackers reach enough confidence, the system performs selective examinations of the tracked objects that draw its attention. Hence, high-level trackers build accurate appearance colour-based models, and analyse the events in which they take part in. This information is then used to act on the lower trackers. Therefore, the output $\mathcal{S}_t$ from high-level attentive tracking algorithms is fed back to the lower levels, tuning pre-attentive cues, and yielding a closed-loop system.

Further, the two implemented operation modes follow also the natural paradigm of first-order and second-order motion perception [66]. While the former is performed by detecting luminance changes in a particular point of the retina, and correlating it with a delayed change at a neighbouring point, the latter depends on moving blobs defined in terms of contrast —difference in the color and brightness with the surroundings— or texture. Thus, an analogy can also be found between each tracking tier with human peripheral or nocturnal vision in contrast to central colour vision.

Finally, an structural biological foundation can be seen in the presented architecture: each level has an inner feed-back loop, but the different levels are part of several outer loops. Thus, like in a vertebrate nervous system, decisions can be locally taken, or given by a higher level [29]. See Appendix E for more information about how a Natural Vision System works, and about the hierarchical system of response

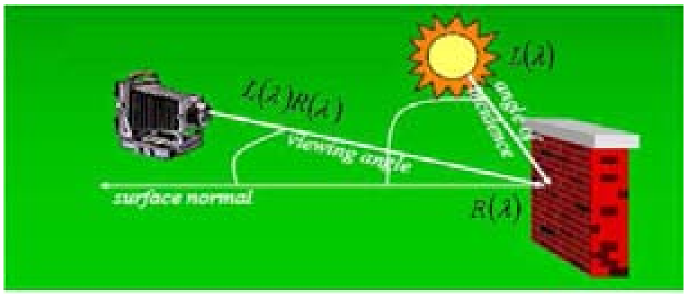


Figure 5.5: Sensor response. The sensor response depends on the illuminant wavelength, and on the object reflectance, apart from the sensor sensitivity. (Figure modified from CS410 notes, Draper, 2006).

generation.

5.3 Detection Level

The first level performs target detection from motion segmentation, see Fig. 5.4. The segmentation task is accomplished following a statistical background-subtraction approach which uses either colour or intensity cues according to sensor response s^c [37]. The segmented image is subsequently filtered, a connected component analysis is performed, and extracted blobs are parametrically represented.

The sensor response s^c —for Lambertain, or perfect matte surfaces— depends on three components: the illuminant spectral power distribution $L(\lambda)$, the object reflectance distribution $R(\lambda)$, and the sensor sensitivity $S^c(\lambda)$:

$$s^c = \int_\lambda L(\lambda)\, R(\lambda)\, S^c(\lambda)\, d\lambda, \tag{5.1}$$

where λ denotes the wavelength, and $c \in \{R, G, B\}$ the colour channel, see Fig. 5.5. Therefore, changes in the illumination —in both brightness and chrominance components— modify the sensor response, see Fig. 5.6. The object reflectance may considerably depend on the both the incident-light angle, and the viewing angle. It also may present strong specular components, that have no information about the object colour. Finally, it depends on the sensor sensitivity, see Fig. 5.7. In addition, the sensor dynamic range must be taken into account. This is defined as the ratio between the maximum possible signal versus the noise signal in dark. Thus, very low or very high brightness distort the observed response. Consequently, these effects should be considered as a source of potential errors during both background modelling and image segmentation.

Fig. 5.8 shows a case analysis of the potential segmentation casuistry using the combination of two background models. These consists on a colour-based one which separates both chrominance and brightness component; and and intensity one computed for those pixels beyond the sensor range.

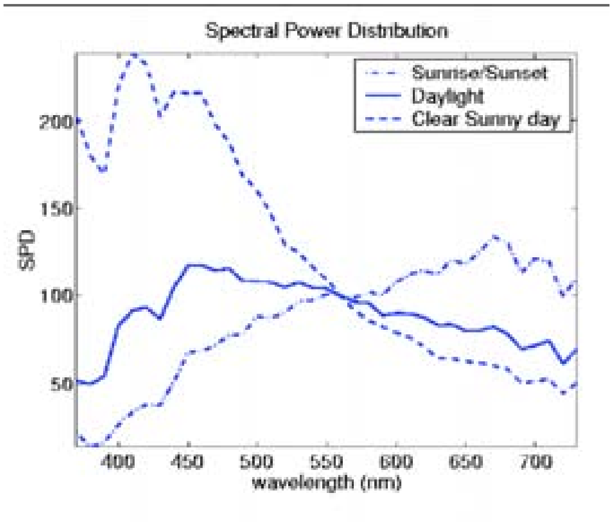


Figure 5.6: Illuminant Spectral Power Distribution. The illuminant SPD may vary, thereby affecting the observed colour. (Figure modified from CS320 notes, Jepson, 2005).

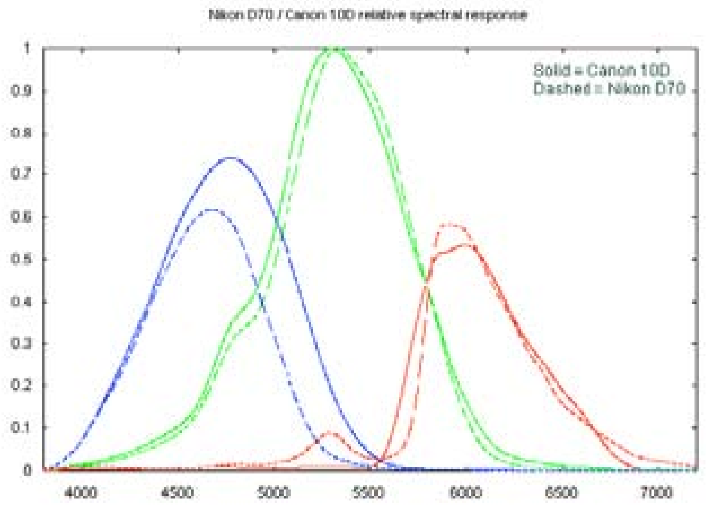
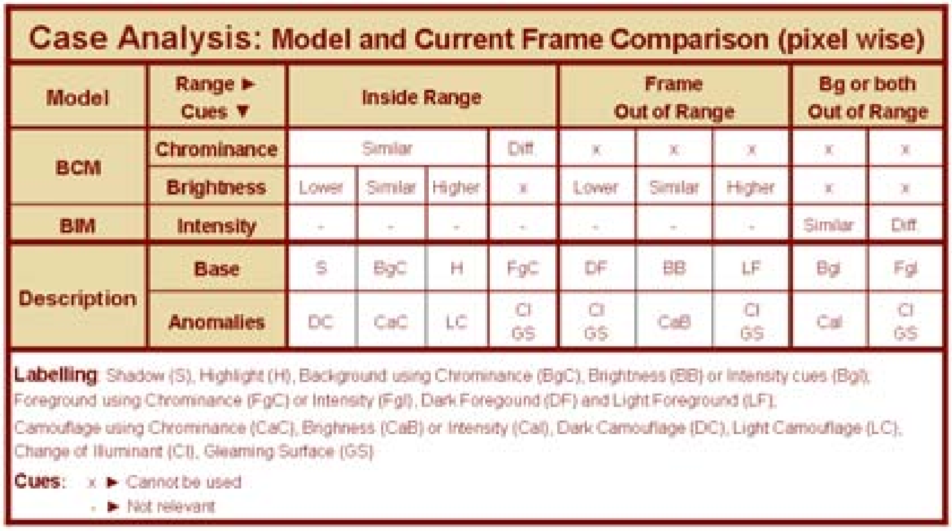


Figure 5.7: Sensor sensitivity. Different sensors present a different response to the same stimulus. (Figure from http://astrosurf.com/build/70v10d/eval_htm).

The *colour base case* is the correct operation of the chrominance model. Thus, a pixel is considered as foreground when it differs in chrominance with the model. Changes in illumination conditions —such as those cause by shadows— are supposed



Case Analysis: Model and Current Frame Comparison (pixel wise)

Model	Range ► Cues ▼	Inside Range				Frame Out of Range			Bg or both Out of Range	
BCM	Chrominance	Similar			Diff	x	x	x	x	x
	Brightness	Lower	Similar	Higher	x	Lower	Similar	Higher	x	x
BIM	Intensity	-	-	-	-	-	-	-	Similar	Diff
Description	Base	S	BgC	H	FgC	DF	BB	LF	BgI	FgI
	Anomalies	DC	CaC	LC	CI GS	CI GS	CaB	CI GS	CaI	CI GS

Labelling: Shadow (S), Highlight (H), Background using Chrominance (BgC), Brightness (BB) or Intensity cues (BgI), Foreground using Chrominance (FgC) or Intensity (FgI), Dark Foregound (DF) and Light Foreground (LF). Camouflage using Chrominance (CaC), Brighness (CaB) or Intensity (CaI), Dark Camouflage (DC), Light Camouflage (LC), Change of Illuminant (CI), Gleaming Surface (GS)

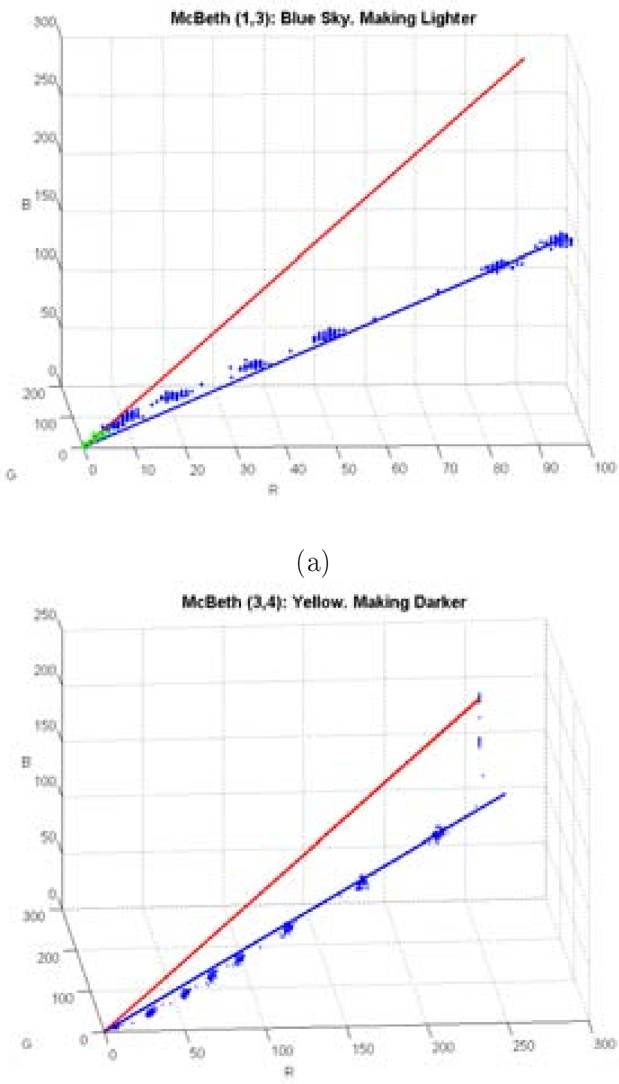
Cues: x ► Cannot be used
- ► Not relevant

Figure 5.8: Segmentation casuistry based on chrominance and brightness. See text for details.

to entail just variations in the observed brightness, but not in the chrominance.

Secondly, very dark pixels do not have enough brightness to reliably compute the chrominance, since they are beyond the sensor dynamic range. A similar problem appears with very light pixels, which have at least one channel component saturated. A series of experiments with a Macbeth board were designed to explore these phenomena, see Fig. 5.9. The experiments show that a wrong background model may be built depending on the illumination conditions. Thus, a Macbeth board is illuminated with a constant light source. Then, the diaphragm is modified in a series of time steps, thereby changing the received luminance. The red line denotes the modelled chrominance line, whereas the blue one corresponds to the actual one. The background was modelled during 50 frames, and the corresponding pixel values are drawn in green. Then, 650 more frames are acquired while changing the aperture. These pixel values are drawn in blue. These cases are addressed using the intensity model, being the *intensity base case.*

Hence, the base case solves some of the segmentation problems, such as shadows and highlights —independently of their being local or global, sudden or gradual— as long as the illuminant has a plain spectral power distribution. The *anomalies* are problems that may appear, since they cannot be disambiguated using colour and intensity cues. The next anomalies should be taken into account: firstly, foreground



(a)

(b)

Figure 5.9: Experiments on a Macbeth board to test the sensor dynamic range. (a) This corresponds to a blue checker which is not observed with enough light during the modelling process. (b) In this case, the chrominance of a yellow checker is modelled while some of the channels are saturated. Consequently, there are important deviations between the inferred and actual chrominance in both cases.

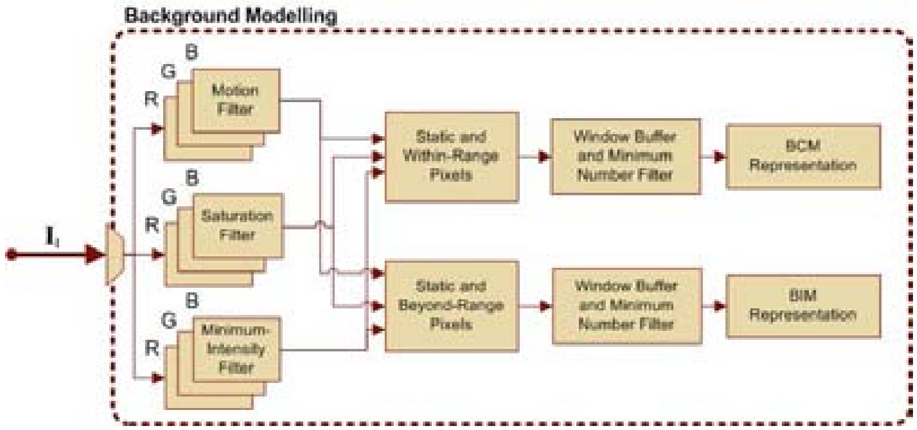

Background Modelling
I_t
R
G
B
Motion Filter
Saturation Filter
Minimum-Intensity Filter
Static and Within-Range Pixels
Static and Beyond-Range Pixels
Window Buffer and Minimum Number Filter
Window Buffer and Minimum Number Filter
BCM Representation
BIM Representation

Figure 5.10: Background modelling approach. See text for details.

pixels with the same chrominance as the background model are not segmented, and are considered camouflaged. Secondly, this is also the case of pixels with lower and higher brightness, that cannot be distinguished from shadows and highlights, respectively. Finally, local and global changes in the illuminant chrominance, as well as gleaming surfaces, cause false-positive segmentations.

This casuistry is here used to implement an image-segmentation algorithm which addresses the base case by combining different cues.

5.3.1 Background Model

The background is modelled on a pixel-wise basis [86, 30, 61, 81], which provides the necessary model accuracy. This is carried out by using a window of T frames. A motion filter is used to remove moving pixels during the modelling stage:

$$\left| I^c_{a,t} - \widetilde{I}^c_a \right| < \max\left(\kappa_m \sigma^c_a, \epsilon\right), \tag{5.2}$$

where $\widetilde{I}^c_a$ is the median value of channel $c \in \{R, G, B\}$ of pixel a during the T frames, σ^c_a their standard deviation, $\kappa_m$the factor that sets the confidence region, and ϵ a small positive quantity. This process is iterated until convergence. Then, just those pixels with a representative number of valid values in the T frames are taken into account for background modelling.

Two cues, colour and intensity, are considered in order to build the background model. On the one hand, those pixels whose RGB values are beyond the linear range of the sensor are also filtered before building the Background Colour Model (BCM). On the other hand, those pixels values which are beyond the sensor dynamic range are used to build the Background Intensity Model (BIM). A sketch of the Background-Modelling Module is shown in Fig 5.10.

The BCM is computed according to the representation shown in Fig. 5.11: first, the RGB mean $\boldsymbol{\mu}_a$ and standard deviation $\boldsymbol{\sigma}_a$ of every image pixel a during the time

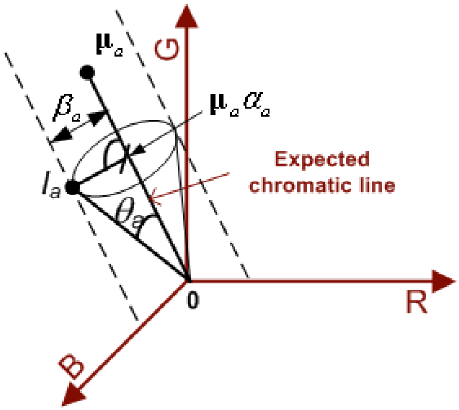


Figure 5.11: Colour-model representation. $\boldsymbol{\mu}_a$ represents the expected RGB colour value for a pixel a, while $\mathbf{I}_a$ is the current pixel value. The line $\overline{\mathbf{0}\boldsymbol{\mu}_a}$ shows the expected chromatic line —all colours along this line have the same chrominance, but different brightness. α_a and β_a give the current brightness and chrominance distortion, respectively.

period $t = [1 : T]$ are computed:

$$\boldsymbol{\mu}_a = \left(\mu_a^R, \mu_a^G, \mu_a^B\right), \tag{5.3}$$

$$\boldsymbol{\sigma}_a = \left(\sigma_a^R, \sigma_a^G, \sigma_a^B\right). \tag{5.4}$$

Once each RGB component is normalised by their respective standard deviation $\sigma_a^c, c \in \{R, G, B\}$, two distortion measures are established: the brightness distortion, $\alpha_{a,t}$, and the chrominance distortion, $\beta_{a,t}$. The brightness distortion can be computed by minimising the distance between the current pixel value $\mathbf{I}_{a,t}$ and the chromatic line $\overline{\mathbf{0}\boldsymbol{\mu}_a}$. This distance is, in fact, the chromatic distortion. Thus, the brightness distortions is given by:

$$\alpha_{a,t} = \frac{\frac{I_{a,t}^R \mu_a^R}{(\sigma_a^R)^2} + \frac{I_{a,t}^G \mu_a^G}{(\sigma_a^G)^2} + \frac{I_{a,t}^B \mu_a^B}{(\sigma_a^B)^2}}{\left(\frac{\mu_a^R}{\sigma_a^R}\right)^2 + \left(\frac{\mu_a^G}{\sigma_a^G}\right)^2 + \left(\frac{\mu_a^B}{\sigma_a^B}\right)^2}, \tag{5.5}$$

and the chromatic one by:

$$\beta_{a,t} = \sqrt{\sum_{c=R,G,B} \left(\frac{I_{a,t}^c - \alpha_{a,t}\mu_a^c}{\sigma_a^R}\right)^2}. \tag{5.6}$$

Finally, the Root Mean Square over time of both distortions for each pixel is computed: $\bar{\alpha}_a$ and $\bar{\beta}_a$, respectively:

$$\bar{\alpha}_a = RMS\left(\alpha_{a,t} - 1\right) = \sqrt{\frac{\sum_{t=0}^{T}\left(\alpha_{a,t} - 1\right)^2}{T}}, \tag{5.7}$$

$$\bar{\beta}_a = RMS\left(\beta_{a,t}\right) = \sqrt{\frac{\sum_{t=0}^{T}\left(\beta_{a,t}\right)^2}{T}}, \tag{5.8}$$

where 1 is subtracted to $\alpha_{a,t}$, so that the brightness distortion is now distributed around zero: positive values mean brighter pixels, whereas negative ones mean darker pixels, with regard to the learnt values. These values are used as normalising factors so that a single threshold can be set for the whole image. This 4-tuple $\left(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a, \bar{\alpha}_a, \bar{\beta}_a\right)$ constitutes the pixel colour background model.

Unfortunately, chrominance cues cannot be used for those foreground pixels beyond the sensor dynamic range. For this cases, the brightness of the BCM is used as segmenting cue.

The BIM consist on a 2-tuple given by the mean pixel intensity, μ_a^I and its standard deviation σ_a^I. It is computed for those non-in-motion pixels which have a representative number of values beyond sensor dynamic range.

5.3.1.1 Automatic Threshold Selection

The model is completed by an automatic threshold computation for a given detection rate. A new frame is presented and normalised distortions are calculated for each pixel:

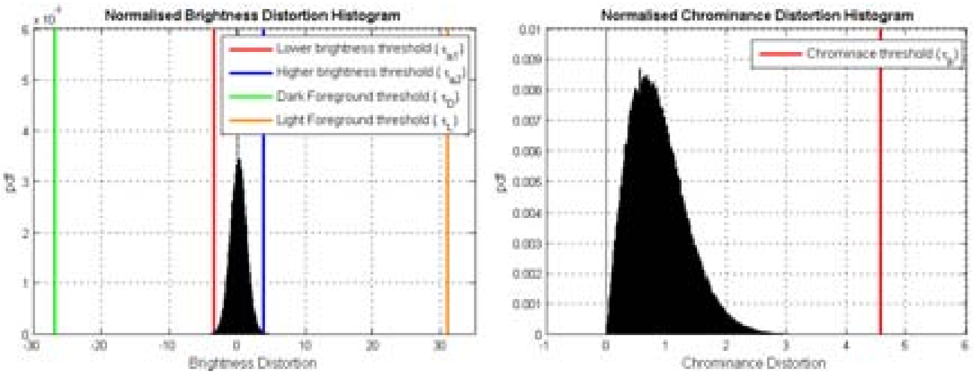
$$\breve{\alpha}_{a,t} = \frac{\alpha_{a,t}}{\bar{\alpha}_a}, \tag{5.9}$$

$$\breve{\beta}_{a,t} = \frac{\beta_{a,t}}{\bar{\beta}_a}. \tag{5.10}$$

This process is repeated during a temporal window of T_{tr} frames in order to avoid errors due to an insufficient number of samples. Subsequently, the histograms of both accumulated measures $\breve{\alpha}_{a,t}$ and $\breve{\beta}_{a,t}$ are computed taking into account all pixel distortions during the temporal window. Detection rates are used to set a lower and higher brightness distortion thresholds, $\tau_{\alpha 1}, \tau_{\alpha 2}$, and a chrominance threshold, τ_β.

Two thresholds are set for both dark and light foreground cases, where the current pixel is beyond the sensor dynamic range:

$$\begin{aligned} \tau_D &= \kappa_D \tau_{\alpha 1}, \\ \tau_L &= \kappa_L \tau_{\alpha 2}, \end{aligned} \tag{5.11}$$



(a) (b)

Figure 5.12: Threshold computation. Thresholds are automatically computed by accumulating histogram values and applying a detection rate.

where usually $\kappa_D = \kappa_L = \kappa$ is a factor that specifies the confidence region. Fig. 5.12.(a) shows the normalised brightness distortion histogram for a given frame, as well as the corresponding thresholds; Fig. 5.12.(b) shows the normalised chromatic distortion histogram and the computed threshold.

Finally, the threshold used for pixel segmentation according to BIM is computed as:

$$\tau_a^I = \max\left(\kappa^I \sigma_a^I, \epsilon\right), \tag{5.12}$$

where κ^I is the factor that sets the confidence region, and ϵ is a small positive quantity.

5.3.2 Image Segmentation

Input images can now be segmented by classifying the pixels according to computed background models and the current sensor response, see Fig. 5.13. Thus, three general cases are considered, and a different model is applied in each one:

- the BCM is applied to those pixels whose current values are inside the sensor dynamic range, and for which a BCM could be built;
- the brightness component of the BCM is applied to segment those pixels whose current values are beyond this range which also have a BCM;
- and, the BIM is applied to the those pixels which do not have enough values within the linear sensor range during the modelling process.

A sketch of the Image-Segmentation Module is shown in Fig 5.14. As a result, a segmentation map $\mathbf{M}_t$ is computed at each time step. Thus, pixels under

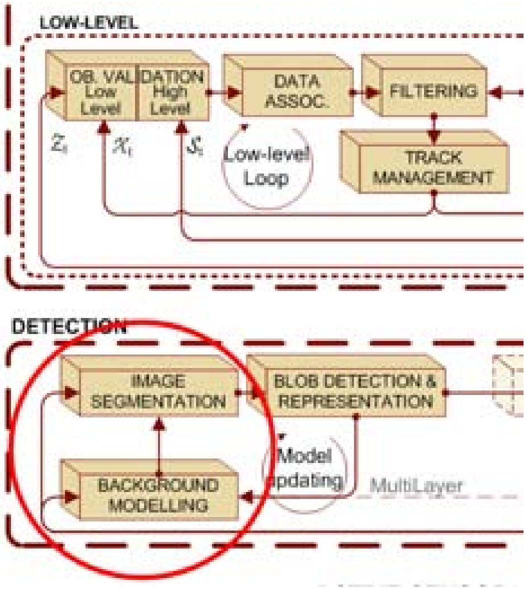


Figure 5.13: Segmentation module.

the first condition are classified as background (BgC), highlight (H), shadow (S), or foreground (FgC); those under the second one as background (BB), or dark foreground (DF) and light foreground (LF); and those under the last one as background (BgI) or foreground (FgI). This process is performed according to the following equation:

$$\mathbf{M}_{a,t} = \tag{5.13}$$

$$= \begin{cases} \text{BgC} & : \exists\text{BCM} \wedge \tau_m < I^c_{a,t} < \tau_n \wedge \tau_{\alpha 1} < \breve{\alpha}_{a,t} < \tau_{\alpha 2} \wedge \breve{\beta}_{a,t} < \tau_\beta \\ \text{S} & : \exists\text{BCM} \wedge \tau_m < I^c_{a,t} < \tau_n \wedge \breve{\alpha}_{a,t} < \tau_{\alpha 1} \wedge \breve{\beta}_{a,t} < \tau_\beta \\ \text{H} & : \exists\text{BCM} \wedge \tau_m < I^c_{a,t} < \tau_n \wedge \breve{\alpha}_{a,t} > \tau_{\alpha 2} \wedge \breve{\beta}_{a,t} < \tau_\beta \\ \text{FgC} & : \exists\text{BCM} \wedge \tau_m < I^c_{a,t} < \tau_n \wedge \breve{\beta}_{a,t} > \tau_\beta \\ \text{BB} & : \exists\text{BCM} \wedge I^c_{a,t} < \tau_m \vee I^c_{a,t} > \tau_n \wedge \tau_D < \breve{\alpha}_{a,t} < \tau_L \\ \text{DF} & : \exists\text{BCM} \wedge I^c_{a,t} < \tau_m \wedge \breve{\alpha}_{a,t} < \tau_D \\ \text{LF} & : \exists\text{BCM} \wedge I^c_{a,t} > \tau_n \wedge \breve{\alpha}_{a,t} > \tau_L \\ \text{BgI} & : \exists\text{BIM} \wedge I^c_{a,t} < \tau_m \vee I^c_{a,t} > \tau_n \wedge \left| I^I_{a,t} - \mu^I \right| < \tau^I_a \\ \text{FgI} & : \exists\text{BIM} \wedge I^c_{a,t} < \tau_m \vee I^c_{a,t} > \tau_n \wedge \left| I^I_{a,t} - \mu^I \right| > \tau^I_a \end{cases}$$

where $c \in \{R, G, B\}$ denotes the colour channel, and $c = I$ the intensity; $\tau_m \tau_n$ give the sensor dynamic range. The whole process is summarised in Algorithm 4.

An example of image segmentation can be seen in Fig. 5.15.(a). As it can be seen, despite the heavy shadows caused by both agents in an environment with several light sources, they are correctly segmented.

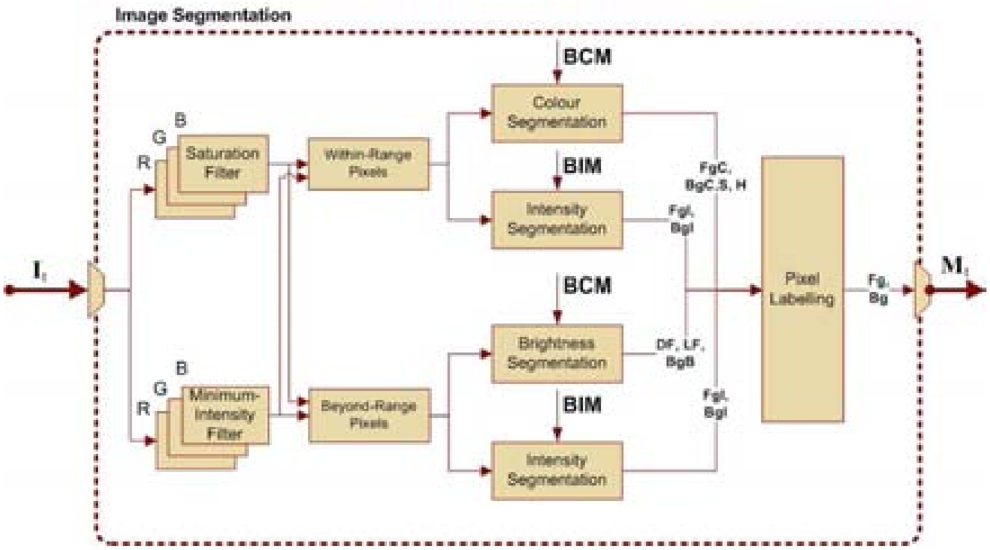
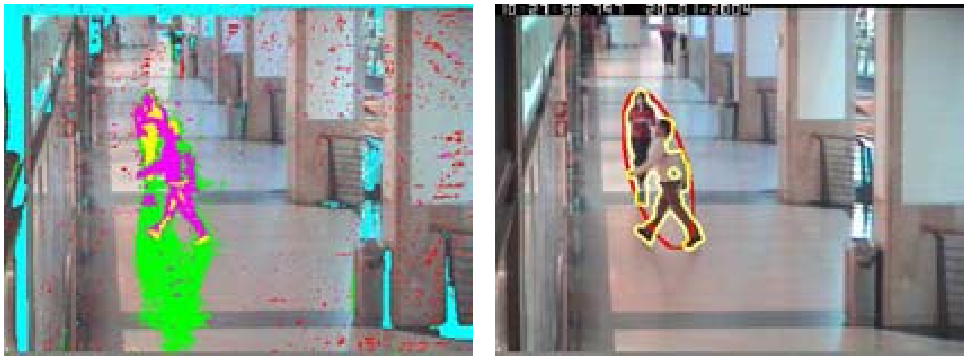


Figure 5.14: Image segmentation approach. As a result of applying background model to the current frame, pixels are classified according to the BCM as as foreground (FgC), background (BgC), shadow (S), and highlight (H); using the BCM on pixels beyond the sensor dynamic range, as dark foreground (DF), light foreground (LF), and background (BgB); and according to the BIM as foreground (FgI) and background (BgI).



(a) (b)

Figure 5.15: (a) Segmentation example: segmented foreground pixels using the BCM are painted on magenta, while shadows are painted on green, and highlights on red; dark-foreground pixels are painted in yellow, and light-foreground ones in orange; segmented foreground pixels using the BIM are painted in lilac, while background ones are in cyan. (b) Detection example: red ellipses represent each target, and yellow lines denote their contour.

Algorithm 4 Image segmentation.

```
• if BCM exists for the current pixel, then:
    – if it is within the sensor dynamic range, then:
        ∗ if it has a different chrominance, then foreground,
        ∗ else if it has lower brightness, then shadow,
        ∗ else if it has higher brightness, then highlight,
        ∗ otherwise, original background.
    – else
        ∗ if it has lower brightness, then dark foreground,
        ∗ else if it has higher brightness, then light foreground,
        ∗ otherwise, original background.
• else if BIM exists, then:
    – if it has lower or higher intensity, then foreground,
    – otherwise, original background.
• otherwise, no background was visible during the training period*
```

* In this case a frame-differencing algorithm can be applied to segment moving pixels, and a new background-modelling process performed in the next temporal window.

5.3.3 Blob Detection and Representation

Subsequently, the blobs that may correspond to targets are extracted, see Fig. 5.16. First, the different foreground masks are fused; then, majority, closing and opening morphological filters are applied on the resulting mask; next, the surviving pixels are grouped into blobs by means of connected-component analysis; finally, a minimum-area filter is used.

Each blob is then labelled, and their contours are computed. Further, blobs are parametrically represented, as explained next. By using such a representation, the spurious structural changes that the blobs may undergo are constrained. These include target fragmentation due to camouflage, or the inclusion of shadows and reflections. Moreover, this representation can be handled by the low-level trackers, thereby filtering the target state and reducing also these effects. Representations based on ellipses are commonly used [14, 67]. Here, an orientable ellipse is chosen —which keeps the blob first and second order moments.

Thus, the j-observed blob at time t is given by the vector $\mathbf{z}_t^j = \left(\widetilde{x}_t^j, \widetilde{y}_t^j, \widetilde{h}_t^j, \widetilde{w}_t^j, \widetilde{\theta}_t^j\right)$,

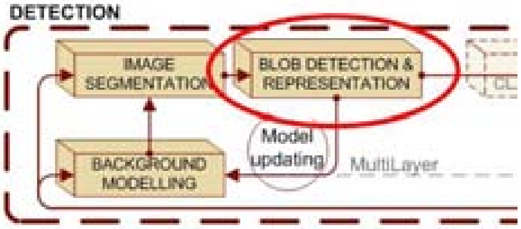


Figure 5.16: Blob-detection and representation module.

where $\widetilde{x}_t^j, \widetilde{y}_t^j$ represent the abscissa and ordinate of the ellipse centroid, $\widetilde{h}_t^j, \widetilde{w}_t^j$ are the major and minor axes, respectively, and the $\widetilde{\theta}_t^j$ gives the angle between the abscissa axis and the ellipse major one. An example of target detection can be seen in Fig. 5.15.(b).

5.3.4 Remarks on the Detection Level

These modules work in cascade, and eventually close the feed-back loop, see again Fig. 5.1 on page 82. Therefore, the background model can be updated taking into account a temporal window of segmentation results.

The proposed modular architecture allows us to substitute the currently-used background subtraction method with another one —which may be found more convenient in the future— without modifying the system architecture. Further, new functionalities can be added by inserting new modules. Thus, targets could be classified into several categories, which include people, vehicles, and unknown objects. At this stage, this would be done according to shape and/or appearance criteria. Further, this a-priori results could be refined after tracking is performed, thereby including stability and motion-based classification criteria.

5.4 Low-level Tracking (LLT)

Low-level motion trackers establish coherent target relations between frames by setting correspondences between observations and state predictions, and by estimating new target states according to the sequence of associated noise observations. In order to accomplish this task, four processes are carried out, see Fig. 5.17.

In the first place, *gates* are computed by the *observation-validation* module. These are the regions where the observations are expected to appear. This is done according to the target state and the system uncertainties. Subsequently, *data association* is performed. In this stage, correspondences between observations and trackers are set based on a nearest-neighbour decision —within the gate— in the observation space. Then, *filtering* is carried out: new target states are estimated according to the associated observations. This is here accomplished by a bank of KFs. Finally, the *track-management* module (i) initiates tentative tracks for those observations which

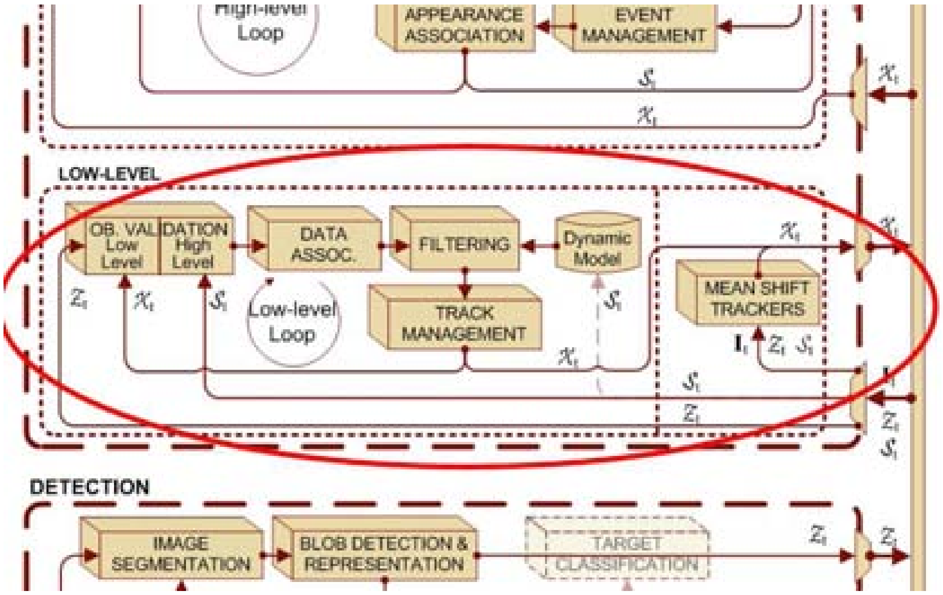


Figure 5.17: Low-level tracking.

are not associated; (ii) confirms tracks with enough supporting observations; and (iii) removes low-quality ones. Results are forwarded to high-level trackers, and fed back to the measure-validation module.

It is interesting to remark that a motion-tracking functionality is here established; however, each of these modules can be implemented using other algorithms without modifying the architecture itself, like a JPDAF for data association, or a UKF to perform the estimation task.

This level also includes an appearance-based tracker which is used to track grouped targets. In case of this event, segmented blobs contain multiple targets, which may actually conform a group, or be an effect of the viewing angle. Thus, tracking based on motion segmentation is not feasible, and therefore an appearance tracking is carried out. This decision is taken by the higher level, once the scene events are analysed.

5.4.1 State-Space Model

In this work, targets are assumed to move slowly enough compared to the frame rate. Since their long-run dynamics are hardly predictable, a first-order dynamic model is adopted. This assumption holds in most HSE applications on trajectory analysis. The target state is defined by $\mathbf{x}_t^j = \left(x_t^j, \dot{x}_t^j, y_t^j, \dot{y}_t^j, h_t^j, \dot{h}_t^j, w_t^j, \dot{w}_t^j, \theta_t^j\right)$, which establishes a state variable for every observation one and adds the target speed and the size change rate. Thus, the model considered is given by a constant-speed approach where the acceleration is modelled as *White Additive Gaussian Noise* (WAGN) —except for the angle variable θ_t^j, whose speed is modelled as noise: this variable is here considered to undergo minor variations, i.e. humans will essentially remain in upright posture.

The LLT dynamic model is defined by the following equations:

$$\begin{aligned} \mathbf{x}_t^j &= \mathbf{A}\mathbf{x}_{t-1}^j + \boldsymbol{\omega}_t \\ \mathbf{z}_t^j &= \mathbf{C}\mathbf{x}_t^j + \boldsymbol{\nu}_t, \end{aligned} \tag{5.14}$$

where $\boldsymbol{\omega}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is the process noise, $\mathbf{Q}$ the noise covariance, $\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is the segmentation noise, and $\mathbf{R}$ the noise covariance. WAGN is assumed to represent both noise processes. It is also assumed that both process and measurement noises are uncorrelated. Finally, the acceleration is supposed to be constant during the sampling period, and independent between periods.

The target dynamics can be described using block matrices for each pair of first-order model variables —such as for example $\mathbf{x}_{t,1} = (x_t, \dot{x}_t)^T$. Thus, previous assumptions allows us to define this system as:

$$x_t = x_{t-1} + \Delta_t \dot{x}_{t-1} + \frac{1}{2}\Delta_t^2 \ddot{x}_t, \tag{5.15}$$

$$\dot{x}_t = \dot{x}_{t-1} + \Delta_t \ddot{x}_t \tag{5.16}$$

$$\ddot{x}_t \backsim \mathcal{N}(0, \sigma_x), \tag{5.17}$$

where Δ_t is the sampling period, σ_x the variance of the noise process which models the acceleration. Thus, the transition matrix is given by:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix}, \tag{5.18}$$

the output matrix is:

$$\mathbf{C}_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \tag{5.19}$$

and the system noise in terms of sampled acceleration:

$$\boldsymbol{\omega}_t = \mathbf{G}_1 \ddot{x}_t, \tag{5.20}$$

where $\mathbf{G}_1$ is the noise matrix for a first-order system, given by:

$$\mathbf{G}_1 = \begin{pmatrix} \frac{1}{2}\Delta_t^2 & \Delta_t \end{pmatrix}^T. \tag{5.21}$$

Thus, the system covariance matrix results in:

$$\mathbf{Q}_1 = \operatorname{cov}(\mathbf{G}_1 \ddot{x}_t) \tag{5.22}$$

$$= \mathbb{E}\left[\mathbf{G}_1 \ddot{x}_t \ddot{x}_t^T \mathbf{G}_1^T\right] \tag{5.23}$$

$$= \begin{pmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 \end{pmatrix} \sigma_x, \tag{5.24}$$

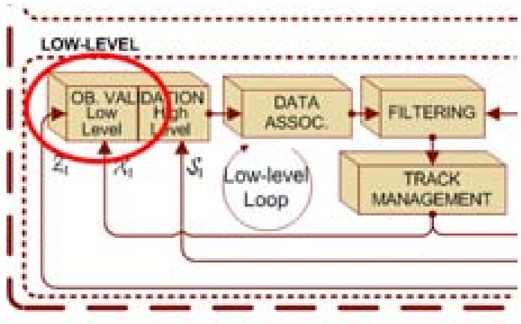


Figure 5.18: Observation-validation module.

where the equality $\mathbb{E}\left[\mathbf{G}_1 \ddot{x}_t\right] = \mathbf{G}_1 \mathbb{E}\left[\ddot{x}_t\right] = 0$ has been taken into account. The output covariance matrix is:

$$\mathbf{R}_1 = \mathbb{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^T\right] = \sigma_v, \tag{5.25}$$

where σ_v is the variance of the observation noise. Thus, the target dynamic matrices are given by the replication of the above-defined block matrices.

5.4.2 Observation Validation

In a MTT scenario, numerous observations may be obtained at every sampling period. In this case, some observations could have been generated by clutter or noise processes, and several observations might correspond to the same target with a given probability. Thus, gates are computed in agreement with the target state and the system uncertainties, see Fig. 5.18.

The observation vector at time t, $\mathbf{z}_t$, is given by the blob detection module. Each target expected observation is predicted according to the system dynamics:

$$\hat{\mathbf{z}}_t \quad = \quad \mathbf{C}\mathbf{A}\mathbf{x}_{t-1}. \tag{5.26}$$

Since the estimation is performed following a Kalman filtering scheme —see Appendix D for details— the prior error covariance matrix is computed accordingly:

$$\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q}; \tag{5.27}$$

and subsequently, the innovation covariance is obtained:

$$\mathbf{S}_t = \mathbf{C}\mathbf{P}_t^- \mathbf{C}^T + \mathbf{R}. \tag{5.28}$$

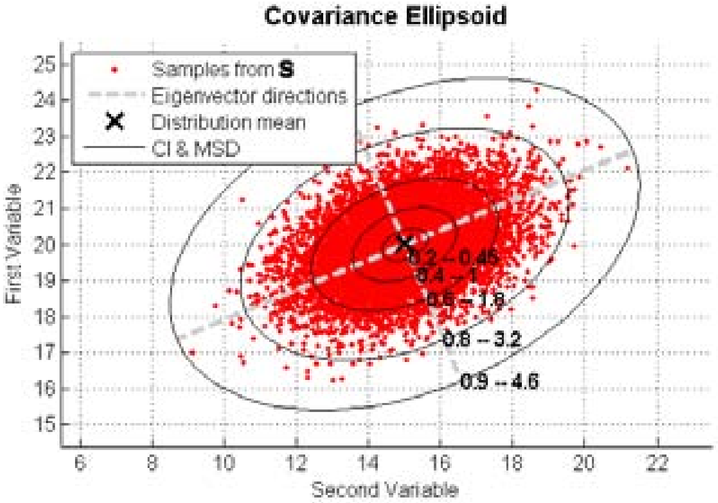


Figure 5.19: Innovation covariance ellipsoid. The predicted observation is given by the mean, and samples represent potential observations. Different ellipsoids are given at several Confidence Intervals (CI), thereby providing the MSD of the sample points lying on them.

This covariance matrix defines an ellipsoid in the observation space whose axes are given by the covariance matrix eigenvectors, and the axis length —for the ellipsoid with unit Mahalanobis radius— is given by the square root of corresponding eigenvalues. A particular Mahalanobis radius defines an ellipsoid, centred at the mean of the distribution, which encloses a probability mass given by the Confidence Interval (CI) associated with the ellipsoid, see Fig. 5.19.

Thus, the Mahalanobis Squared Distance (MSD) is given by:

$$d^2_{Mahal,t} \quad = \quad \left(\mathbf{z}_t - \hat{\mathbf{z}}_t\right) \mathbf{S}_t^{-1} \left(\mathbf{z}_t - \hat{\mathbf{z}}_t\right)^T , \tag{5.29}$$

and, provided that the observation follow a d-dimensional Gaussian pdf, the MSD is distributed according to a Chi-squared distribution with d degrees of freedom [24]:

$$d^2_{Mahal} \sim \chi^2_d. \tag{5.30}$$

Hence, the Mahalanobis radius corresponding to the ellipsoid with a given confidence interval can be computed by evaluating the inverse of the cumulative distribution function of the Chi-squared distribution. This means that measures can be validated for a given confidence interval by calculating the MSD between the predicted observation and the actual one, and comparing this value with the Mahalanobis radius for this confidence interval.

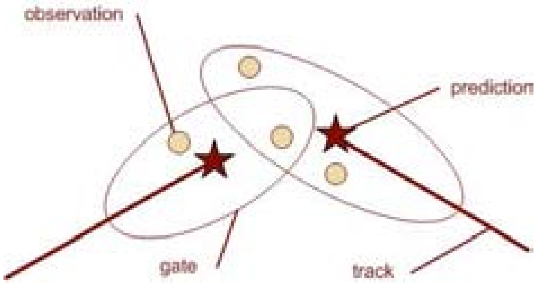


Figure 5.20: Observation association. Example with two predicted locations and their gates given by the respective trackers, and four observations. In both cases several observations are validated; one observation is even validated for both trackers.

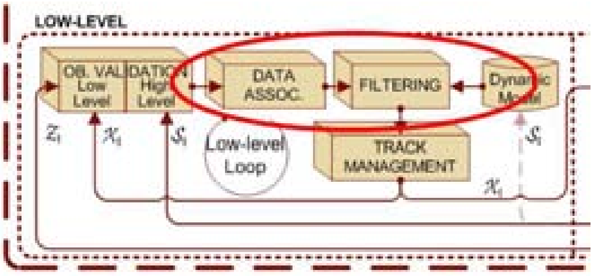


Figure 5.21: Data-association and Filtering modules.

5.4.3 Data Association and State Filtering

Once the gates have been computed, setting the correspondence between observations and trackers may not be straightforward: multiple observations may lie in the same gate, and some observations may be shared by more than one gate, see Fig 5.20.

Here, observations are associated to the nearest tracker in whose gate they lie, see Fig. 5.21. A more complex data association method, such as JPDAF, is not considered to be necessary since observations are usually just within one target gate. This is intrinsic to the segmentation method: if two targets are so close in the observation space as to introduce ambiguity in the data association process, the segmentation module is likely to segment just one blob corresponding to the group formed by both targets. This issue will be later discussed at the high-level tracker section.

A bank of Kalman filters is implemented to estimate the state of all targets detected within the scene. The LLT dynamic model is given by Eq. (5.14), where the system matrices are built according to the above-defined block matrices[5].

As a special case, if no observation is associated to a particular target, its state is estimated using a Kalman Gain equal to zero, that is, it is just propagated according to the dynamic model. See Appendix D for details.

[5] Independence between position and size state components is assumed.

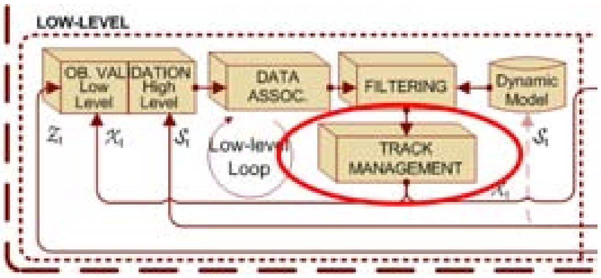


Figure 5.22: Track-management module.

5.4.4 Track Management

This module manages the target tracks by instantiating, confirming and removing them: (i) not associated observation initiates tentative tracks; (ii) tracks with enough supporting *satisfactory* observations are confirmed; and (iii) those tracks which lose confidence are removed, see Fig. 5.22. This is done according to the values of two indicators: the square root of the covariance matrix determinant $|\mathbf{S}|^{\frac{1}{2}}$, and the observation MSD.

The first one is related to the track uncertainty: the determinant is given by the product of the matrix eigenvalues, which correspond to the variance of the dimensions given by the respective eigenvectors. That means that while an observation is associated, the track uncertainty decreases to its asymptotic value, and the time taken depends only on the system dynamics and uncertainties. Thus, the innovation covariance matrix is calculated recursively according to Eq. (5.28), which just depends on the time-independent and known system matrices $\mathbf{A}, \mathbf{C}$, and $\mathbf{Q}, \mathbf{R}$.

That is to say, the track uncertainty does not depend on the observation MSD. It is however a good indicator of how many observations have been associated, and whether there have been frames without any observation. This is done without the need of setting thresholds and specifying cases: it is intrinsic to the behaviour given by the system dynamics.

Nevertheless, the quality of the observation must also be taken into account, and therefore, the MSD of each target associated observation is evaluated. The MSD, seen as the Mahalanobis radius of the ellipsoid, is used to qualify those observations which lied inside the ellipsoid of a given variance, τ_{σ^2}.

Therefore, a track is instantiated every time an observation has not being associated to any existing trackers. When the track uncertainty is below a certain percentage of its asymptotic value, and the MSD is lower than a given ellipsoid variance, the track is confirmed as stable. That means that a sequence of observations where successfully associated in the past recent frames, and there is a little error between the prediction and the current observation. If track uncertainty grows beyond a pre-set confidence value, the LLT is deleted, and the Kalman filter removed.

An example of the evolution of the track-management indicators can be seen in

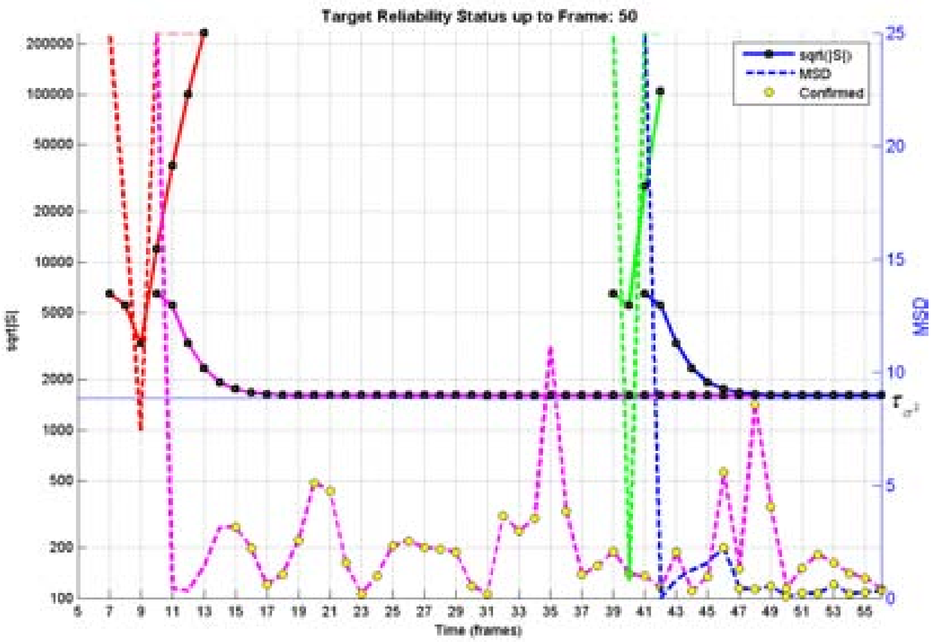


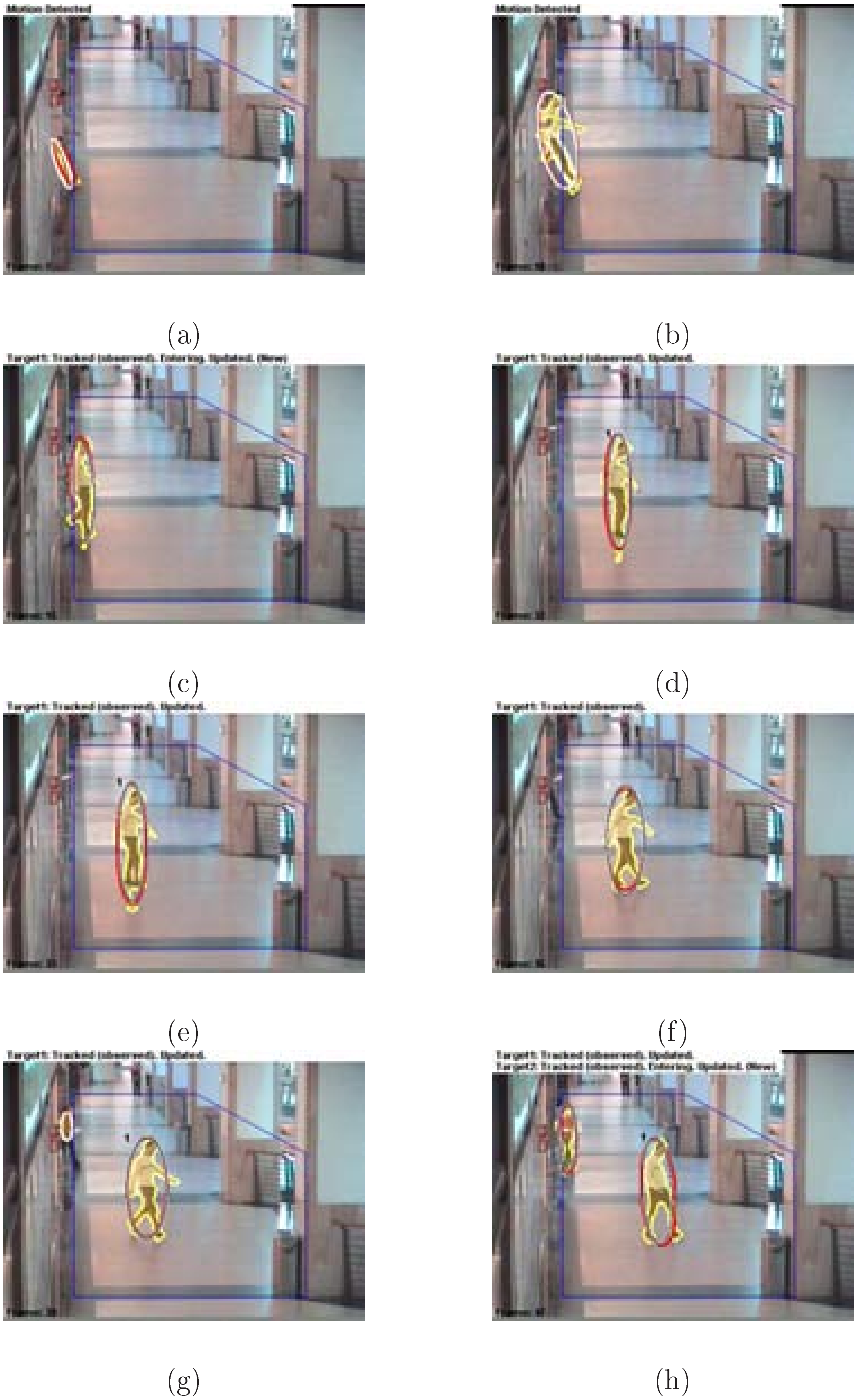
Figure 5.23: Track management. Tracks are confirmed when both track uncertainty and MSD are low enough. Tracks with high track uncertainty are removed. See text for details.

Fig. 5.23, and some sample frames[6] are shown in Fig. 5.24: at frame 7 —Fig. 5.24.(a)— a target starts entering the scene, an observation is received and a tracker is instantiated; while new observations are associated, the track uncertainty decreases. However, at frame 10 —Fig. 5.24.(b)— a major change happened —because the target has completely entered the visual field— and the MSD is so high that the observation is not associated to the existing tracker. Consequently, a new one is instantiated. The former one stops receiving observations and its tracks uncertainty keeps growing until frame 13, when the tracker is finally removed[7].

At frame 15 —Fig. 5.24.(c)— the track uncertainty of this second LLT is close enough to its asymptotic value, and the MSD is lower than the equivalent distance defined in terms of the variance. Thus, the track is confirmed. During frames 32 and 33 —Fig. 5.24.(d) and (e)— shadows and specular reflexions are included in the segmented blob. At frame 35 —Fig. 5.24.(f)— an abrupt correction causes a

[6] The following notation is used: blob contours are painted in yellow, while red ellipses represent detected blobs, and white and black ones give low- and high-level tracker estimates, respectively. The blue box denotes the ROI.

[7] It must be said that problems caused by target entering and exiting are also handled by the high-level trackers, as it will be explained in the corresponding section.


Motion Detected
Motion Detected
Target1: Tracked (observed). Entering. Updated. (New)
Target1: Tracked (observed). Updated.
Target1: Tracked (observed). Updated.
Target1: Tracked (observed).
Target1: Tracked (observed). Updated.
Target1: Tracked (observed). Updated.
Target2: Tracked (observed). Entering. Updated. (New)

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.24: Sample frames for track management. See text for details.

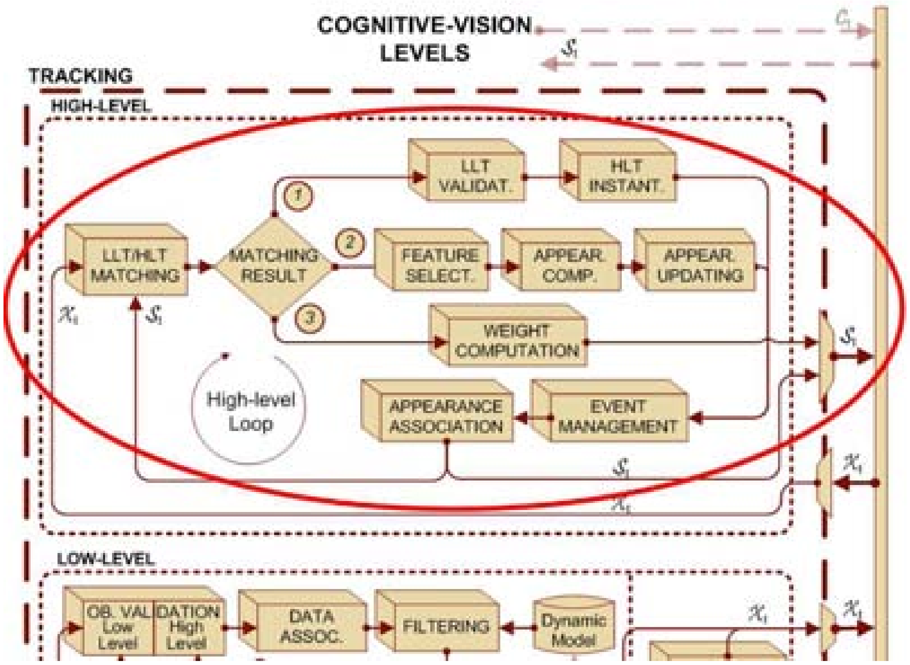


Figure 5.25: High-level tracking.

MSD high enough so that the tracker is temporarily non-confirmed. When a new target enters the scene at frame 39 —Fig. 5.24.(g)— a new track is instantiated —the observation is far beyond the gate boundary— and the previous process is repeated. At frame 47 both tracks are confirmed —Fig. 5.24.(h).

Several considerations must be taken into account. In the first place, depending on the system matrices, the time needed to reach a value close to the asymptotic value of the track uncertainty may considerably vary. Thus, if $|\mathbf{Q}|$ grows, the dynamics are less reliable, the Kalman Gain grows, the state variables are more affected by the observation values, and the convergence is faster. On the other hand, if $|\mathbf{R}|$ grows, the measure is less reliable, the Kalman gain decreases, the predicted values are less affected by the current observation, and the convergence is slower.

Secondly, it is worth to notice that if the target shape or position abruptly changes, the observation may lie outside the tracker gate. In this case, a new Kalman filter is instantiated, and both, the old and the new one are now competing for the observations. Consequently, a high-level analysis is required to assign a common identifier to both trackers.

5.5 High-Level Tracking (HLT)

At the top of the system architecture, high-level trackers aim to obtain robust and accurate state estimates for every target within the scene, see Fig. 5.25. Motion LLT's cannot cope with situations of continuous target-segmentation failures. Among the causes of these failures, grouping events, partial and complete occlusions, non-smooth changes in position or shape, and target camouflage may be found, to cite a few. In this cases, the corresponding motion low-level trackers would gradually lose confidence due to the lack of associated observations, and will eventually be removed. Therefore, these issues must then be addressed by HLT. These trackers build appropriate target appearance models, and infer conceptual knowledge about the targets' situation.

Once this is achieved, the higher level can act on the lower ones following a top-down approach in numerous ways: by selecting MBT or ABT operation mode, by preventing the creation of non-feasible low-level tracks, by validating the association of observations to LLT, by associating several LLT to the same HLT, by maximising the discrimination between the target model and potential distracters, and by enabling the incorporation of a motionless objects into the background.

5.5.1 Tracking Operation Modes

As it has been stated, the proposed system implements two tracking approaches: MBT and ABT. The higher level selects the most appropriate operation mode according to the current situation in which the targets are involved. This is done by the *Matching module* from the information given by the *Event-Management module*.

In our experience, MBT usually outperforms ABT in every situation where no a-priori knowledge is available about the scene or the targets, specially when their appearances evolves over time: in open-world scenarios, the target appearance cannot be specified in advance, and an accurate initialisation is often not feasible. Further, it should be continuously updated, since it strongly depends on the target position, its orientation to the camera and the different light sources, or —in case of human targets— the body posture. However, the need of adaptation usually leads to the phenomenon known as model drift[8].

Nevertheless, in case that no accurate segmentation can be produced, MBT is no longer feasible. The target state could be propagated according to the learnt dynamic model, but this usually does not suffice, since its motion is generally subject to sudden changes, and the probability of losing the target increases with the time the it is non-detected.

If, for example, the target is grouping, just a single blob, whose boundary encloses all connected pixels in motion, is detected. A coarse localisation —obtained by considering that the target is inside the group region— could be also considered, but it cannot tackle any complex situation, like for instance those in which a group of more-than-two members split, see again Fig. 2.12 on page 36. These cases require the use of ABT methods.

Thus, in our system, segmentation by motion is used whenever this is possible, and the system takes advantage of these situations to build accurate target models. ABT tracking success will be determined by the ability of distinguishing the target from potential distracters. In order to be able to track them under difficult situations

[8] Classical adaptive tracking problem, where the model gradually drifts as misclassified pixels are used to update it. This contamination leads to further localisation errors, and eventually to a complete tracking failure.

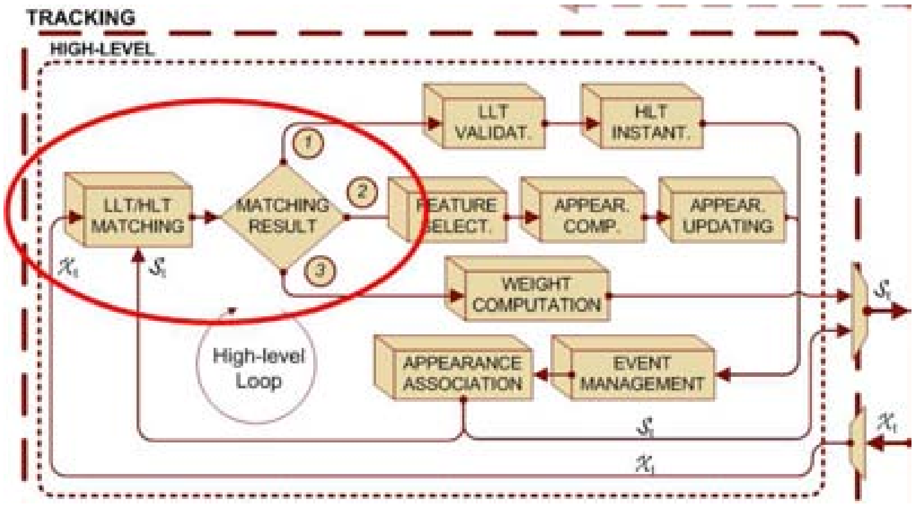


Figure 5.26: Matching module.

—those in which target segmentation is not feasible— the target appearance is represented by taking into account the local background clutter, as well as other targets with whom it may interact. Then, a robust ABT method is applied.

5.5.2 High-Level Tracker Management

HLT are instantiated by indication of the Matching module, which performs the association between existing low-level and high-level trackers, see Fig. 5.26. The matching procedure may lead to different kinds of conclusions, and for each of them the system exhibit a particular response. This process works as follows.

The module considers three cases: (i) the first time a low-level tracker is confirmed, a high-level tracker is instantiated and associated, see matching result (1) in Fig. 5.26. This may correspond to an isolated target, or to a group of them. The actual situation is determined according to the information relative to target collision provided by the Event-Management module. In case that the new-born tracker corresponds to an isolated target, the target appearance is then computed. In other case, it is marked as a group tracker. (ii) If a LLT is already associated, the high-level tracker parameters relative to the target position and shape are updated in subsequent tracker matchings, see matching result (2). Further, while the track is still confirmed, this situation is pointed out so that the appearance will also be computed and updated.

While the associated low-level motion tracker exists, the targets are tracked by motion. However, this matching process is not always feasible, since LLT's may have been removed during long-duration segmentation failures due to the continuous lack of an associated observation. Thus, (iii) those targets which have no correspondence are tracked in a top-down process using low-level ABT, see matching result (3). This makes possible target tracking even when image segmentation is not feasible, such as during long-term occlusions, grouping events or target camouflage. During these situations, the LLT gradually loses confidence due to the lack of associated observations, and is eventually removed. Hence, the aim of including an ABT is (i) to track those trackers which have no LLT, and (ii) to refine the localisation of those HLT with no associated observation.

Subsequently, an event-management module determines what is happening within the scene: the target-interaction events are inferred, and the entering or exiting of targets into/from the scene is established. Among the formers, interactions such as which targets are grouping or splitting, or whether stable groups are being formed, are set. This include complex combinations of them, since one target may be involved in different kind of interactions with several other targets.

Ultimately, those HLT which have no LLT associated are evaluated in order to decide whether a correspondence can be established with other HLT's, since new trackers are instantiated over targets that have undergone an event which cause LLT removal, once the event is over. If there are no tracker candidates, or they are not similar enough, in the appearance sense, the appearance-based operation mode holds[9] until the target can be associated to a new high-level tracker. Results are fed back and used for low-level and high-level tracker matching.

In the following, each module of the high-level tracking is explained in detail.

5.5.3 Appearance Representation based on Soft and Hard Pixel Weighting

The target appearance is here represented by means of colour histograms [61, 14, 67, 12]. Histograms are broadly used to represented target appearance, since they are claimed to be less sensitive than colour templates to rotations in depth, the camera point of view, non-rigid targets, and partial occlusions. They are also usually used to represent non-parametric distributions, provided that they allow one to achieve real-time performances given the low computational cost required.

The histogram of a target is given by:

$$\mathbf{p} \quad = \quad \{p_k ; k = 1 : K\}, \tag{5.31}$$

where K is the number of histogram bins, and the discrete probabilities of each bin are calculated as:

$$p_k \quad = \quad C \sum_{a=1}^{P} g_E \left(\|\mathbf{x}_a\|^2 \right) \delta \left(b \left(\mathbf{x}_a \right) - k \right), \tag{5.32}$$

where δ is the Kronecker delta, $\{\mathbf{x}_a ; a = 1 : P\}$ the pixel locations, P the number of target pixels, and $b(\mathbf{x}_a)$ a function that associates the given pixel a to its corresponding histogram bin. C is a normalisation constant required to ensure that:

[9] This is, for instance, the case when the targets are grouped.

$$\sum_{k=1}^{K} p_k = 1. \tag{5.33}$$

Finally, $g_E(x)$ is the convex and monotonic profile of an isotropic kernel which allows one to perform gradient-based searches, which need differentiable similarity functions. Further, by assigning lower weights to pixels farther from the centre, the influence of boundary clutter is diminished. Here, an Epanechnikov kernel has been used [14, 67]. To eliminate the influence of different target dimensions and aspect ratio, the kernel is first rescaled to an ellipse of the target size. Thus, pixels are here soft-weighted according to their spatial distribution.

The above-defined appearance histograms are computed given a cropped image region. Two sources of information are available to decide which pixels should be considered as belonging to the target, namely the silhouette of the associated observation —given by the detection module— and the filtered ellipse —given by the LLT.

In this work a conservative approach has been used in order to minimise the risk of failures caused by model drift. Thus, only those pixels which belong to both the detected silhouette and the filtered ellipse are taken into account to build the target model. By doing so, background pixels which have been erroneously detected —e.g due to reflections— or those inside the tracked ellipse are likely to be removed. Also, non-reliable boundary foreground pixels —such as those of the end of the limbs— are usually not taken into account. This can be seen as a hard pixel weighting.

Groups should be handled in a different way, since their shape can rarely be modelled as an ellipse[10]. Thus, a rectangular bounding box is used, and the group region is given by just considering the current detection instead. In this case, detection errors are not critical, since no appearance model is computed for the whole group, but for the different partners, and this location is only used for collision-detection purposes.

5.5.4 Feature Selection on Colour Cues

Colour cues have been here selected to model the target appearance, see Fig. 5.27. Numerous colour spaces can be used, and each of them has tunable parameters, resulting in an enormous space of potential features. By selecting the most appropriate ones, a maximum discrimination between the target and local distracters is obtained.

The following feature-selection technique has been evolved from the one presented in [12] by generalising it taking into account multiple clutter sources. Features are here selected considering not only the best distinction between the local background and the target, but also taking into account other nearby targets, which will be called *group partners* in the following. The information provided by the event module is used to decide which targets are partners, and in what sense. Thus, features are selected from a set of linear combinations of the R, G and B channels:

[10] Of course this refers to groups identified as so. Non-detected groups —due to the targets had entered the scene together— are tracked using an ellipse representation until the targets split.

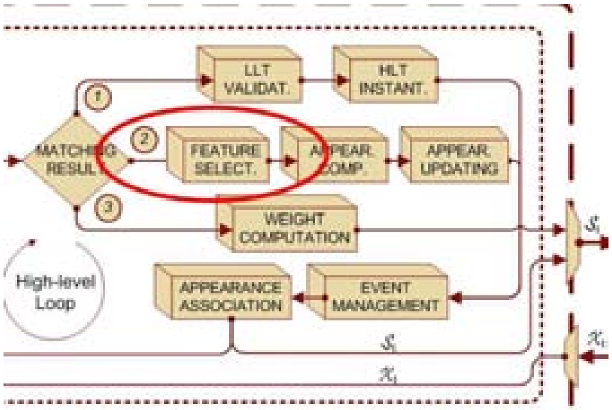


Figure 5.27: Feature-selection module.

$$\mathcal{F} = w^R R + w^G G + w^B B \, | w^c \in \{-2, -1, 0, 1, 2\} \, , \tag{5.34}$$

where $c \in \{R, G, B\}$. Hence, this set includes raw R, G, and B, intensity, and common chrominance approximations. The total number of candidates is 5^3. Non-independent combinations are removed, leaving a set of 49 features. Computed values are then normalised to the range $[0 : 255]$, and subsequently discretised. In the present implementation the number of bins is set to $K = 64$. This is a sensitive decision since a low number of bins will prevent from target-clutter disambiguation, but, on the other hand, a high value favours erroneous representations that appear when distributions are estimated from an insufficient number of samples, and thereby over-fitting the model and making it too sensitive to minor illumination changes.

5.5.4.1 Feature Selection in a n-class Problem

The target histograms are given by $\mathbf{p}^{i,j}$, where i denotes the feature index, and j the target one; and $\mathbf{q}^i$ provides the local background distribution according to the $i-th$ feature. Features are then ranked in the following way: first, the log-likelihood ratio of each feature is computed[11]:

[11] Here it is assumed that the joint distribution for the tuned feature can be computed by reusing the distributions already computed for the background and different targets involved. This essentially entails that the regions from which these distributions were computed should have a similar number of pixels. Although exact distributions can be computed by re-accumulating them, it is done in this way for sake of efficiency.

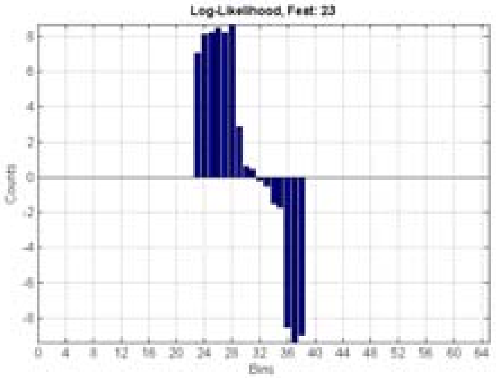


Figure 5.28: Log-likelihood. A negative value indicates that the bin has a higher occurrence in the clutter region, whereas positive ones correspond to a majority of target pixels.

$$\Lambda_k^{i,l} \quad = \quad \log \frac{\max\left(p_k^{i,l}, \varepsilon\right)}{\max\left(\frac{1}{J}\left(q_k^i + \sum_{j=1,j\neq l}^{J} p_k^{i,j}\right), \varepsilon\right)}, \tag{5.35}$$

where J gives the number of partners including the current $l-$target, and ε is set to prevent dividing by zero or taking the logarithm of zero, but avoiding also magnifying the corresponding log-likelihood value, see Fig. 5.28. Thus, shared colour bins have a log-likelihood close to zero, whereas target bins have a positive one, and clutter bins a negative one.

The variance of the log-likelihood according to a general discrete distribution $\boldsymbol{\phi}$ can be computed as:

$$\mathrm{var}\left(\boldsymbol{\Lambda};\boldsymbol{\phi}\right) \quad = \quad \mathbb{E}\left[\boldsymbol{\Lambda}^2\right] - \mathbb{E}\left[\boldsymbol{\Lambda}\right]^2 = \sum_k \phi_k {\Lambda^2}_k - \left(\sum_k \phi_k \Lambda_k\right)^2, \tag{5.36}$$

so features are then evaluated according to the variance-ratio of the log-likelihood:

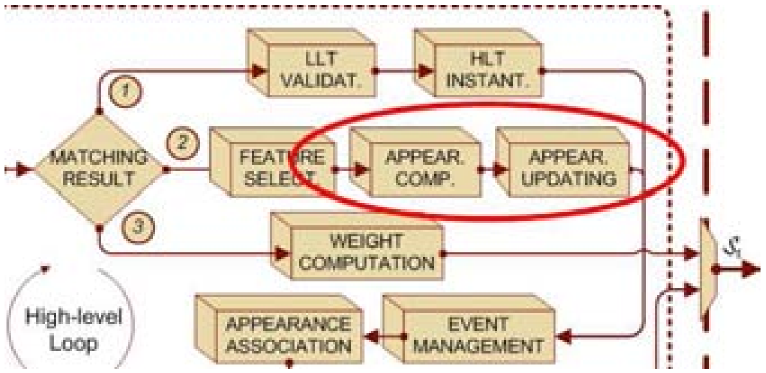


Figure 5.29: Appearance-modelling modules.

$$
\begin{aligned}
V^{i,l}\left(\mathbf{\Lambda}^{i,l};\mathbf{p}^{i,1},...,\mathbf{p}^{i,j},...,\mathbf{p}^{i,J},\mathbf{q}^{i}\right) &= \\
&= \frac{\operatorname{var}\left(\mathbf{\Lambda}^{i,l};\frac{1}{J+1}\left(\mathbf{q}^{i}+\sum_{j=1}^{J}\mathbf{p}^{i,j}\right)\right)}{\operatorname{var}\left(\mathbf{\Lambda}^{i,l};\mathbf{q}^{i}\right)+\sum_{j=1}^{J}\operatorname{var}\left(\mathbf{\Lambda}^{i,l};\mathbf{p}^{i,j}\right)},
\end{aligned}
\tag{5.37}
$$

and subsequently, they are ranked according to these values: the higher, the better. Thus, the selection maximises the inter-class variance —that is, the distance between clutter and target clusters of bins— while minimising the intra-class variance —tightly clustering both clutter and target bins. Thus, in order to allow the system to build reliable appearance models using the features which best distinguish a target from its potential distracters, once a grouping event is detected, the partners histograms are also used in the feature selection procedure.

5.5.5 Appearance Modelling

As it has been above stated, target models are based on histogram representations using colour cues. Summarising the approach so far, histograms are calculated from a given image region, once an Epanechnikov kernel has been applied to it. This region is defined by the intersection of the segmented silhouette and the filtered ellipse. Histograms are computed in a feature space given by a linear combination of the R, G and B channels. Channel weights are selected in order to maximise the target discrimination form potential distracters.

Upon this basis, multiple models for each target are built and kept updated, see Fig. 5.29. Self-similarity statistics are also computed. In these way, we aim to solve the initialisation, smooth the representation, and complete it so that tracker association is feasible once the event that cause the target loss is over. The possibility

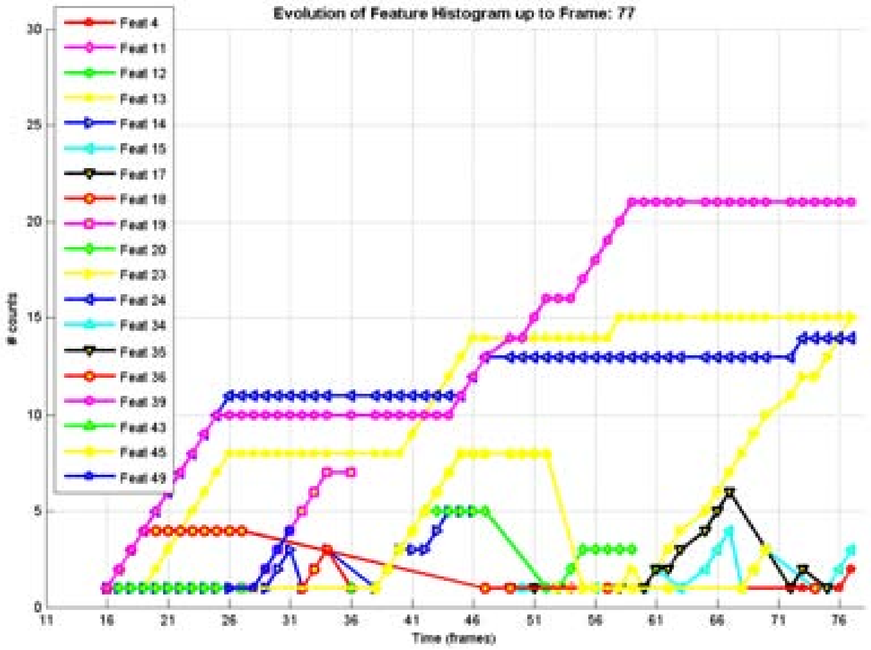


Figure 5.30: Feature pool. The best M features at time t, and the best N long-run features are kept for appearance modelling.

of an inconsistent localisation due to feature switch is also minimised, by introducing the distinction between long-run features and the current best ones. Thus, long-run features are here kept and smoothed. The representation scheme proposed in [12] is therefore considerably enhanced in this work —in addition to the fact of using the background and partner models to obtain a maximum target discrimination.

5.5.5.1 Model Pool

By keeping a set of long-run features, the system robustness is significantly increased. Histograms can be smoothed, thereby making the representation less sensitive to potential initialisation and subsequent localisation errors. This can also cope with sudden and temporal appearance changes, for instance due to illumination fluctuations. Further, past features may be crucial for tracker association after a tracking failure.

Hence, a pool of $M + N$ features is kept: the best M features at time t, and the best N long-run features, i.e. those which have been at the top of the feature ranking more times. These features are only dropped when new features enter the pool, and eventually overcome the formers.

An example of how the feature pool evolves over time is shown in Fig. 5.30. At each time step, the number of times a particular feature has been selected among the

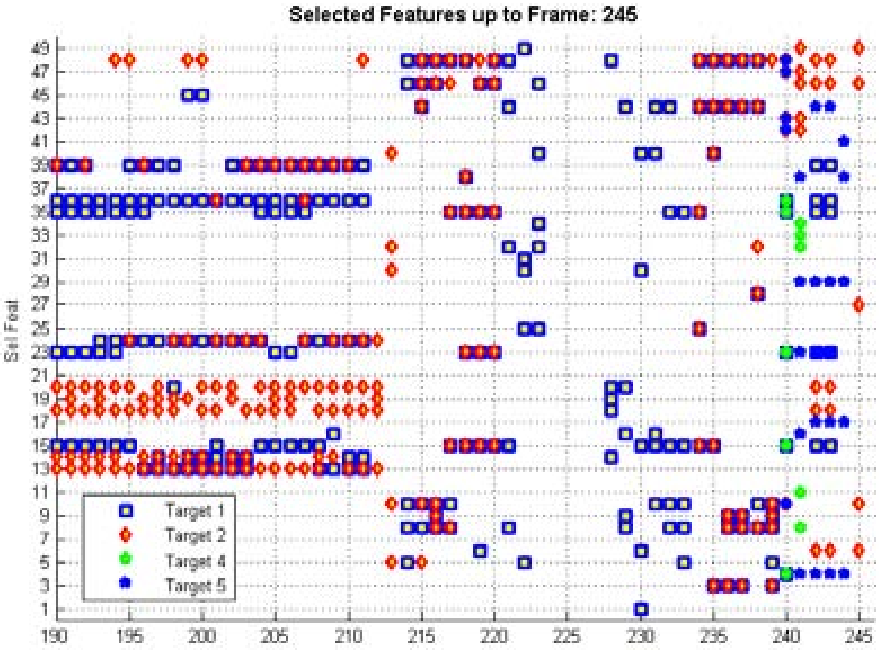


Figure 5.31: Example of selected features on a given interval where targets are grouped.

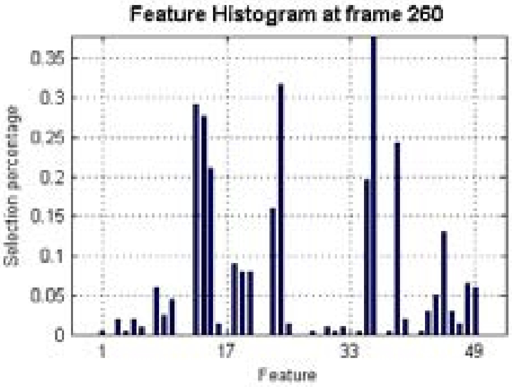


Figure 5.32: Histogram of feature selection on the whole sub-sequence.

best M ones is represented. Features selected in a given interval and the corresponding histogram are shown in Figs. 5.31,5.32 respectively. By analysing the evolution of the model pool, several facts can be noticed: some features are periodically among the best ones (in this case, features number 13, 24 and 39); this repetitive behaviour is presumably due to similar agent orientations and gait during tracking. Some features join the pool and quickly become one of the best ones (feature number 23), as the

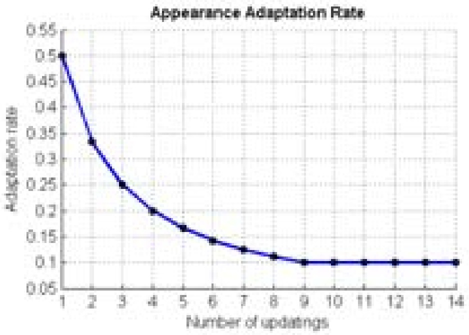


Figure 5.33: Adaptation rate. It presents a transient and a steady-state response to accommodate contrary requirements.

agent moves and the local background changes. Finally, other features (20, 36, 45) are dropped and re-selected several times; they are periodically among the best ones ones, but they are not selected enough times, and due to the pool size are dropped when others join the set. These behaviours strongly suggest that keeping a stable set of features may be useful for tracker association after a tracking failure.

5.5.5.2 Model Updating

Whenever there is enough confidence on the tracker to update the appearance, all $M + N$ models are updated. This is done in a recursive way using an adaptive filter:

$$\overline{\mathbf{p}}_t^{i,j} = \overline{\mathbf{p}}_{t-1}^{i,j} + \alpha_{\mathbf{p}} \left(\mathbf{p}_t^{i,j} - \overline{\mathbf{p}}_{t-1}^{i,j} \right), \qquad (5.38)$$

where $\alpha_{\mathbf{p}} \in [0:1]$ denote the adaptation rate which weights the most recent values versus the historic ones, and $\overline{\mathbf{p}}_t^{i,j}$ the smoothed histogram of target j at time t using the i-th feature. Old values are exponentially forgotten according to this rate: the bigger it be, the faster old data are forgotten. However, contrary requirements must be fulfilled: (i) when a feature is recently added, the model should be fast adapted, in order to cope with potential detection errors during the initialisation; (ii) medium-term models should not be excessively adapted, to prevent model-drift phenomena; (iii) long-term models should be3 adaptive enough, so that the system can handle unexpected appearance changes. This suggest defining the adaptation rate in terms of time, and to employ a principled function $\alpha_{\mathbf{p}}(t)$. Thus, a recursive mean filter is first used, thereby fulfilling the two first requirements, but the adaptation rate is fixed to a high enough value after an initialisation period, and thus model adaptation could be performed during arbitrary long time periods. An example is shown in Fig. 5.33.

In this way, once a target is detected, and the corresponding low-level motion-based tracker is confirmed, the target is being tracked while it is simultaneously being modelled by the high-level tracker. New features can be added, while stable ones build robust appearance models, even during hard situations, as it will demonstrated later on.

5.5.5.3 Model Similarity

A similarity measure between two histograms is computed using the following metric [14, 67]:

$$d_{Bhat} = \sqrt{1 - \rho\left(\mathbf{p}, \mathbf{q}\right)}, \tag{5.39}$$

where

$$\rho\left(\mathbf{p}, \mathbf{q}\right) = \sum_{k=1}^{K} \sqrt{p_k q_k} \tag{5.40}$$

is the Bhattacharyya coefficient. A similarity criterion is set in order to establish when two histogram are close enough. For this purpose, every time the smoothed histogram is updated, the mean and variance of the Bhattacharyya metric d_{Bhat} between the former histogram and the new one are also recursively updated:

$$\mu_t^{i,j} = \mu_{t-1}^{i,j} + \frac{1}{n^{i,j} - 1}\left(d_{Bhat,t}^{i,j} - \mu_{t-1}^{i,j}\right), \tag{5.41}$$

$$\left(\sigma_t^{i,j}\right)^2 = \frac{n^{i,j} - 3}{n^{i,j} - 2}\left(\sigma_{t-1}^{i,j}\right)^2 + \left(n^{i,j} - 1\right)\left(\mu_t^{i,j} - \mu_{t-1}^{i,j}\right)^2, \tag{5.42}$$

where $n^{i,j}$ is the number of times this particular feature histogram has been updated. In this way, the metric distribution is parameterised and used to establish a confidence measure.

5.5.6 Appearance-Based Tracking (ABT)

This operation mode is chosen by the HLT to cope with those situations in which MBT is not feasible, such as target camouflage, grouping and partial occlusion, see Fig. 5.34.

However, in general, ABT methods are very sensitive to changes in the illumination conditions. Further, the background and nearby targets can act as appearance distracters, thereby causing the tracker to erroneously lock on them. In this work, a distracter-robust mean-shift method is developed and used when no valid target observation is available.

A smart system should take advantage of all possible sources of information in order to minimise the risk of target loss when no accurate motion-segmentation can be performed. Potential sources of information include, among others, an updated

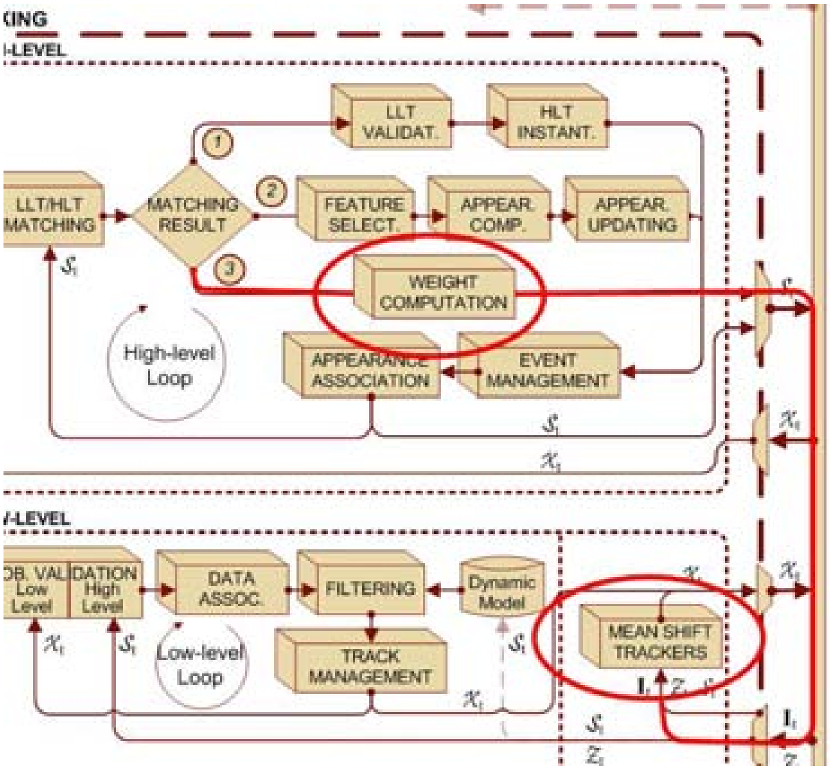


Figure 5.34: Mean-shift module.

background model, the current frame segmentation, the estimate state of all targets within the scene, and their appearance models, and the prediction of collisions and occlusions according to the learnt dynamic models and appearance ones.

Thus, a mean-shift procedure is here enhanced by making a principled use of all the knowledge inferred. This methods weight each candidate pixel according to its supposed membership to a determine target, given its appearance model. However, the target's appearance evolves in a unknown manner over time, and the local background and nearby targets may mimic its appearance. In order to achieve a successful tracking, this ambiguity must be minimised.

First, multiple appearance histogram models are simultaneously used. These have been built during the MBT stage by taking into account the most appropriate features which provide a maximum discrimination between the target and local distracters.

Target candidate regions should be wisely chosen, since neither the detection, nor the estimation is free of errors: segmented regions may include shadows and reflections, but may not enclose all target pixels due to camouflage problems; the estimate target region may include pixels of the background and of nearby targets

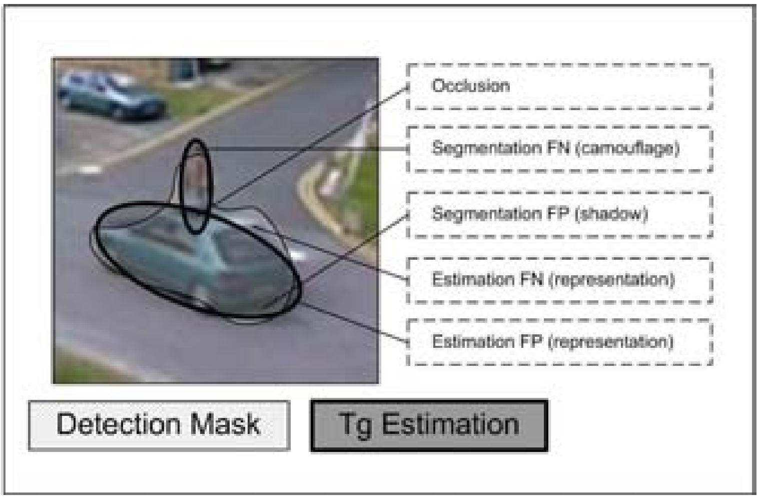

Occlusion
Segmentation FN (camouflage)
Segmentation FP (shadow)
Estimation FN (representation)
Estimation FP (representation)
Detection Mask
Tg Estimation

Figure 5.35: Merging targets. While the target estimate regions may include pixels of the background and of nearby targets, the detection mask include shadows and reflections, but do not enclose all target pixels due to camouflage problems. (FP and FN denote a False Positive and a False Negative, respectively).

due to errors introduced by the state representation, see Fig. 5.35. Therefore, the current motion segmentation is used to help discriminating background pixels. Thus, those pixels which are not segmented are weighted according to an estimate error-segmentation rate. Occluded regions are also taken into account when building the target candidate histograms.

Further, shared model bins with both the background and nearby targets are made less significant. Finally, an spatial exclusion is set in order to avoid that a same pixel significantly contributes to locate more than one target. The complete approach which combines appearance, motion and spacial cues to perform target localisation is shown in Fig. 5.36.

Subsequently, potential drift of the appearance models is precluded by performing a careful updating according to the detected events and the evaluation of the tracking results. These procedures are explained in the following.

5.5.6.1 Mean-shift Technique

This technique achieves target localisation by performing a deterministic gradient-descent search on a image region of interest —the basin of attraction— which is previously weighted [14]. In the following, a brief explanation is given.

The target model is given by the histogram $\overline{\mathbf{p}}$, while the target candidate distribution at the image location $\widehat{\mathbf{x}}_0$ is represented by $\hat{\mathbf{p}}\left(\widehat{\mathbf{x}}_0\right)$. The similarity between two histograms is computed using the metric defined in Eq. (5.39).

The mean-shift procedure recursively moves the candidate position to a new lo-

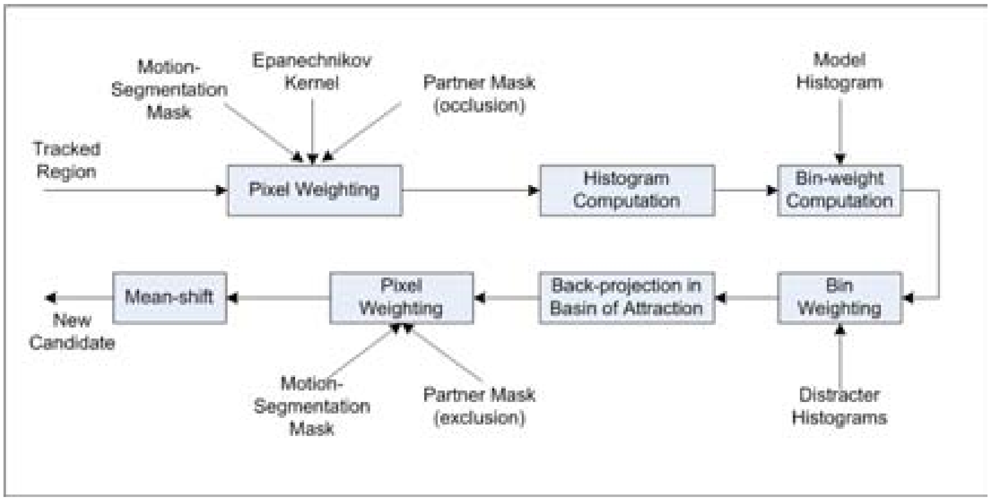


Figure 5.36: Multiple-cue Mean-shift. The approach combines motion, appearance and spatial cues to perform target localisation in presence of distracters.

cation, while searching the local minimum according to the aforementioned metric. That is to say, a new location is searched in the neighbourhood of the former one by maximising the similarity between the target model and the candidate one, computed from the current image at this location. This is approximately equivalent to minimise the second term of the Taylor expansion of the Bhattacharyya coefficient which represents a weighted-data density estimate computed with the kernel profile [14]. Thus, the new location is given by:

$$\hat{\mathbf{x}}_1 = \frac{\sum_{a=1}^{M} \mathbf{x}_a w_a \dot{g}_E \left(\left\| \widehat{\mathbf{x}}_0 - \mathbf{x}_a \right\|^2 \right)}{\sum_{a=1}^{M} w_a \dot{g}_E \left(\left\| \widehat{\mathbf{x}}_0 - \mathbf{x}_a \right\|^2 \right)}, \tag{5.43}$$

where the weights w_a are given by:

$$w_a = \sum_{k=1}^{K} \sqrt{\frac{\overline{p}_k}{\hat{p_k} \left(\hat{\mathbf{x}}_0 \right)}} \delta \left(b \left(\mathbf{x}_a \right) - k \right). \tag{5.44}$$

By choosing an Epanechnikov kernel, both kernel profile derivatives $\dot{g}_E$ in Eq. (5.43) can be removed by taking into account that the derivative of the profile of an Epanechnikov kernel is a constant. The complete algorithm is shown in Algorithm 5.

Algorithm 5 Mean-shift method.

```
1. the histogram of the target candidate $\hat{\mathbf{p}}$ is computed at location $\hat{\mathbf{x}}_0$,
2. weights are computed according to Eq. (5.44),
3. the next target location $\hat{\mathbf{x}}_1$ is derived following Eq. (5.43),
4. if $\|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_1\| < \epsilon$, or the maximum number of iterations has been reached,
   stop. Otherwise set $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_1$ and go to step 1.
```

5.5.6.2 Basin of Attraction and Target Candidate Region

A mean-shift procedure assigns weights to each histogram *bin* according to a relation between the model and candidate histograms, and then back-projects these values into image *pixels*, before computing the new proposed localisation. Thus, each pixel is weighted according to its supposed membership to a determine target, given its appearance model.

The tracked region is given by the previous estimated location. Both size and orientation are kept fixed. This yield an ellipse from which the candidate histogram is computed.

Bin weights are back-projected in a basin of attraction given by the rectangle of dimensions $h * w$ pixels —the one which encloses the tracked ellipse— plus an outer margin of $\kappa_m * \max(h, w)$. κ_m is usually equal to 0.1. This margin provides better chances of tracking success in case of low frame rates —which may cause that successive target region do not overlap— and aspect ratio changes.

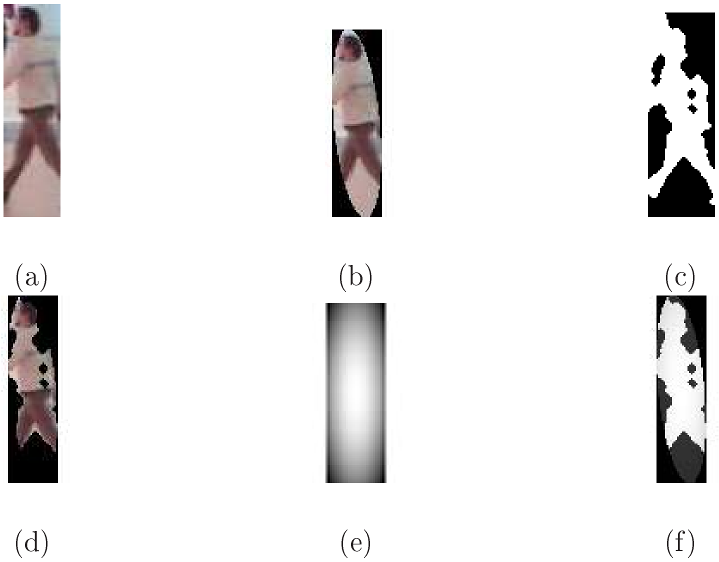
5.5.6.3 Introducing Motion Cues. Pixel weighting

The current segmentation can be used to weight the influence of each pixel on the candidate histogram, and on the weighted sub-image where the search is performed. By doing so, the system is making use of the results obtained by the detection level, but without neglecting the possibility of segmentation errors.

Thus, according to motion information, those pixels within the target candidate region which are not segmented are also weighted according to an estimate error-segmentation rate, see Fig. 5.37. Further, the same procedure is applied to the weights of the pixels of the basin of attraction after back-projection.

In addition, candidate pixels contribute to the histogram with a value in the interval $[0:1]$, according to the applied kernel. In this proposal, the Epanechnikov kernel is combined with the detection mask, thereby minimising the risk of over-weighting significant distracters bins.

Finally, in case that the tracked target is partially occluded —what is inferred by the Event-Management module— the affected parts are not taken into account when computing the target histogram.



(a) (b) (c)

(d) (e) (f)

Figure 5.37: Pixel weighting. (a) Cropped candidate. (b) Tracked region. (c) Applied Motion-segmentation mask (cropped to the basin-of-attraction size). (d) Candidate region. (e) Epanechnikov kernel. (f) Epanechnikov kernel applied to the candidate mask —which is given by the conjunction of the mask of the tracked region and the motion-segmentation mask.

5.5.6.4 Bin-weighting

So far, background and partners' information has been used to select the features that best discriminate the target from a local environment, see Fig. 5.38. However, even for the best features, histogram bins could be shared between the target and potential distracters. This fact leads to an erroneous localisation, which finally ends causing the drift of the appearance models. This can also be accelerated due to the fact that the foreground is hardly ever perfectly delineated.

To minimise tracking failures due to this issue, the following approach is proposed: the background-weighting approach proposed in [14] is here generalised by including other sources of information: the appearance models of the partners and the learnt local background model. Further, this is applied to each appearance model computed from a particular feature. The learnt background presents the advantage that it contains no foreground information. However, it may differ from the current one, for instance, due to the occlusion of some light source.

A conservative approach has here also been chosen: all significant bins in any of the aforementioned sources of knowledge about potential distracters will have its importance diminished, see Fig. 5.39. Thus, an histogram of the local background is computed using the learnt background model:

$$\mathbf{q}^i \quad = \quad \left\{q_k^i; k = 1 : K\right\}. \tag{5.45}$$

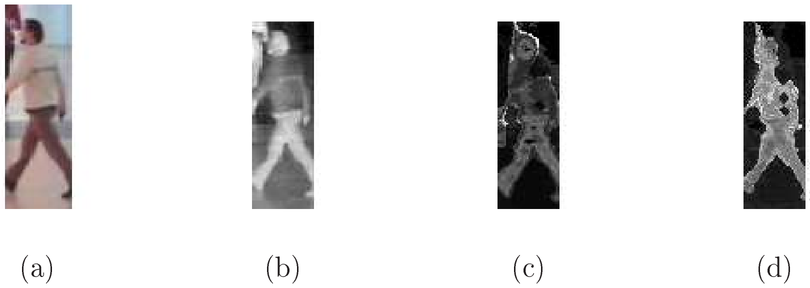


Figure 5.38: Weighted images. (a) Tracked region. (b) Image mapped according to a selected feature (Feature 36, $V = 0.7653$, 25th in feature rank). (c) Corresponding weighted image. (d) Weighted image for a feature with higher Variance Ratio (Feature 19, $V = 1.0123$, second in feature rank). Notice that the latter feature is much more discriminative than the former one —which is in the model pool for being a long-run feature.

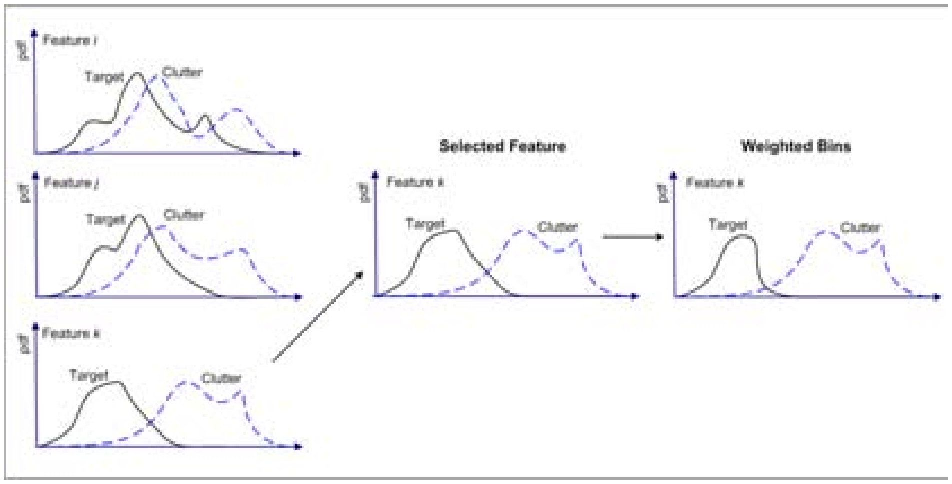


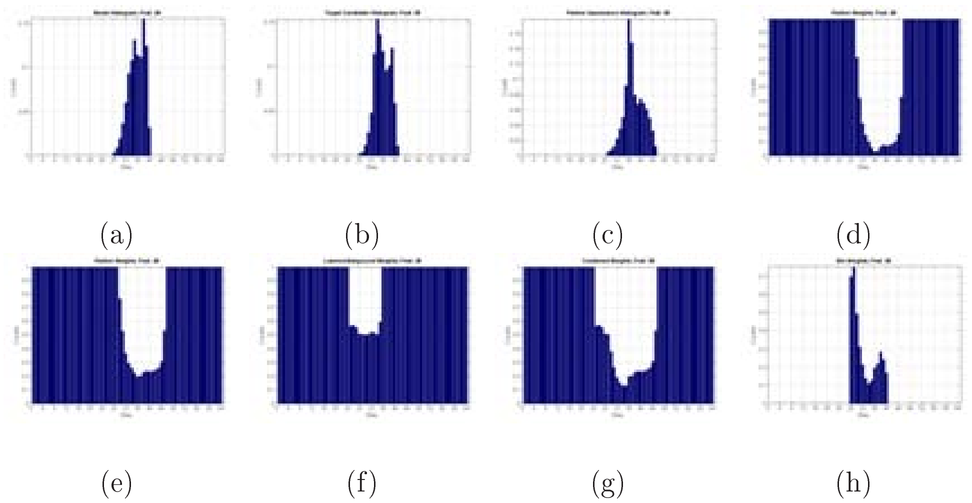
Figure 5.39: Maximisation of target discrimination. The best discriminant features between the target model and the clutter are selected. Then, shared bins are made less significant.

Then, a weight for each bin is derived from its significance on this histogram:

$$w_k^{i,q} \quad = \quad \left\{ \min \left(\frac{q_k^{i*}}{q_k^i}, 1 \right); k = 1 : K \right\}, \tag{5.46}$$

where $\overset{i}{q_k^*}$ is the minimum non-zero value. These values are equalised according to a predefined maximum rate of exclusion, $\eta^q \in [0:1]$:

$$\tilde{\mathbf{w}}^{i,q} = \eta^q + \frac{\left(\mathbf{w}^{i,q} - \min\left(\mathbf{w}^{i,q}\right)\right)}{\max\left(\mathbf{w}^{i,q}\right) - \min\left(\mathbf{w}^{i,q}\right)} \left(1 - \eta^q\right). \tag{5.47}$$



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.40: Bin weighing. (a) Model histogram. (b) Candidate histogram. (c) Partner histogram. (d) Partner bin weights. (e) Partner equalised bin weights. (f) Background equalised bin weights. (g) Combined bin weights. (h) Resulting bin weights for back-projection.

The same technique is used to compute weights for each partner, $\tilde{w}_k^{i,j}$. Thus, for the l-target among the J targets of the group, the total weight of each model bin, given the i-th model feature, is obtained by combining these weights:

$$\omega_k^{i,l} = \tilde{\omega}_k^{i,q} \prod_{j=1,j\neq l}^{J} \tilde{\omega}_k^{i,j}. \tag{5.48}$$

These weights can then be applied to the target model to diminish the importance of those bins which are shared with potential distracters:

$$w_a = \sum_{k=1}^{K} \omega_k^{i,l} \sqrt{\frac{\overline{p}_k}{\hat{p_k}}}. \tag{5.49}$$

Bin weighting according to the appearance of local distracters can be seen like a probabilistic exclusion principle. Such a technique has also been used in [57] in order to avoid that an edge feature can correspond to several targets. In other words, one particular evidence must not contribute to mutually exclusive hypotheses. In our case, shared model bins must not reinforce the different local maxima in the weighted image where the mean-shift is computed. The maximisation of the target discrimination is graphically shown in Fig. 5.40.

5.5.6.5 Spatial Exclusion

Following the aforementioned exclusion principle, the back-projected values of the candidate pixels are also weighted in other to avoid that the same pixel contributes to locate the centroid of more than one target. Thus, for each target, an exclusion kernel is computed from the location of the partners. A flat kernel is applied by setting the partner region to a pre-defined exclusion rate. Notwithstanding, this approach could be enhanced by computing probabilistic masks for each target. These would record the likelihood of the target being observed at that pixel.

5.5.6.6 Criteria to Perform an Appearance Updating

A bank of $M+N$ mean-shift procedures is run, and each of them uses an appearance model tuned at one of the selected features. These models need to be updated even when the targets are grouped, since their appearances are always subject to undergo significant changes —specially when the targets are in motion. However, the updating of the appearance models must be carefully done in order to avoid model drift.

Therefore, the multiple results obtained are first evaluated and filtered according to appearance and localisation criteria. Then, the surviving results are eventually fused in order to produce a robust estimate. This final estimate can be used to perform model updating.

First, those mean-shift procedures which have not converged after a number of iterations are not considered reliable enough to take them into account to perform the updating. Next, an appearance gate is computed to filter those features whose histograms significantly differ from the models according to the learnt feature statistics of self-similarity:

$$d_{Bhat,t}^{i,j} \quad < \quad \mu_t^{i,j} + \kappa_{ABT}\sigma_t^{i,j}, \tag{5.50}$$

where κ_{ABT} is the factor which set the confidence region. Finally, a robust target localisation is obtained by filtering potential position outliers among the remaining features. This avoids that a feature model locked on a distracter similar in appearance corrupts the localisation computation. This is done by computing the position mean and variance, and removing the outliers. The procedure is iterated until convergence.

When at least one model survives the appearance filtering, the target similarity is again evaluated at the final estimate localisation. If the result is still satisfactory, then the estimate is considered reliable enough to perform model updating and feature selection.

Given that a candidate localisation is always necessary, even in the event of non having any reliable result according to the appearance criterion, the above robust-target localisation is performed, and the position is updated, but the appearance models are kept unmodified.

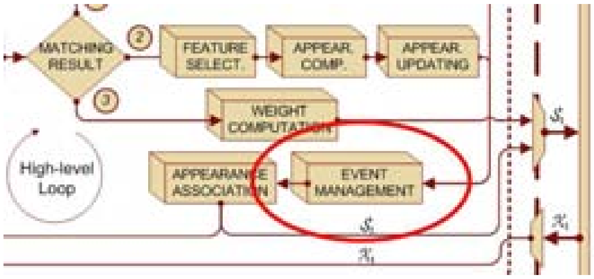
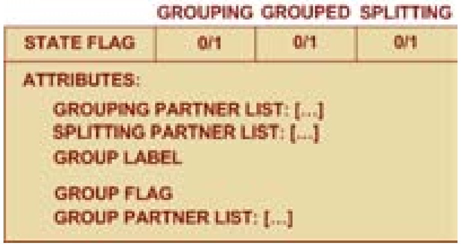


Figure 5.41: Event-management module.



	GROUPING	GROUPED	SPLITTING
STATE FLAG	0/1	0/1	0/1

ATTRIBUTES:

GROUPING PARTNER LIST: [...]
SPLITTING PARTNER LIST: [...]
GROUP LABEL

GROUP FLAG
GROUP PARTNER LIST: [...]

Figure 5.42: Target state coding.

5.5.7 Event Management

High-level understanding of motion events is a critical task in any system which aims to analyse dynamic human-populated scenes. MTT requires considering potential target interactions among them, specially when no assumption is made with respect to their trajectories. These kind of events will be referred in the following as *interaction events*. Among these, occlusions events deserve being explicitly addressed, given their particular difficulties.

Further, in open-world applications, targets can enter and exit the scene, or a Region of Interest (ROI) defined on it. These events will be referred as *scene events*, and they have an important role in matching low-level and high-level trackers, and in handling the latters.

However, current tracking techniques still do not address complex interaction events among multiple targets. In this thesis, a principled event management is proposed and embedded in the tracking architecture, see Fig. 5.41. Within the HSE framework, this module is located at the Conceptual-Integration Level, see Fig. 5.2 on page 83.

Multiple-target interaction events, and a proper scheme for tracker instantiation and removal according to scene events, are considered. Further, this allows the system to switch among different operation modes. Both types of events, and occlusions as a special interaction event, will be managed as follows.

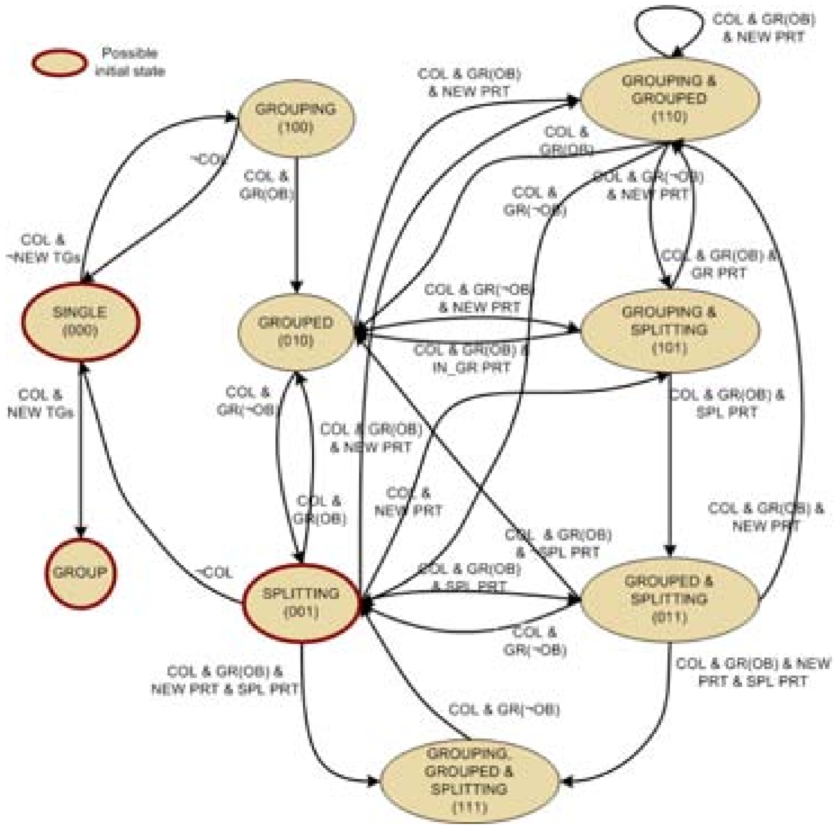


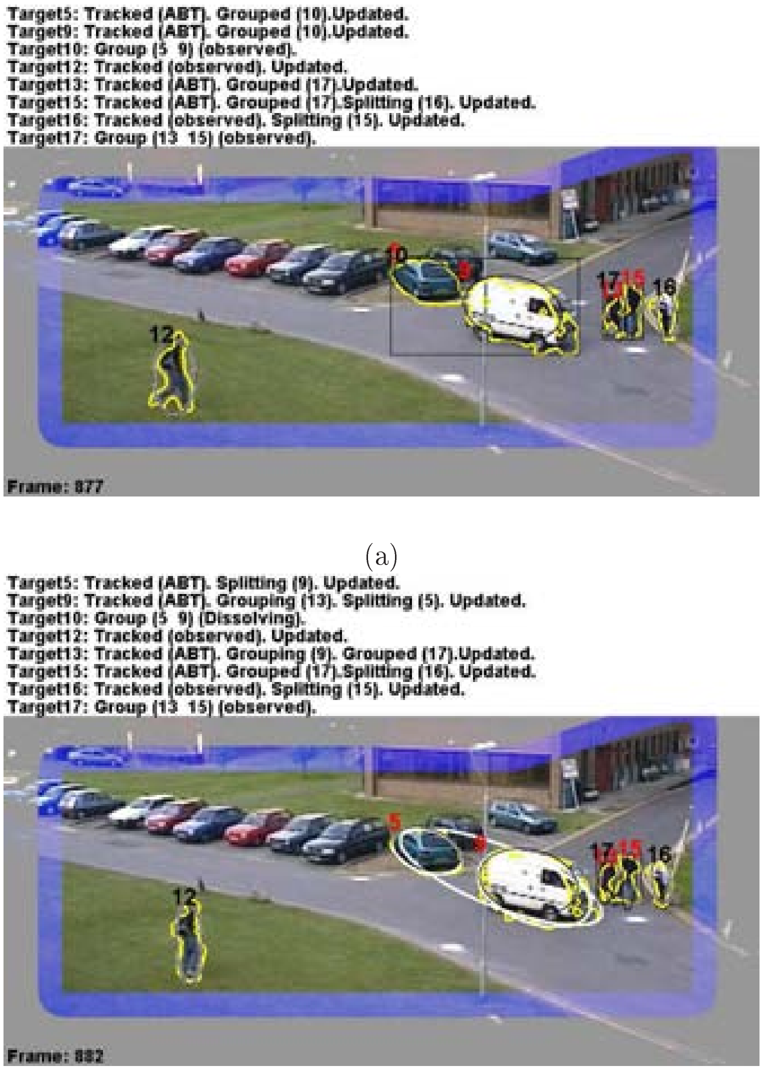
Figure 5.43: Group management. Eight possible target states (represented by ellipses), and a state for group trackers, are defined. Interaction events are denoted by arrows. Notice that some of the less frequent transitions are not drawn for the sake of clarity.

5.5.7.1 On Interaction Events

A proper detection of interaction events is crucial to achieve successful performances, since a different tracking approach must be used in each case: on the one hand, whenever a detected blob clusters more than one target, tracking by motion detection is no longer feasible, and no accurate target position can be obtained; on the other hand, ABT methods suffer from a poor target localisation, and therefore they are not the optimal choice when an appropriate detection can be performed. Thus, by detecting these events, several operation modes could be introduced and properly selected. Further, this represents a significant knowledge which can be used for scene understanding.

Two targets are said to be *in-collision* when their *safety areas*[12] superpose them-

[12] These areas are defined according to the targets' sizes by following the centre-surround approach shown in Fig. 4.5 on page 76



```
Target5: Tracked (ABT). Grouped (10).Updated.
Target9: Tracked (ABT). Grouped (10).Updated.
Target10: Group (5  9) (observed).
Target12: Tracked (observed). Updated.
Target13: Tracked (ABT). Grouped (17).Updated.
Target15: Tracked (ABT). Grouped (17).Splitting (16). Updated.
Target16: Tracked (observed). Splitting (15). Updated.
Target17: Group (13  15) (observed).
Frame: 877
```

(a)

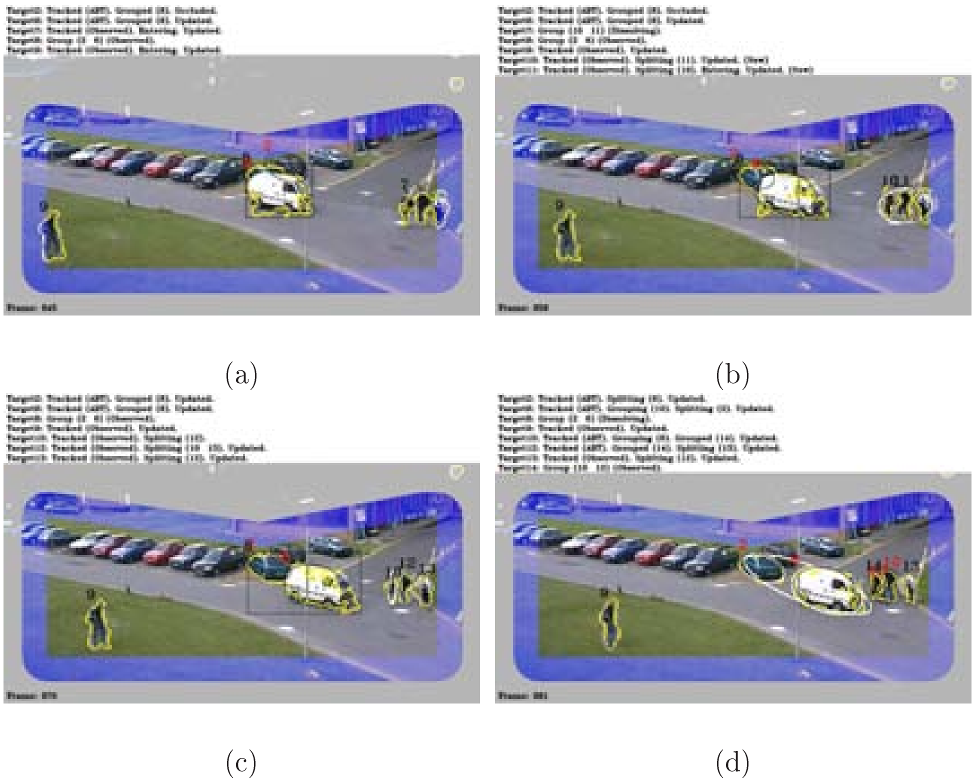
```
Target5: Tracked (ABT). Splitting (9). Updated.
Target9: Tracked (ABT). Grouping (13). Splitting (5). Updated.
Target10: Group (5  9) (Dissolving).
Target12: Tracked (observed). Updated.
Target13: Tracked (ABT). Grouping (9). Grouped (17).Updated.
Target15: Tracked (ABT). Grouped (17).Splitting (16). Updated.
Target16: Tracked (observed). Splitting (15). Updated.
Target17: Group (13  15) (observed).
Frame: 882
```

(b)

Figure 5.44: Sample tracking through interaction events. See text for details.

selves. Thus, once all targets' positions and sizes are estimated, a collision map is computed. The collision map is also used to determine whether a new-born tracker represents a group: in this case, it is instantiated over a collision zone.

The following states can be now defined: (i) a target is considered as *single* if it does not collide with any other target within the scene; (ii) targets are said to be *grouping* if they do collide, but no group is being tracked in their area; (iii) targets are considered as *grouped* if they collide, they are over a group tracker area, and the



(a) (b) (c) (d)

Figure 5.45: Sample tracking showing the detection of multiple targets that had not entered the scene isolately. (a) Target 7 is in fact a group, but is has not been detected yet since a single observation has always been received —the yellow contour shows the segmentation; (b) Target 10 and 11 are detected and marked as splitting, while 7 is now marks as a dissolving group —this information is shown in the corner box; (c) Target 12 and 13 are detected as splitting from 11, which was in fact a group of two; (d) a new group is conformed by target 10 and 12; this is marked as 14.

group tracker is currently associated with an observation; (iv) finally, trackers are said to be *splitting* once the group has no longer an observation, but they do still collide. The frame rate is supposed to be high enough so that a target cannot change from grouped to single without ever being splitting.

Unfortunately, the above-presented classification does not suffice in complex scenarios where clusters of more than one target may be formed —for instance, one target could belong to a stable group of several targets, while being grouping with some other targets at the same time as splitting from other ones. Hence, the aforementioned scheme should be generalised by taking into account multiple and different target interactions.

The interaction state is coded using a three-bit vector, where each bit point outs whether the target is grouping, grouped or splitting, see Fig 5.42. When every bit is set to zero, the target state is single. Otherwise, the state could be a mixture of the previously defined situations.

Secondly, several attributes are associated with each state. These point out relevant information to solve interesting queries about current interaction events: which targets are interacting? in which sense? which ones are simultaneously grouping and splitting? which are the partners of some grouped target? etc. Thus, the eight possible states include all potential tracking situation, and these, along with the associated attributes, constitute all the necessary knowledge to solve any query relative to target interaction.

Two cases concerning the attributes are distinguished, depending on whether the tracker tracks a target or a group of them. In the first case, two lists of grouping and splitting partners are kept. Further, the group label, if this exists, is stored. In the second one, a flag which points out that the tracker is actually tracking a group is set. In addition, a list of grouped targets is also kept.

Finally, several events must be taken into account in order to define state transitions. These include issues such as target collision with another target (COL), or with a group (GR), whether the group has an associated observation (GR(OB)) or not, if there are new partners in collision (NEW PRT), or splitting partners are still so (SPL PRT).

The state machine that models the group management is defined by eight plus one states, see Fig. 5.43. The formers are defined for target trackers, and the latter for group trackers. Thus, there are 56 potential transitions between target states, although a fraction of them are not feasible according to the aforementioned assumptions. For instance, grouped targets cannot become single, since they have to split before. It is also possible to perform changes in the attributes without this meaning a state transition. This is the case when several targets are already grouping, and a new one joins them.

As an example of complex interaction, consider a target whose state is grouped; then, the following events take place: (i) it is colliding with some other targets (COL), (ii) the group has currently no associated observation (GR($\neg$OB)), and (iii) new partners are also colliding (NEW PRT). As a result, it changes its state into *grouping and splitting*. Multiple of such complex interactions are shown in Fig. 5.44. The previous example is the case of target 9. The changes on interactions between the two shown frames are summed up in Table 5.1.

Although the current proposal do not allow yet to independently track people who do not enter into the scene isolated, a tracker is instantiated over the group region. However, the system does detect the targets as they split, recognises the former target as a group, and creates new trackers for each partner that splits from the group, see Fig. 5.45.

5.5.7.2 Occlusions events

Partial and complete occlusions may lead tracking to unrecoverable failures, if they are not properly handled. In a 2-D approach, these may occur due to target grouping

Target label	State ($t = 877$)	Attribute	State ($t = 882$)	Attribute
5	grouped	10	splitting	9
9	grouped	10	splitting	5
			grouping	13
12	–		–	
13	grouped	17	grouped	17
			grouping	9
15	grouped	17	grouped	17
	splitting	16	splitting	16
16	splitting	15	splitting	15
Group label		Attribute		Attribute
10	–	5,9	x	x
17	–	15,16	–	15,16

Table 5.1: Interactions of targets shown in Fig. 5.44.

—real, or due to or the effect of viewing angle— and background objects.

Partial occlusions cause inaccurate position and size updating. However, as long as the frame rate is high enough to ensure smoothed changes, this represents just a temporal estimate deviation. The main problem comes from the possibility of model drift during ABT. Further, basins of attraction cannot be accurately chosen, and partner features can be wrongly computed. This could cause target loss in a few frames. Further, the contaminated model prevents from any posterior target recovering. Thus, a proper occlusion handling is crucial for tracking success.

The following approach is here used to infer the occlusions status. The collision rate is known given the estimated target positions and sizes. The likelihood of the target estimate is also known, according to Eq. (5.50), where the updating was decided. Thus, a significant collision along with a remarkable fall in the target likelihood in comparison with historic values allow the system to infer that the target is being occluded by other group partner.

Once the target is considered as occluded many actions can be taken. First, the collision are is no taken into account in future appearance updatings. Further, this area is also discarded while appearance tracking is performed. Finally, this area can be securely taken to compute partner weights during ABT.

The occlusion status remains while any ambiguity exists. Thus, just in case that no collision is predicted, or the collision is no longer significant, and the target likelihood is high enough to perform an appearance updating, this status is changed.

5.5.7.3 On Scene Events

A proper handling of scene events is essential in order to achieve successful system performances in open-world applications. In these, the number of targets within the scene is not a-priori known, and it may vary as new targets enter the scene, or other ones exit it. By defining a Region of Interest within the scene boundaries three aims

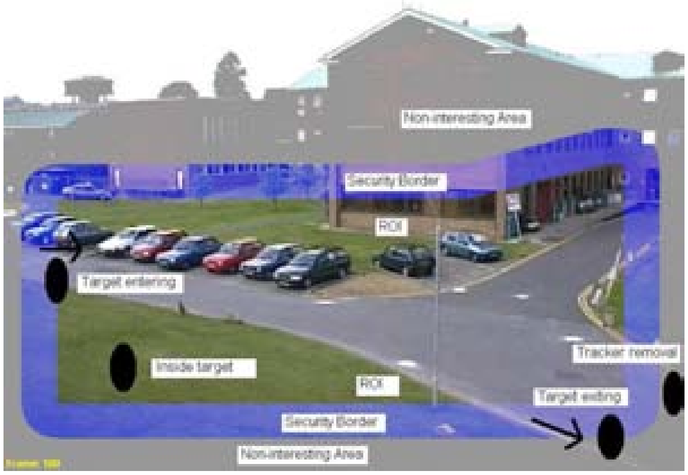


Figure 5.46: Scene regions. Example of scene events on an image from PETS database. The three regions define the ROI, a security border, and non-interesting areas. Events according to the target positions are shown.

are achieved: (i) it is not necessary to fully process the whole image, and therefore this favours accomplishing real-time performances; (ii) the number of false positives can be effectively reduced, by avoiding detections in non-plausible or non-interesting areas, like the sky in a pedestrian-surveillance application; and (iii) targets can be almost completely segmented, thereby avoiding major shape changes in targets partially out the field of view.

Three regions are here defined: a ROI, a security border, and non-interesting areas, see Fig. 5.46. These are used to determine where targets can be detected, where LLT's and HLT's can be instantiated, and when they can be removed. The security border prevents the system from alternatively creating and removing a tracker placed on the ROI frontier, in addition to avoid errors in the estimated shape in HLT.

The different system main tasks —pixel segmentation, blob detection, low-level blob tracking, and high-level target tracking— are performed according to noticeable changes in particular set of these regions.

Thus, pixel segmentation is carried out in the whole image, since targets' sizes are not a-priori known. However, targets are only detected —that is, they constitute an entity for the system— if a part of them is inside the ROI, or the centroid of the corresponding blob lies at least within the security border.

For each detected target, a low-level tracker is instantiated. However, not every LLT instantiate a high-level tracker. This requires that two conditions hold: (i) the LLT has been confirmed and it currently has an associated observation —which implies that the detection has been correctly performed according to what is above stated; and (ii) the target is at least partially within the ROI. High-level trackers are instantiated as *entering*, except when they come from a group that have split —in this case they are *appearing*. Entering status last until they completely lie within the

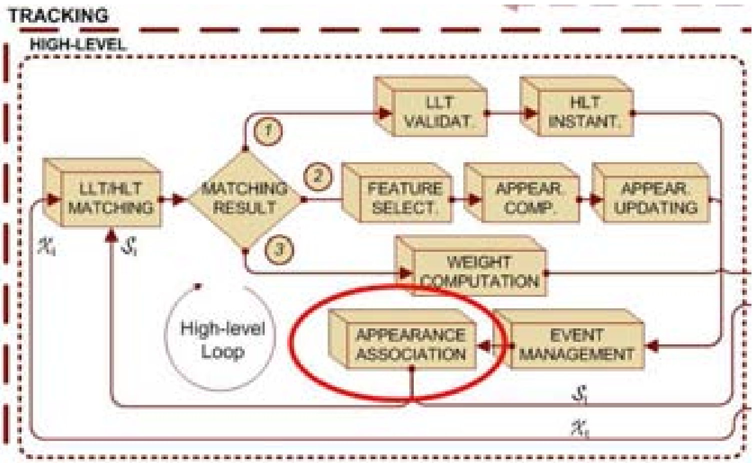


Figure 5.47: Appearance-association module.

ROI. Appearing status last a pre-defined number of frames in which the HLT had confidence enough to be updated.

When a part of the target is partially outside the ROI and the security border, the target is marked as *exiting*. The target can now either return to the ROI, or lie completely outside the area defined by the ROI and the security border. The latter implies the tracker removal. Trackers are also removed if they are partially within the outer zone and they are being tracked by a low-confidence ABT, thereby avoiding a senseless gradient-based search when the target has actually exited.

5.5.8 Appearance Association

Low-level trackers lose their track during long-term segmentation failures, such as background camouflage, target grouping and occlusions. Once the particular event is over, the target is again detected, and a new LLT is instantiated. When this track become stable, the LLT is confirmed and a HLT is created. The former HLT state was estimated according to ABT, and the appearance models were updated when the localisation was reliable enough. A tracker association process can now be performed, and the system should conclude that both trackers are in fact representing the same target, see Fig. 5.47.

This is done as follows: first, a potential tracker association matrix is built upon several state criteria which are detailed in the following. Subsequently, candidates are filtered according to their shape. Then, they are gated by appearance using the multiple smoothed appearance models and similarity statistics previously learnt. Finally, the tracker association process is performed.

Thus, the potential association matrix is built according to a set of rules. These define which HLT can be selected as candidates, and which HLT require an association process. On the one hand, the formers must be new appearing HLT, which track single targets, with an associated LLT still confirmed. On the other, among the formers, no

HLT tracking a group is considered, and they must have no LLT associated; further they are discarded if the ABT is being successfully performed —which implies that there is enough confidence to perform model updating— and they do not collide with the candidate.

Shape gating is performed according to the innovation covariance matrix of the candidate corresponding LLT. Thus, the lost HLT must lie within an ellipsoid defined by the uncertainties on axes lengths and orientation.

Finally, the similarity between the histograms of each feature of the candidate and the tracker are evaluated using the computed statistics of the above-defined metric as in Eq. (5.50) learnt for the former HLT. Those histogram corresponding to the former tracker are in fact smoothed models computed while the segmentation was reliable. However, since feature selection depends on the local environment, and the targets move while they are grouped, the feature pool is subject to changes —whenever the appearance models were successfully updated during the mean-shift procedure execution. Then, in case that non-coincident features are present, new histograms are computed for the candidate.

Thus, features are gated, and the resulting mean distance is used to perform the association between former high-level trackers and candidates according to a nearest-neighbour criteria in the appearance sense. The association process ended by updating the interaction status taking into account the new situation.

If none of the candidate trackers is within the gate of the former one, this is still considered lost, and a new association process is applied at the next time step.

5.5.9 Some Significant Top-down Pathways

At is has been shown, the proposed architecture follows an attentive approach by analysing the events in which the HLT's are involved. According to these, the two main presented actions include a principled target-appearance modelling, and the selection of the operation mode.

Further, an attentive approach is also used to feed back the LLT tier, by validating the observation gating process, and by allowing multiple LLT correspondence to one HLT. This process is done as follows.

5.5.9.1 High-level Validation of Gated Observation.

Observations were validated for a given LLT according to the result of comparing the Mahalanobis Squared Distance between the predicted observation and the actual one, and the Mahalanobis radius given by the covariance ellipsoid at a given confidence.

Observations were then associated according to a nearest neighbour approach. This suffices to disambiguate among the different targets, since the only ambiguity appears when two targets are close enough. However, in this case a single blob corresponding to the group is obtained.

Nevertheless, problems arise between the group LLT and the target ones. If the group observation may also correspond to the target state —which happens in case of complete occlusions, for instance— the observation is validated for both LLT. This

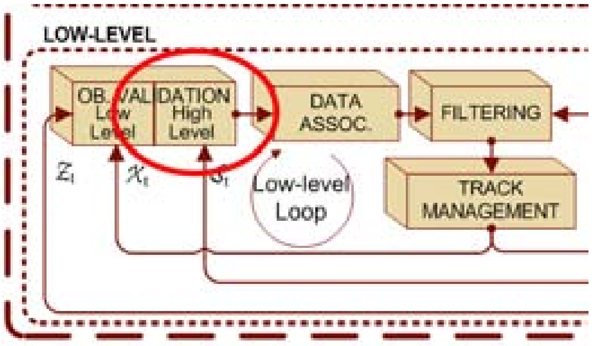


Figure 5.48: High-level observation-validation module.

Algorithm 6 High-level validation of gated observations.

```
1. Derive if the observation is over a collision area,
2. compute the number of targets in the support map,
3. **if** there are several targets, **then**:
    (a) search for a group in the HLT's corresponding to LLT's with validated observations
    (b) **if** a group is found, **then**:
        i. compare the number of group partners and the number of targets in the support,
        ii. **if** this matches, **then** validate the observation just for the LLT associated to the group HLT
4. **else**, **then** prevent the validation of the observation for the LLT associated with the group HLT.
```

issue is even more noticeable due to the fact that the covariance ellipsoid of the grouped target is expanding while no observation is associated.

Thus, a principled validations of observations based on the inferred events is required, see Fig. 5.48. The following approach is used: every observation associated to an existing LLT is analysed according to the collision map. If the observations lies over a collision area, the number of targets in the corresponding *support map*[13] is

[13] This is a binary map which keeps the record or target occupancy for each scene pixel. For instance, if a certain pixel has at the support map a corresponding value of 0110, it means that both targets two and three are over that particular pixel.

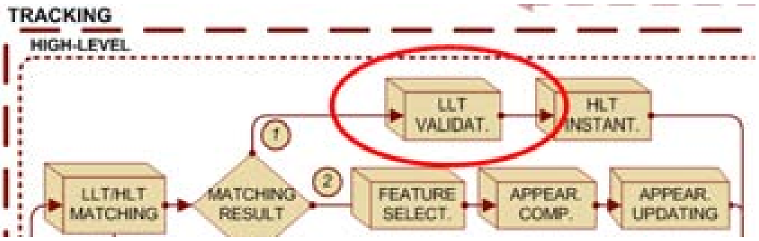


Figure 5.49: LLT validation module.

computed. Then, two actions can be taken: on the one hand, if more than one target is found, the LLT for which the observations are validates, are examined. If any of them tracks a group, the number of group partners is compared with the number of targets in the support map. In case this matches, the observation is validated just for the LLT associated with the group.

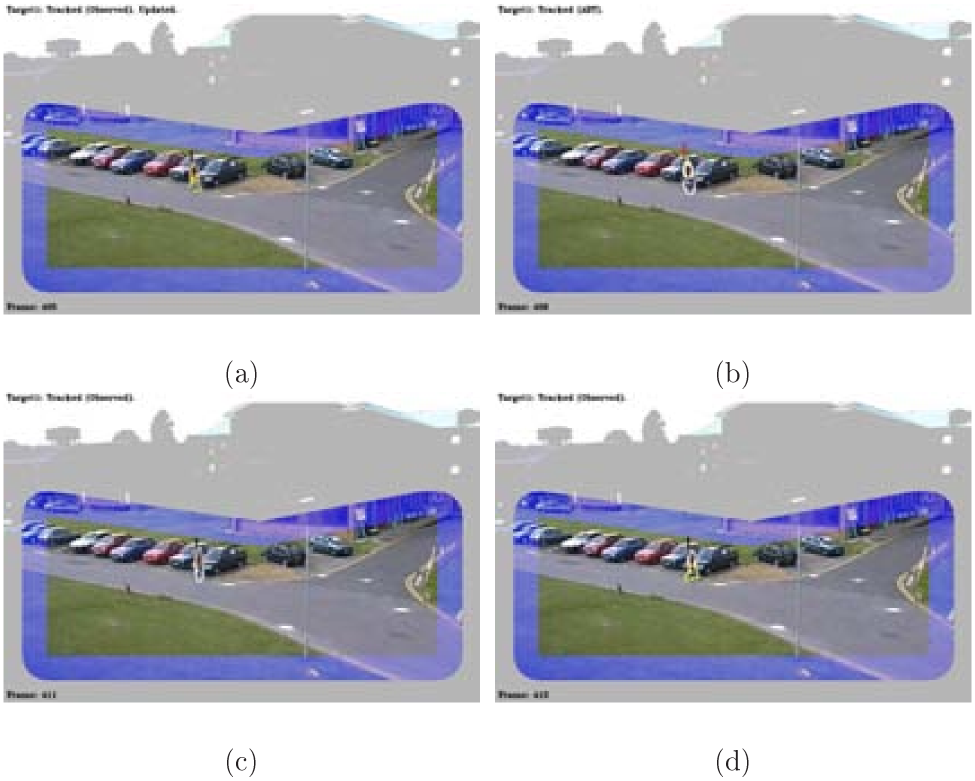
On the other hand, if just one target or less lie in the support map, it is concluded that the observation cannot be validated for the LLT of the group, despite being within its covariance ellipsoid. This procedure is summarised in Algorithm 6.

5.5.9.2 Multiple LLT Correspondence

LLT's represent intermediate entities that require being associated to a HLT, once they have being ever confirmed as stable. Thus, every time a non associated LLT is confirmed the following process take place: first, its position in the scene is evaluated. If it is validated according to the scene criteria set in section 5.5.7, the possible constitution of a group is considered according to the number of targets in the support map, whether they are enclosed in the observation area, and these targets actually collide. Then, either an HLT representing a group is instantiated, or the LLT corresponds to an appearing target requires a new target HLT.

However, given the frequent segmentation errors due to changing environmental conditions, another case has to be taken into account. Due to significant camouflage problems, a LLT may be instantiated on a target already tracked —if the observation lie beyond the confidence ellipsoid. The new and the former LLT's can then compete for the observations, and eventually the second LLT may be confirmed and requests for and HLT.

In order to avoid that a second HLT is instantiated for a same target, an LLT validation process is performed, see Fig. 5.49. Existing LLT's are gated using the innovation covariance matrix of the just confirmed one, according to Eq. (5.29). In case that another LLT lie within the defined ellipsoid, and no other HLT overlap with this region, the new LLT is assigned to the HLT of that LLT. Both LLT's coexists, but just one HLT is created, see Fig. 5.50.


Target1: Tracked (Observed). Updated.
Target1: Tracked (ABT).
Target1: Tracked (Observed).
Target1: Tracked (Observed).

(a) (b)

(c) (d)

Figure 5.50: Sample tracking showing multiple LLT correspondence on a *PETS* sequence. (a) The associated LLT —in white— receives an observation, and the HLT —in black— is consequently updated —as shown in the box; (b) A wrong segmentation —in yellow— due to camouflage causes the instantiation of a new LLT; the HLT is tracked in ABT mode; (c) The second LLT is confirmed and associated to the *same* HLT, thanks to the LLT-validation module; (d) the first LLT is again confirmed once the problem is over, and the HLT recovers it.

5.6 Discussion

In this chapter —the main one— a principled and structured system is presented in an attempt to take a step towards solving the numerous difficulties which appear in unconstrained tracking applications. The system here proposed implements a hierarchical but collaborative architecture, in which each level is composed of several modules which are devoted to specific tasks. These are performed by particular algorithms, but they can be substituted by any enhanced one without modifying the architecture itself. Therefore, albeit the different modules have been here developed or improved, we consider the architecture itself as the main contribution: it introduces the synergies between the algorithms which permit to tackle a problem with such an inherent complexity.

This structured framework combines in a principled way both bottom-up and top-down tracking approaches: each level feeds the higher one with its computed results, and is itself fed back with high-level results. In this way, by taking advantage of both approaches, the system is allowed to benefit from bottom-up capabilities, such as simultaneous modelling and tracking without making used of a-priori knowledge; but also, high-level analysis is performed, granting accurately tuned models, and proper operation-mode selection. In addition, each level has an internal loop which also provides the system with adaptive capabilities by updating the background model, making use of the knowledge about existing tracks, or selecting the most appropriate approach according to the events in which the targets are involved.

In this way, the proposed approach follows the natural paradigm, where visual-stimuli analysis is performed by the combination of pre-attentive and attentive processes. Further, it makes use of first-order and second-order motion perception.

A principled event management is proposed and embedded in the architecture. This provides a valuable knowledge in order to obtain high-level scene descriptions, while allowing the system to switch among different operation modes. The latter is crucial to achieve successful performances, since non-supervised MTT is a complex task which demands different approaches according to different situations.

This remarkable characteristic of the system in managing multiple interactions among several targets leads to another important contribution. This focuses on tracking several targets independently while they are grouped, thereby yielding an accurate and robust target localisation. Thus, feature-selection and appearance-computation modules have been developed, by paying special attention to the particular characteristics of grouping situations. Features are selected considering not only the best distinction between the local background and the target, but also between the target and its group partners.

A model pool is built, and long-run features are kept and smoothed. These features are useful after a target loss caused by occlusion, grouping or camouflage events to recover the target. Further, by smoothing the histograms the representation is less sensitive to potential initialisation and subsequent localisation errors. Then, a second operation mode, an ABT, is added to tackle those events which prevent from a proper segmentation. Motion and appearance cues, relative to potential distracters, are taken into account when performing a gradient search. A principled model updating scheme is followed to avoid model drift.

Thus, the proposed architecture successfully tracks multiple targets simultaneously —as shown in the chapter devoted to experimental results. This is achieved even in hard conditions of cluttered background and uncontrolled illumination. Targets present a high appearance and shape variability. Complex tracking events —in which numerous targets are simultaneously involved in different grouping and splitting situations— take place. In spite of these difficulties, experiments on complex indoor and outdoor scenarios have yielded robust and accurate results, thereby demonstrating the system ability to deal with unconstrained and dynamic scenes. No a-priori knowledge about either the scene or the targets, based on a previous training period, is required. The method is adaptive in the sense of number of targets, the best appearance representation, or the most appropriate tracking algorithm according to the events which are taking place.

Still, many cases remain in which no positive discrimination is obtain between the background and a particular target. Thus, target segmentation can be enhanced by making use of new cues. For instance, gradient-change detection can be used to attenuate target camouflage. Further, shadow removal techniques could be very useful to address those false detections due to changes in the illuminant chrominance. This can be carried out by considering the techniques proposed in [21].

Further, a multi-layered background can be built by including characteristics of left objects. Therefore, motion segmentation of new targets over former ones could be achieved, while ghost detection —in the event that the object be again removed— is avoided.

A target classification module —which requires a-priori learnt knowledge— could distinguish among people, vehicles and other objects in motion. This will also help to segment targets who enter the scene within a group.

Target representation can be refined by including structure components —such as body-part histograms— and shape cues —such as SIFT descriptors. This will enhance agent tracking during long-term occlusions.

Finally, the system may benefit from high-level information about the context and current situations provided by cognitive levels of the HSE framework, while making use of multiple active cameras from several point of views. Further, learning methods can be considered to tune algorithm parameters according to the particular conditions of a given scenario.

5.7 Resum

S'ha presentat un sistema estructurat per tractar les nombroses dificultats que apareixen en aplicacions genèriques de seguiment. El sistema que aquí es proposa implementa una arquitectura jeràrquica col·laborativa, en la qual cada nivell es compon d'uns quants mòduls que estan dedicats a tasques específiques. Per això, per bé que els diferents mòduls han estat aquí desenvolupats o millorats, considerem l'arquitectura mateixa com la contribució principal de la Tesi, ja que defineix les sinergies entre algoritmes que permeten tractar un problema amb una complexitat tan inherent.

Aquesta estructura combina enfocaments tant de Baix-a-Dalt com de Dalt-a-Baix: cada nivell alimenta de més alts amb els seus resultats, i és alimentat amb els resultats dels nivells més alts. El sistema permet beneficiar-se de les capacitats de Baix-a-Dalt, com el modelatge i el seguiment simultani sense utilitzar coneixement a-priori; però també es realitza un anàlisi d'alt nivell, generant models refinats més acuradament, garantint així la selecció del model de funcionament més adient. A més a més, cada nivell té un bucle intern que també dóna capacitats al sistema d'adaptació.

S'ha proposat una gestió d'esdeveniments que s'ha incrustat en l'arquitectura. Això proporciona un coneixement valuós per obtenir descripcions d'escena d'alt nivell, mentre deixa el sistema canviar diferents modes d'operació.

Aquesta característica del sistema per gestionar les interaccions entre els objectes ha portat a una altra contribució important. Aquesta se centra en seguir independentment múltiples objectes mentre s'agrupen, produint així una localització d'aquests més acurada i robusta. Així, s'han desenvolupat uns mòduls per al càlcul de l'aparença i uns algorismes per a la selecció de les millors característiques. Aquestes característiques se seleccionen considerant no només la millor distinció entre el fons i l'objecte, sinó també entre l'objecte i els seus companys de grup.

S'ha construït un model on les millors característiques a llarg terme s'enmagatzemen i se suavitzen. Aquestes característiques seran útils per recobrar l'objecte després d'una pèrdua provocada per oclusions, agrupacions o camuflament. A més, amb la suavització dels histogrames, les representacions dels objectes són menys sensibles a errors potencials de localització i inicialització. Llavors, un segon model d'operació, l'ABT, s'afegeix al sistema per tractar aquells errors degut a la falta d'una bona segmentació. Així es tenen en compte informació de moviment i d'aparença dels potencials distractors, que són tinguts en compte per a realitzar el descens del gradient. A continuació se segueix amb un esquema d'actualització del model per evitar la deriva del model.

Així, l'arquitectura proposada segueix satisfactòriament múltiples objectes simultàniament. Això s'aconsegueix fins i tot en condicions fortes de fons sorollós i d'il·luminació no controlada. En estos casos els objectes presenten una variabilitat molt gran d'aparença. Llavors, els esdeveniment de seguiment que es produeixen són molt complexos de tractar. Malgrat aquestes dificultats, s'han realitzat experiments en entorns complexos, tant tancats com oberts, que han donat uns resultats robustos i acurats, demostrant així l'habilitat de sistema per tractar escenes no restringides i dinàmiques. Tampoc s'exigeix cap tipus de coneixement a-priori ni sobre de l'escena ni sobre els objectes, que estiguin basats en un període d'entrenament previ. El mètode és adaptatiu al número d'objectes, a la millor representació d'aparença, i a l'algoritme escollit més apropiat que millor segueixi, segons els esdeveniments que estan tenint lloc.

Chapter 6

Experimental Results

The tracking task requires reasoning over time under uncertainty. This uncertainty involves not only probabilities about some event or condition, but degrees of truth about them. Given the practical and theoretical ignorance about all the involved processes, it is not possible to have access to a ground truth about what is taking place in the real world. Further, all human assessments of a particular situation entails an important subjective component, thereby presenting significant deviations among them[1]. However, it is often assumed that a human visual determination, or the juxtaposition of multiple ones, provides a error-free ground truth.

MTT applications usually imply real-time requirements, in conjunction with extreme robust performances. Hence, algorithms should be flexible enough in order to deal with unexpected situations. Therefore, the considered assumptions should be kept to minimum. However, accuracy requirements can be relaxed in comparison with applications concerned with action, gesture or facial expression recognition, for example.

In the following, some considerations on tracking performances are stated. Subsequently, numerous experimental results in several scenarios are presented, and the performance of the different algorithms analysed according to various criteria.

6.1 On Tracking Performance

Real-time processing, extreme robust performances, high accuracy, low power, or low cost may be critical for the application purpose. Unfortunately, a trade-off must usually be found among these requirements: enhancing the system accuracy and robustness often implies increasing the algorithm complexity, and thereby the computation time.

[1] Another way of mitigating the inherent human subjectivity consists on testing the tracking algorithms on synthetic sequences about virtual environments. Synthetic data allow us to achieve two goals: first, access to the ground truth is granted, and therefore deviations and performances can be accurately measured; and secondly, experiment conditions can get harder on each aspect independently, and thereby maximum performances can be measured.

A differentiation is here made about practical requirements —which may depend on the budget, or may change as the technology make progress— and the evaluation of the obtained results. In the opinion of the author, research should not be restricted by the technology state of the art[2].

6.1.1 Achieving Real-time Performances

MTT in unconstrained and dynamic scenarios are one of the most computationally demanding Computer Vision topic. Nevertheless, real-time performances may be achieved without excessively compromising accuracy and robustness by placing special care in three main tasks, namely, specific hardware implementation[3], code optimisation, and algorithm designing.

Thanks to a large number of recent technological advances in the hardware domain, image-sequence capture and transfer is already feasible in real-time[4]. Hence, image processing remains in many application as the only bottleneck [73]. Other bottlenecks may appear in case of needing to store of visualise processed images. Large-scale non-volatile storage can be mandatory[5].

Nevertheless, significant speed improvements can be achieved by processing pixel-wise operations in parallel. Many systems can benefit from specific hardware implementations. Among these, Application Specific Integrated Circuits (ASIC), Digital Signal Processors (DSP), Graphic Processors Units (GPU), and Fully Programmable Arrays (FPGA) can be considered.

Systems present in the related literature are often prototypes. Robustness and accuracy in unconstrained conditions requires capabilities to switch among different operation modes and algorithms [60]. However, cost requirements imply keeping the hardware viable. Albeit ASICs provide better speed performances, low power and low cost, they preclude future developments. On the other hand, DSPs allow programmable architectures at the expense of higher individual cost and power consumption. GPUs offer the possibility of offload specific processes, thereby speeding up low-level algorithms at economical cost. Finally, FPGAs provide large-scale parallel processing, efficient pipe-lining, and high I/O capabilities, which support simultaneous access to multiple external memory banks.

It should be remarked that the computational load at any time t depends on particular issues which cannot be controlled, such as the number of targets within the scene, and the size of these targets or the scene itself in number of pixels. It also depends on design decisions which may be critical to achieve successful performances

[2] However, it should be close enough in order to allow practical applications in a near future.

[3] Enhanced performances via hardware can also be obtained by making use of more powerful computers, overclocking them, or including dual- and quad-core processors, for example.

[4] For example, at the time this thesis was written, Giga-Ethernet cameras provide high resolution and frame rate in progressive colour acquisition modes.

[5] Again, current fastest hard disks rotate at 15,000 rpm, and data transfer is limited to a maximum of 110 MB/s. Thin Film Transistor Liquid Crystal Display (TFT LCD) monitors are experimenting an important development, and the newest ones reach response times of 1ms.

—like the number of histogram bins, or the number of features selected in the proposed tracking architecture.

Further processing speed improvements can be achieved by optimising the code. This specifically means to modify the code and its compilation settings on a given computer architecture to produce more efficient software. Performance bottlenecks are often due to language limitations rather than algorithms or data structures used in the program. Low-level languages which gives more direct access to the underlying machine allow faster computation as the expense of less readability and maintainability. An special case are interpreted languages. These are executed from source form, and are consequently slower. However, the code is often more flexible, allowing a faster prototyping. Finally, a remark must be said against *premature optimisation*[6], which describes a situation where a programmer lets performance considerations affect the design of a piece of code.

Finally, the system can be speed up by implementing asymptotically optimal algorithms, that is, those which for large inputs performs at worst a constant factor worse than the best possible algorithm, thereby allowing algorithm scalability. In this case, the performance is evaluated in the sense of time complexity, i.e, the number of steps that it takes to solve an instance of the problem as a function of the size of the input.

6.1.2 Evaluating Accuracy and Robustness

This field still being a *novel* open research line, there is unfortunately a lack of widely accepted test data-bases, ground-truth data, and evaluation criteria. Performance evaluations are often based on quantitative metrics which depends on qualitative events, or even results are evaluated by means of visual inspection. In order to allow algorithm comparisons, a standard methodology for performance evaluation must still be developed and assumed. Public test sequences should be synchronised and calibrated, and ground truth data must be available.

At least, some efforts have been made in both issues. Several workshops on Performance Evaluation of Tracking and Surveillance[7] (PETS) have taken place since 2000. Data sets are provided in order to allow algorithm performance comparison. In particular, PETS 2001 Test Case Scenario has been widely used by the community since its release. It contains three different views of an outdoor scenario which includes roads, parking places, and green areas surrounding several buildings. The resolution is PAL standard: 768 x 576 pixels, at 25 frames per second (fps). Files are compressed in low quality JPEG, thereby presenting many visual artifacts.

CAVIAR (Context Aware Vision using Image-based Active Recognition) database[8] has been used in PETS 2004. It contains indoor sequences corresponding to two different data sets. The first one were filmed with a wide angle camera lens in an entrance lobby. The second one was recorded in a mall centre corridor from two different point of views. In both cases, the resolution is half-PAL standard (384 x 288

[6] Tony Hoare —the quick-sort designer— and Donald Knuth —who can be considered the father of algorithm analysis— repeatedly warned against this practice.

[7] http://peipa.essex.ac.uk/ipa/pix/pets

[8] http://homepages.inf.ed.ac.uk/rbf/CAVIAR

pixels @ 25 fps). Sequences are synchronised, calibrated, and some ground truth data representation is available.

The SCEPTRE project[9] (Service to Evaluate the Performance of Tracking and Recognition) provides two data sets of football matches from eight viewpoints each. The resolution is 720 x 576 at a frame rate of 25 fps. Many others performance-evaluation workshops and projects have recently provided a wide diversity of datasets, such as CLEAR 2006 and 2007 (Classification of Events, Activities and Relationships[10]), and the AMI project (Augmented Multi-Party Interactions[11]).

Despite this effort, most results are given as samples of a small number of processed frames, and there is still a lack of accepted performance criteria. Nevertheless, an increasing number of authors are proposing performance measures in recent times. Thus, Senior et al. [81] compute a set of error measures between the tracking results and a ground truth determined by a human user. The performance is evaluated according to:

1. the centroid position error;
2. the bounding-box area error;
3. the object detection lag;
4. the track incompleteness, which is given by the rate of number of frames missing from the result track plus the number of frames erroneously associated by the common number of frames between results and ground truth;
5. false positive and negative track error rates;
6. and, the number of object classification errors.

Zhao and Nevatia [99] evaluate their algorithm performance without the need of an accurate ground truth. Thus, this is done according to the following measures:

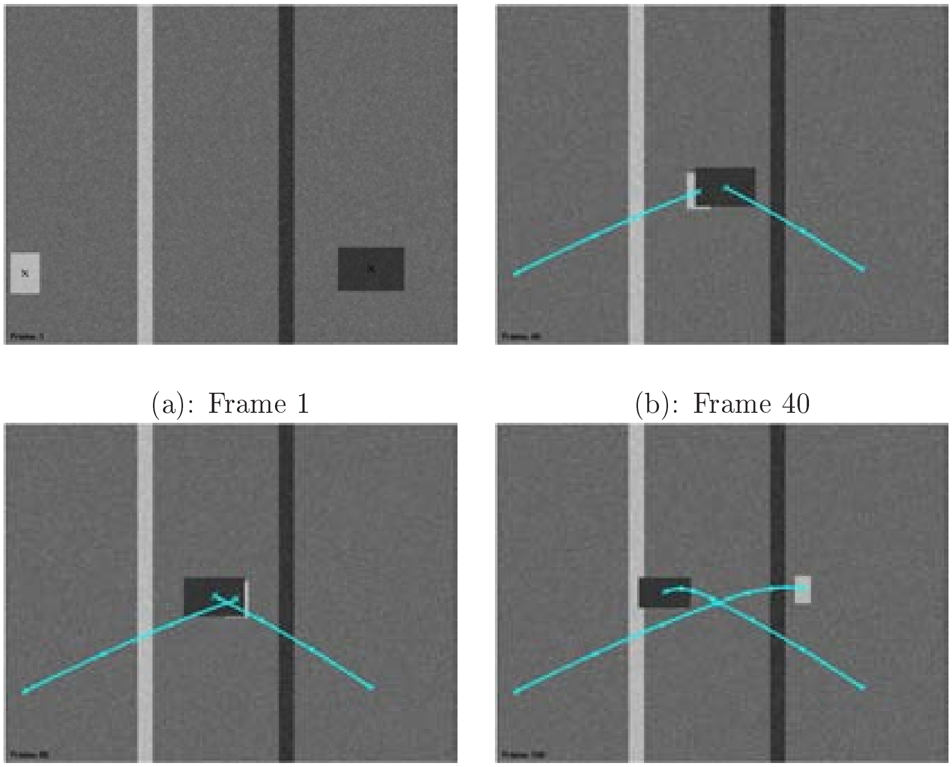
1. the trajectory-based error rate, given by the number of times an identification is broken, and the number of objects;
2. the detection lag;
3. the detection rate;
4. and, the false alarm rate.

Event-based error measures have also been proposed [72]. In these, events reported by the system are compared to reference ones. Thus, cumulative counts of specific event types of different orders are used to perform an evaluation on several sub-sequences. Low relative errors at lower orders imply good responses for those event of interest, while low errors at higher orders imply also good track continuity.

[9] http://sceptre.king.ac.uk/sceptre/default.html

[10] http://www.clear-evaluation.org/

[11] http://corpus.amiproject.org/



(a): Frame 1

(b): Frame 40

(c): Frame 50

(d): Frame 100

Figure 6.1: Example of same ground-truth frames for a given scenario.

Finally, the ETISEO project[12] (Evaluation du Traitement et de l'Interprétation de Séquences Vidéo) proposes a data structure for content annotation, video annotation rules, and a set of metric definitions.

Summarising, there is still a lack of widely accepted test data-bases. Performance evaluations are often quantitative comparisons using qualitative metrics, or even results are evaluated by means of visual inspection —given a set of sample frames— and usually no ground-truth data is available. A standard methodology for evaluating performances is mandatory.

6.2 Evaluating the Performance of the Particle-Filter Approach.

The performance of the algorithm has been tested using both synthetic and real data. A series of synthetic experiments has been designed in order to evaluate the

[12] http://www.etiseo.net/

Mean number of samples per target		
	Target 1	Target 2
Run 1	49.5101	50.4899
Run 2	49.6577	50.3423
Run 3	50.4195	49.5805
Run 4	49.9866	50.0134
Run 5	50.3456	49.6544
Run 6	50.0705	49.9295

Table 6.1: Results of the proposed weight-normalisation approach.

Mean normalised error		
	Target 1	Target 2
Run 1	0.1163	0.1309
Run 2	3.8864	0.1182
Run 3	0.1222	0.1226
Run 4	0.0980	0.1038
Run 5	0.1612	0.1131
Run 6	0.1101	2.4679
Mean*	0.1216	0.1177

(a) Performance without regularisation and speed feedback

Mean normalised error		
	Target 1	Target 2
Run 1	0.0715	0.0716
Run 2	0.0849	0.1163
Run 3	0.0987	0.1289
Run 4	0.0645	0.0595
Run 5	0.0679	0.1173
Run 6	0.1233	0.0840
Mean*	0.0851	0.0963

(b) Performance using the proposed regularisation and speed feedback

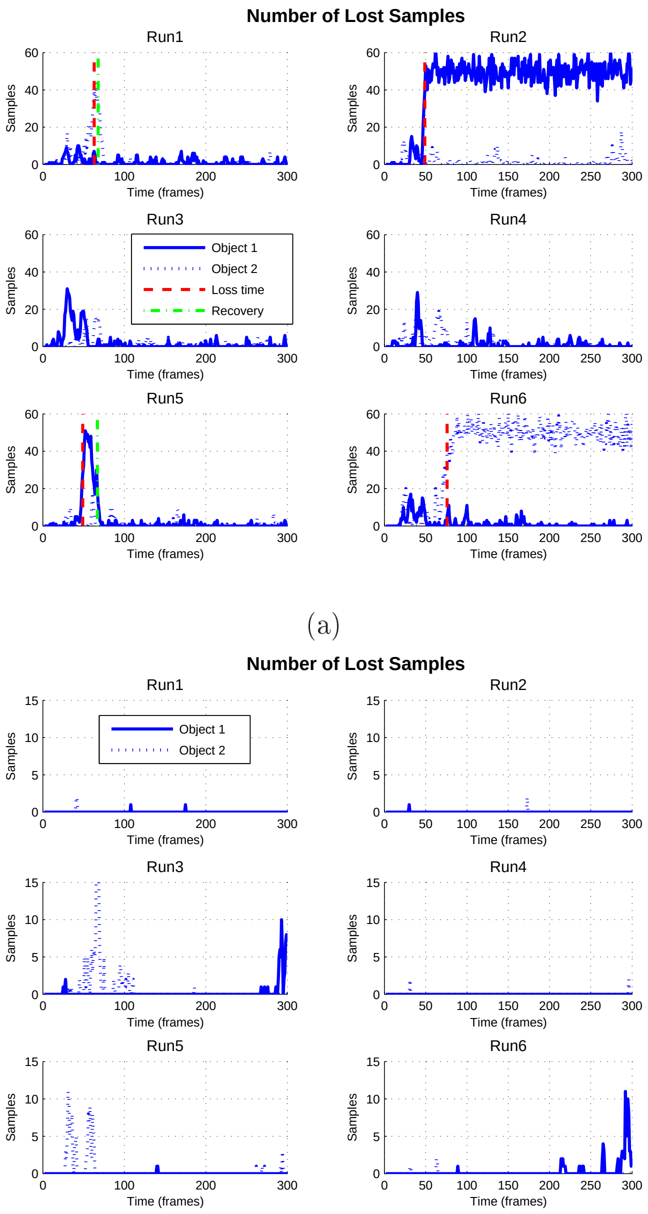
* The mean is computed just for those non-lost targets. Thus, in Table (a) the second run for target 1, and the sixth one for target 2 are not taken into account. Otherwise, even a higher difference would have been yield.

Table 6.2: Mean normalised error

performance of the different design improvements. They cover several difficulties a tracker can run into, see Fig 6.1. Thus, the scenario implies an experiment in which two moving targets with highly non-linear dynamics are considered. Both target size and aspect ratio change over time. They move through a scene with complex clutter. Two strips are drawn in the background. Their distributions are identical to both targets' distributions, thereby mimicking them. Strong acquisition device noise is simulated. The targets are in different planes but their image trajectories cross over causing a complete occlusion through several frames. Tracking is performed over $T = 300$ frames using $N = 100$ samples.

As it have been stated in Chapter 4, no detection is ever used after the initialisation. Thus, targets are tracked by means of prediction and weighting the different hypotheses, while the image is not scanned by performing any motion segmentation.

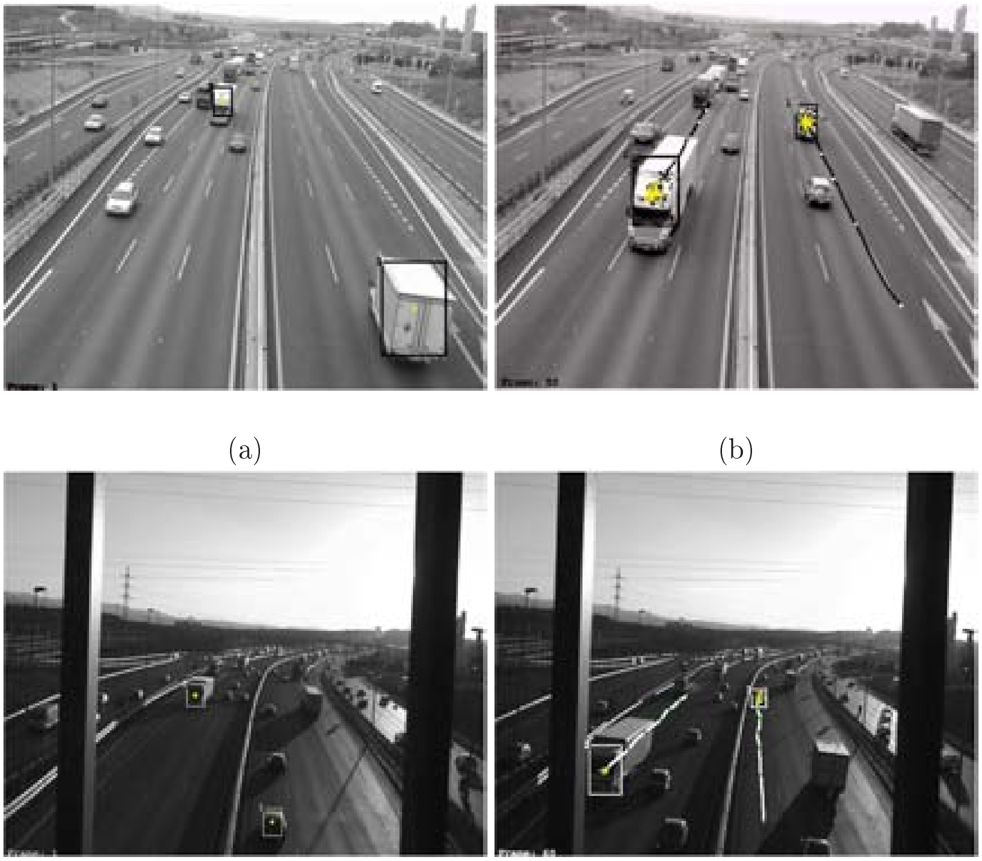
Numerous runs have been carried out with and without the proposed weight



(a)

(b)

Figure 6.2: Number of lost samples. (a) without the regularisation and speed feedback; (b) using the proposed regularisation and speed feedback. (Notice the reduction of 75% in the scale of the axes)



(a) (b)

(c) (d)

Figure 6.3: Target performance on traffic sequences

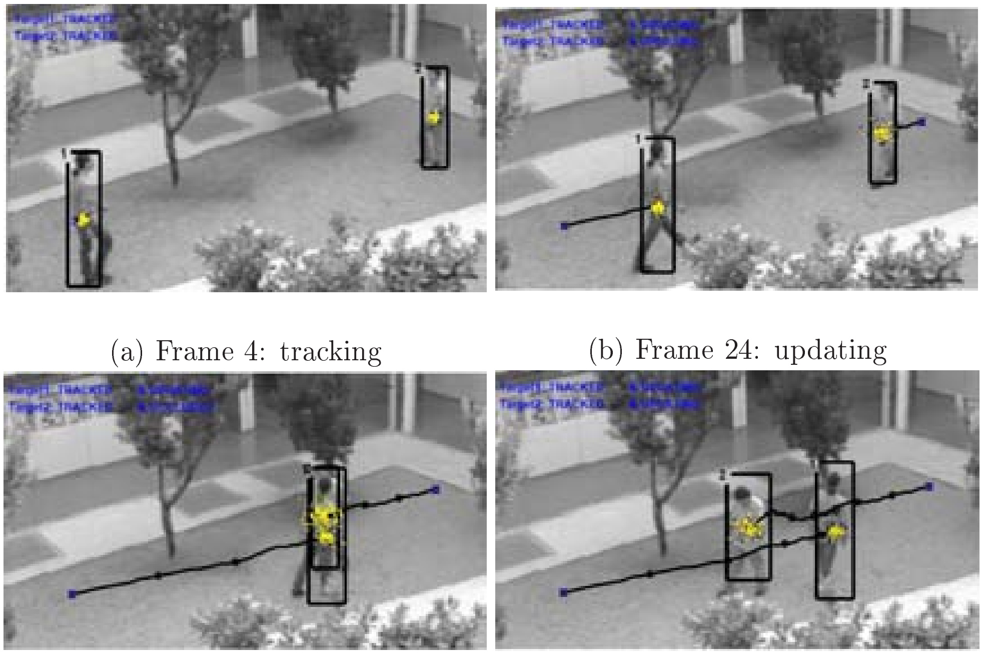
normalisation[13]. In case of no-using it, one target is lost due to the lack of samples in five of the runs. In the remaining one, at time $t = 300$ one target got 92 out of 100 samples. A target is considered lost when the normalised Euclidean distance, according to the target's size, between the target and the estimation position is higher than a threshold set at 0.5, that is, the overlapping between the sample and the real target is reduced a half of its area[14].

After the proposed weight normalisation, the mean number of samples per agent fluctuates between 49.5 % and 50.5%, as seen in Table 6.1.

Multiple runs have been performed to test the effect of regularising both position

[13] Here results are presented for just six runs. This is however enough —in our opinion— since the tracker is dealing with a synthetic scenario with all its parameters fixed.

[14] Due to the lack of a standard about this issue, we have considered convenient to establish such a strict criterion.



(a) Frame 4: tracking

(b) Frame 24: updating

(c) Frame 80: occluded

(d) Frame 140: recovery

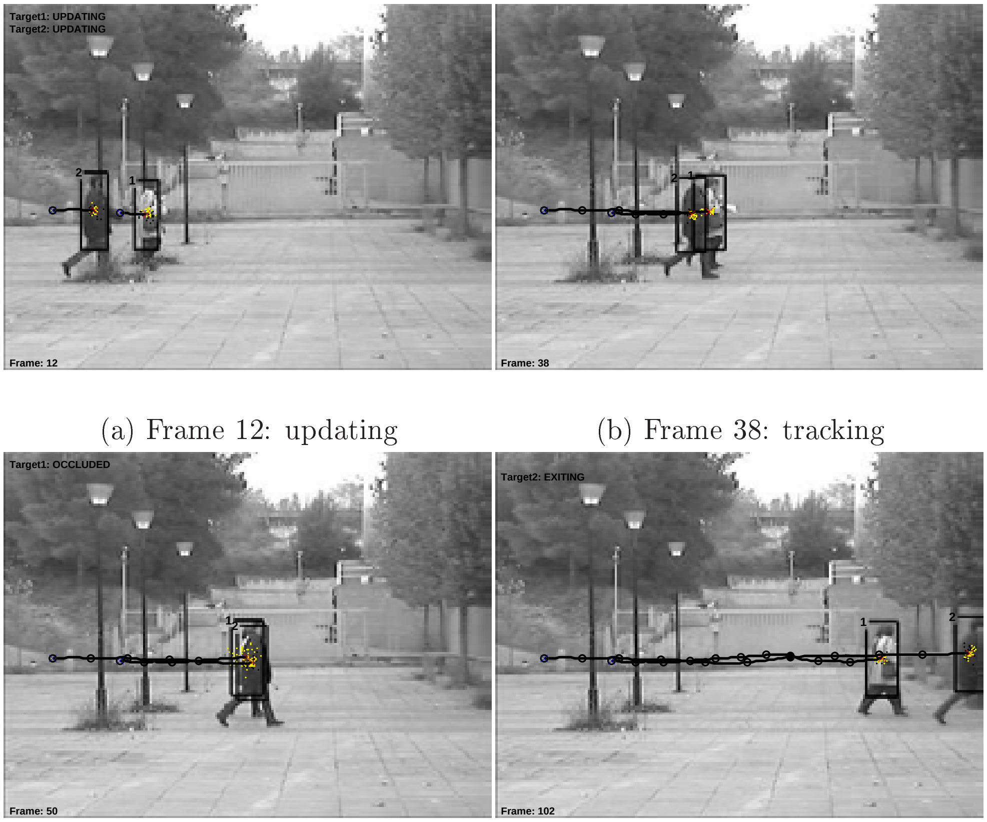
Figure 6.4: Experiment involving an opposite translation and merging.

and speed, not estimating the target speed from the speed of the samples, and feeding back the estimated speed into the prediction stage. Thus, Table 6.2.(a) shows the mean normalised error —according to the target size— in estimating the target position without the regularisation, while Table. 6.2.(b) shows the same results after applying it. A significant error reduction can be appreciated.

Further, Figs. 6.2.(a) and (b) compare the number of samples per target that have lost the target. After considering the regularisation, a significant sample loss reduction is observed: the number of lost samples is negligible, except for specific instants in which the target is over clutter, see *run 3* in Fig. 6.2.(b). In addition, none of the targets is ever lost, since the effective number of samples has been increased avoiding sample wastage. The trajectory jitter is considerably reduced.

In the next, both particle filter approaches —using appearance models based on intensity templates, and based on colour histograms— are tested on real sequences. Two hundred samples have been used in all the analysed sequences. Trackers are initialised by hand.

Fig. 6.3 shows results using the template approach in traffic sequences taken in a motorway during 60 frames. Figs. 6.3.(a), 6.3.(b) show results where large size and speed changes are present, as they can be noticed according to the milestones and the bounding box sizes; Figs. 6.3.(c), 6.3.(d) exhibit tracker performance under heavy shadow and reflectance conditions.



(a) Frame 12: updating

(b) Frame 38: tracking

(b) Frame 50: occluded

(c) Frame 102: recovery and exiting

Figure 6.5: Experiment involving an overtaking

The performance of the grey-scale template-based algorithm has also been tested using several sequences involving humans. Two targets are tracked simultaneously, despite their being articulated and elastic objects whose dynamics are highly nonlinear, and that move through an environment with complex clutter.

The first sequence involves an opposite translation and merging. Both targets start moving from opposite positions and meet near the second actor's initial position. In this case 120 images of 320 x 240 have been analysed. The number of samples is also fixed at 200, and trackers are manually initialised.

The first target's speed decreases unevenly from five pixels per frame and the second one from two pixels per frame to nearly zero during the first part of the sequence. The first target is almost completely still from frames 70 to 130, occluding the second target. The latter crosses at a very low speed while performing a rotation. Thus, significant speed, size and appearance changes can be observed. The background intensity levels are so similar to the target ones that constitute a source of clutter.

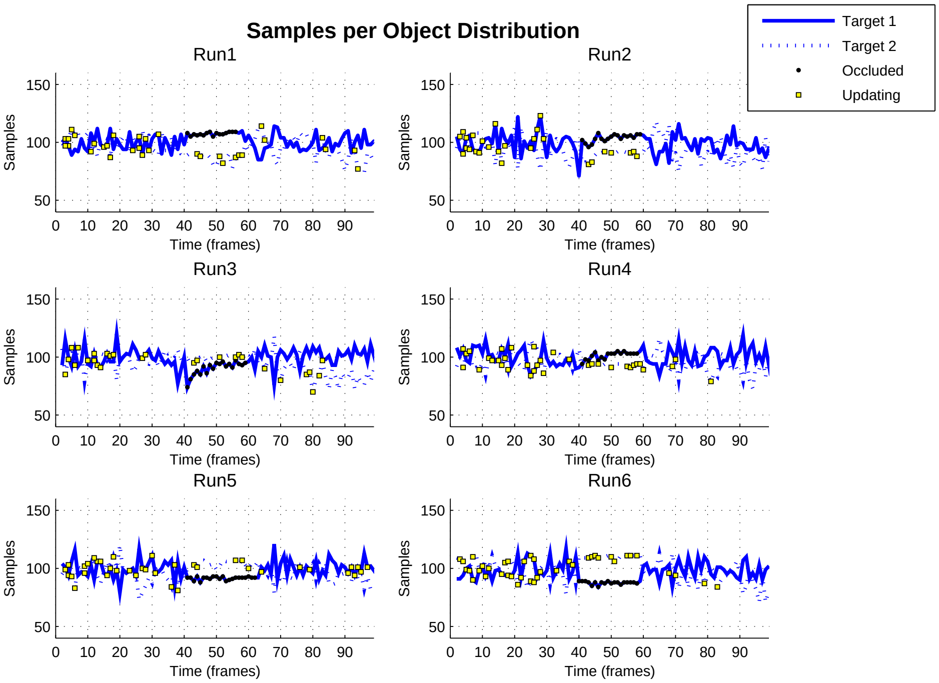


Figure 6.6: Sample distribution in the overtaking sequence

The tracker performance is shown in Fig. 6.4. Both targets' appearance models are updated when reliable measures are obtained, see Fig. 6.4.(b). Occlusion is correctly detected avoiding re-sampling of samples of the occluded target and erroneous dynamic and appearance models updating, see Fig. 6.4.(c). The tracker recovers from occlusion, see Fig. 6.4.(d).

The second sequence involves an overtaking, see Fig. 6.5. This sequence have 130 images of 384 x 288. Two hundred samples have been also chosen to track both targets. Trackers are also initialised by hand.

The second target moves faster than the first one —which is in fact a group of two people— overtaking it. An almost complete occlusion can be observed from frame 40 to 60, see Fig. 6.5.(b), Fig. 6.5.(c). The street-lamps constitute a source of clutter and cause partial occlusions to both targets, see Fig. 6.5.(a).

In the following paragraphs, quantitative data concerning the overtaking sequence are presented. In this way, the algorithm robustness can be discussed and the drawbacks exposed, as well as the ways of solution. Results concerning six runs are presented.

Fig. 6.6 shows the sample distribution among the targets present within the scene. Occlusion and appearance-model updating situations are also pointed out. As can be observed, the sample are evenly distributed. Thus, the number of samples per object fluctuates around the fifty percent, given that two objects are tracked. During occlusions, the samples corresponding to the occluded target are not re-sampled.

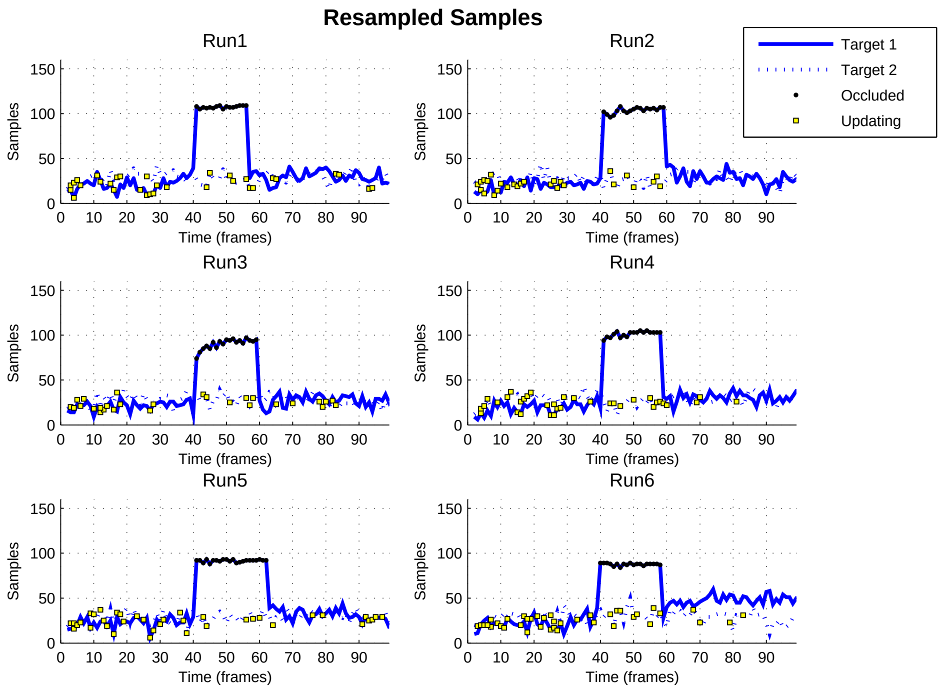


Figure 6.7: Re-sampled samples in the overtaking sequence

Sample survival rate		
	Target 1	Target 2
Run 1	38.6	28.5
Run 2	39.6	27.2
Run 3	36.9	27.3
Run 4	39.0	28.4
Run 5	41.2	28.7
Run 6	45.5	29.0
Mean	40.1	28.2

Table 6.3: Sample survival rate.

Thus, the number of them is constant while this situation holds, as can be observed in the aforementioned figure. The loss of a target due to the lack of samples have been avoided.

Fig. 6.7 shows the number of re-sampled samples. As before, it can be noticed that samples belonging to occluded targets are just propagated without pruning them. The survival rate is shown in Table 6.3. In the experiments carried out, the mean survival rate is 28.2% (values for target 1 are biased since during the occlusion no

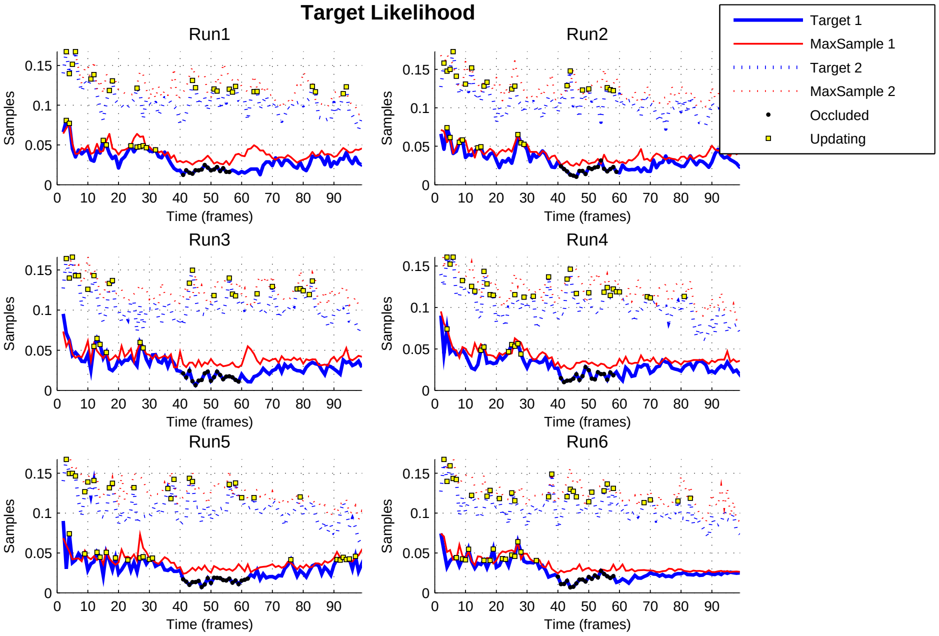


Figure 6.8: Target likelihood in the overtaking sequence

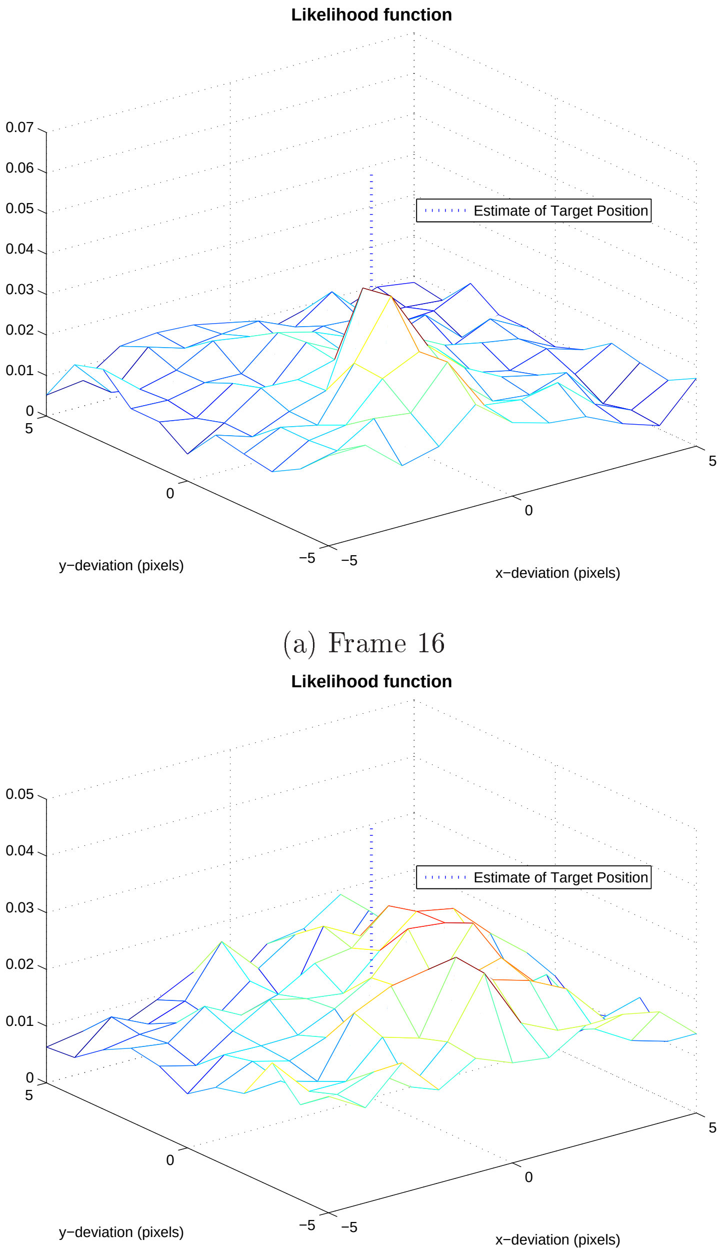
sample is re-sampled). Low values of survival rate indicate that there are significant differences among the likelihood values of the different samples. By making better predictions, this rate may be increased, a fact which represents an increment in the number of effective samples —those which are in fact tracking the target. Thus, the number of required samples may be reduced.

However, this is an endemic problem in particle filtering [3]. Despite the numerous approaches that have been tried —Partition Sampling [58], Covariance Sampling [85], Annealing Filtering [16], Unscented Particle Filter [89], etc— the problem is still open. Thus, according to [57], the evaluation of the *survival diagnostic*:

$$D = \sum_{i=1}^{N} \left(\pi^i\right)^2, \tag{6.1}$$

for the conventional particle filter [40] —given a 10-frame sequence with two targets using 2000 samples— yields a value below 5%; the same evaluation using Partition Sampling yields a value below 15%.

Fig. 6.8 shows the evolution of the targets' likelihoods. Target one corresponds to the two women whereas target two corresponds to the man. Two women being tracked, one behind the other, using just one tracker cause the significant lower values in the target likelihood, since bigger appearance changes occur between successive frames. The maximum sample likelihood is also drawn, using a thin red line. Usually, this value is higher than the target likelihood, since targets' states are obtained by



(a) Frame 16

(b) Frame 31

Figure 6.9: Likelihood values

(a) Appearance model

(b) Ground truth

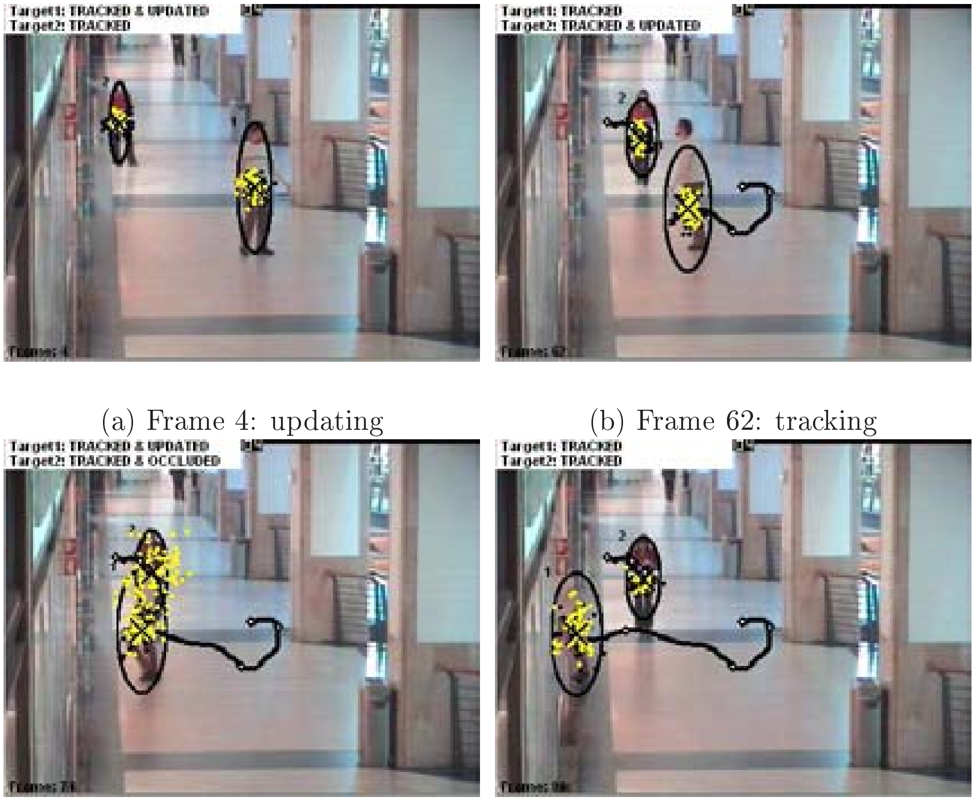
Figure 6.10: Target's appearance

averaging the weighted samples. In some cases, the sample with maximum likelihood corresponds to spurious state, given by a reduction of the target size caused by background clutter. In other cases, the maximum sample likelihood is lower than the target likelihood. This is caused by the fact that using a limited number of samples to model a highly dimensional pdf implies that this space cannot be densely populated, and thereby 'holes' are left.

Clutter is really a significant problem in these sequences. Numerous zones of the background mimic the target appearance in many pixels. Intensity is used as image feature. We try to overcome this problem using colour image features, and making use of global target characteristics, such as computing histograms. This last issue would prevent the effects of dealing with articulated and elastic targets, which likelihood, under certain conditions, may present significant falls due to pixel misalignments.

In the Fig. 6.9, the likelihood values around the target position at two different frames are shown. As can be seen, the likelihood function is highly multi-modal and present low values at the true position. The first target's appearance model —the two women— is show in Fig. 6.10.(a) and the corresponding image section in Fig. 6.10.(b). Significant differences in the corresponding pixels can be observed due to the articulated nature of the targets. These result strongly suggest that other likelihood functions should be explored.

The performance of the approach based on colour histograms has been tested using the CAVIAR database. In the sequence *OneLeaveShopReenter1cor* (CAVIAR dataset2, 389 frames @ 25 fps, 384 x 288 px), two targets are tracked simultaneously, despite their being articulated and deformable objects whose dynamics are highly non-linear, and that move through an environment which locally mimics the target colour appearance. The first target performs a rotation and heads towards the second one, eventually occluding it. It also presents challenging difficulties due to the fact the background colour distribution is so similar to target one that it constitutes a


Target1: TRACKED & UPDATED
Target2: TRACKED
Target1: TRACKED
Target2: TRACKED & UPDATED
Target1: TRACKED & UPDATED
Target2: TRACKED & OCCLUDED
Target1: TRACKED
Target2: TRACKED
1
2

(a) Frame 4: updating

(b) Frame 62: tracking

(c) Frame 74: occluded

(d) Frame 90: recovery

Figure 6.11: PF performance on *CAVIAR* sequence. Each target's estimated position is denoted by an ellipse and tagged accordingly; milestones are placed on the target trajectory every 25 frames; each predicted sample is drawn using a dark dot, whereas re-sampled particles are drawn in a light ones.

strong source of clutter. Furthermore, several oriented lighting sources are present, dramatically affecting the target appearance depending on its position and orientation (notice the bluish effect on the floor on the right of the corridor, and the reddish one on the floor on the left of the corridor). Thus, significant speed, size, shape and appearance changes can be observed, jointly with events such as people grouping, partial occlusions and group splitting. the environment locally mimics the target colour appearance, and several oriented lighting sources are present.

The tracker performance is shown in Fig. 6.11. Both targets' appearance models are updated when reliable measures are obtained, see Fig. 6.11.(a). Poor localisations and occlusions are correctly detected, thereby avoiding re-sampling of samples of the occluded target and erroneous dynamic and appearance models updating, see Fig. 6.11.(b), (c). The tracker successfully recovers from occlusion, see Fig. 6.11.(d).

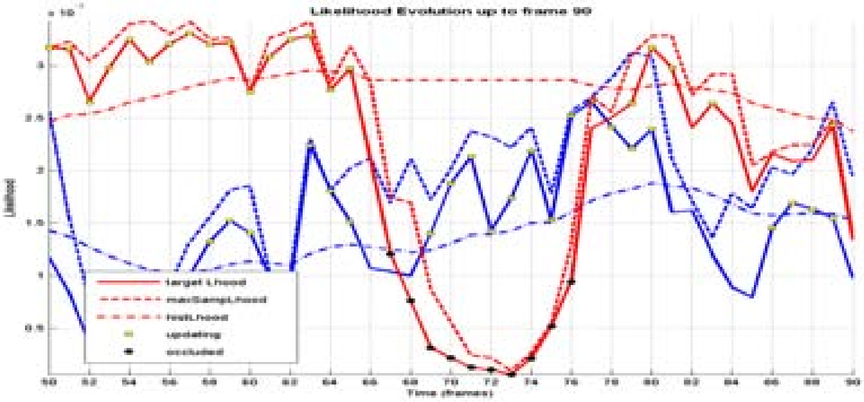


Figure 6.12: Likelihood evolution.

The maximum sample and target likelihoods, and the likelihood indicator is shown in Fig. 6.12.

Despite the achieved improvements, the experimental results show the limitations on the approach based on particle filters. Constrained appearance models are required. In the presented results, targets were initialised by hand. An automatic initialisation entails the necessity of dealing with common detection errors in cluttered and uncontrolled scenes. This approach would not cope with such ill-pose models.

Further, the need of model updating due to changing illumination conditions, or the non-rigid nature of the targets, implies assuming model contamination. Since the targets cannot be perfectly delineated, and small position errors are always present, the models will unavoidable drift. In addition, likelihood functions are not discriminative enough to mitigate the drift of the models. Thus, tracking in long-term sequences would not be feasible.

Obviously, these facts may be overcome by generating new samples from detection, and performing a data association process like in [41, 91]. However, this would just mask tracking misbehaviours what leads one to question about the feasibility of this kind of approaches.

Finally, due to the lack of a constrained dynamic model, an despite the improvements introduced, there is still a significant sample wastage. By assigning samples to an specific target instead of using a state vector which includes variables from all targets, we have tried to cope this effect —which is increased due to the curse of dimensionality. Other authors have tested other approaches like partitioned sampling [58], but the cause remains.

This considerations have lead to the development of the approach which results are presented in the following section.

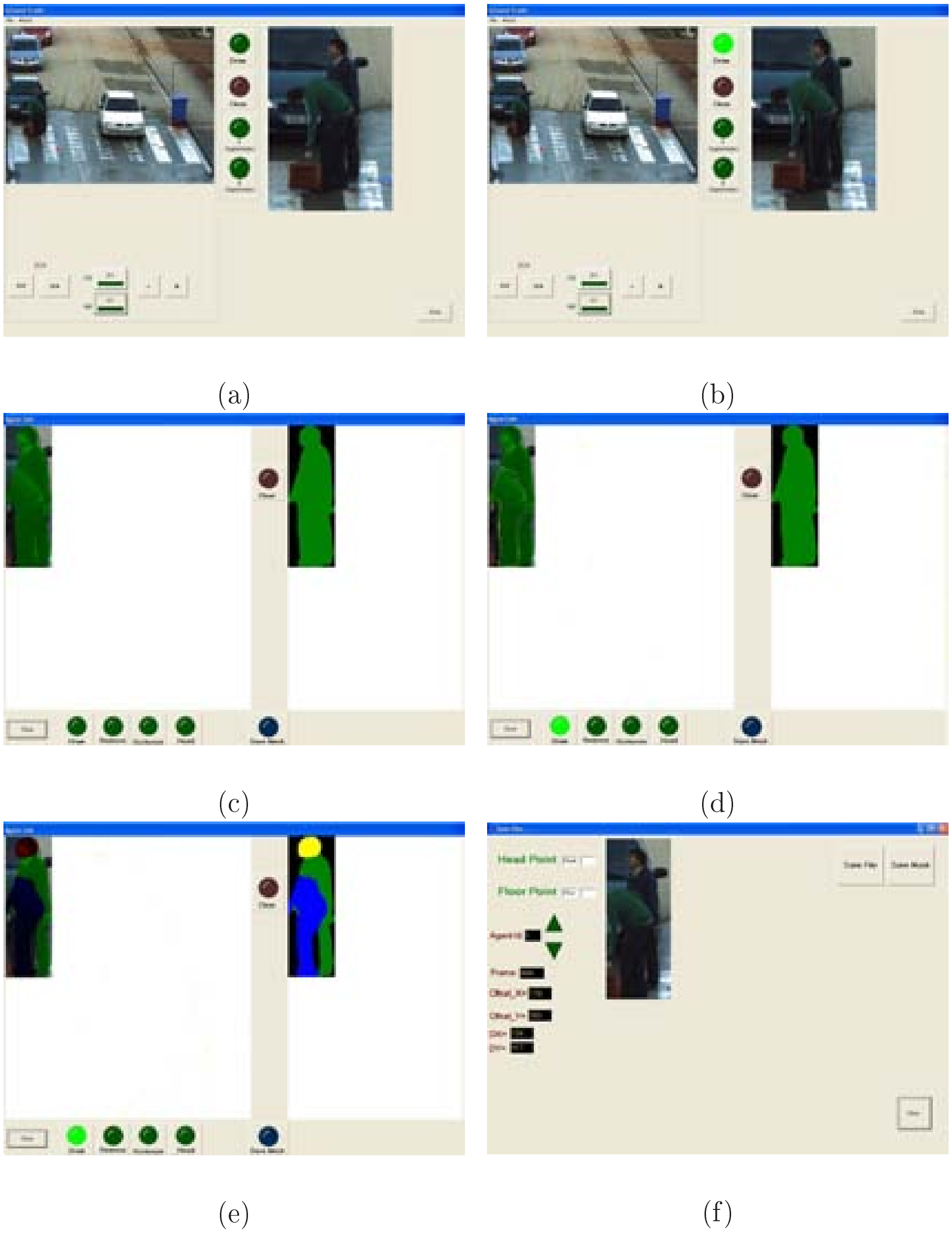


Figure 6.13: Annotation tool. (a) Main Windows and cropped region. (b) Segmented contour. (c) Annotation window. (d) Occluding target. (e) Results pointing out occluded regions and Head. (f) Identification window, target and frame labelling.

6.3 Evaluating the Performance of the Proposed Hierarchical Tracking Architecture.

The performance of our system has been tested using sequences taken from both public well-known databases, and own ones. Successful tracking results have been achieved in all processed sequences[15].

Further, a ground-truth annotation tool has been developed[16], and the interaction between human and computer is aided by using a pen tablet, see Fig. 6.13. Thus, foreground regions can be annotated, visualised and edited. Targets are labelled, and visible and occluded regions are pointed out, as well as significant parts as head or feet. As a result, a XML file is generated with the annotation data, and a set of target image masks are stored.

Significant processed frames of the previously used CAVIAR sequence are shown in Fig. 6.14. The following notation is used in all presented images: the contour of segmented blobs is painted on yellow; LLT's are denoted by black ellipses, wheres HLT's are represented by white ones; the security border is faded on blue, while outer areas are in grey.

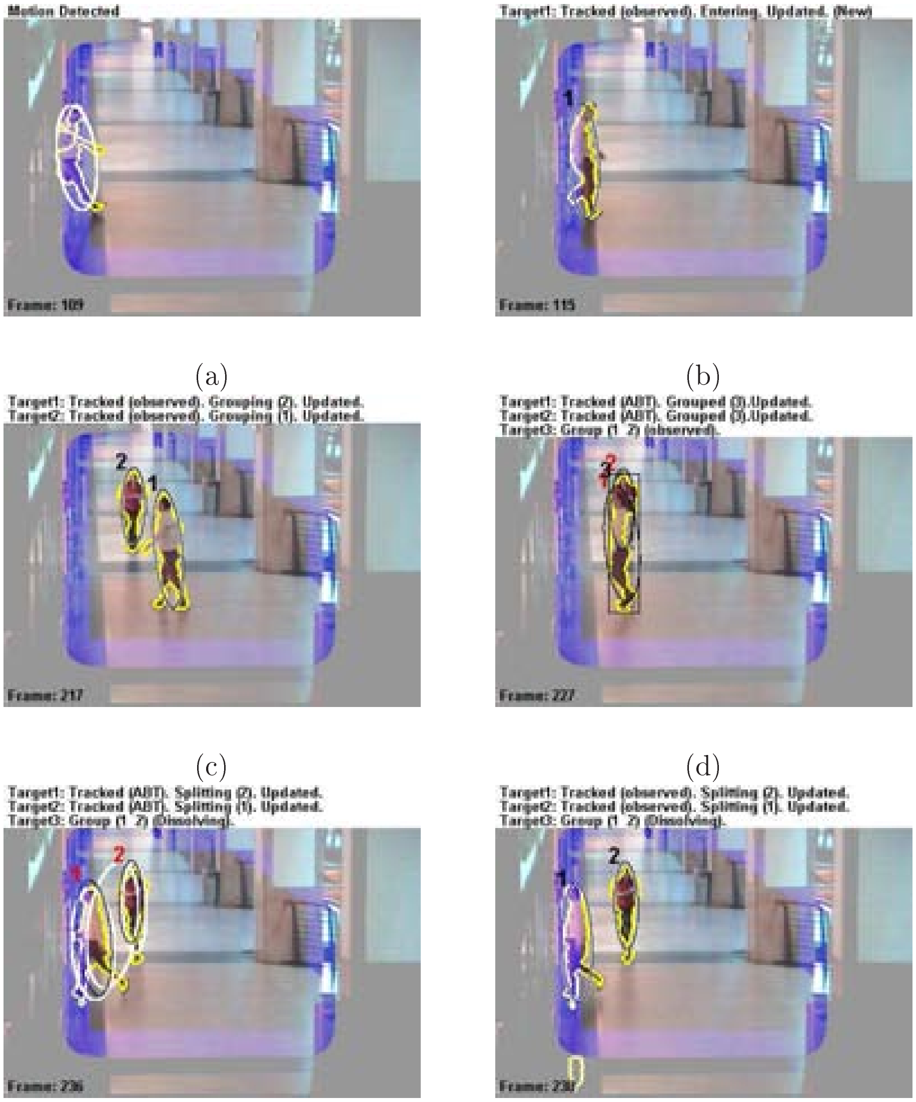
Tracking information is displayed in the box: the target label or identification (ID) is followed by the tracking status —tracked or lost— the operation mode —observed, or ABT— the interaction event and as attributes the partners involved, and whether the tracker is being updated, the target is occluded, or is entering/exiting the scene. For instance, in the Fig. 6.14.(e) targets one and two are both tracked in ABT mode, they are splitting one from each other, and the trackers have confidence enough to update the colour models; the group which both targets conformed is dissolving.

The sequence *DATASET1_ TESTING_ CAMERA1* (PETS 2001 database, 2688 frames @ 29.97 fps, 768 x 576 px) presents a high variety of targets entering into the scene: three isolated people, two groups of people, three cars, and a person who exits from a parked car. These cause multiple tracking events in which several targets are involved in different grouping, grouped, and splitting situations simultaneously. Samples of tracking results can be seen in Fig. 6.15.

A crosswalk sequence is analysed in *Zebra1* (CVC database, 1344 frames @ 25fps, 720 x 576 px). Four people are involved in different interaction events. Further, several vehicles cross the scene in a front plane, and people walk behind various streetlamps and trees, resulting in multiple partial an complete occlusions of the targets. Fig. 6.16 shows some significant frames.

[15] The reader is encouraged to see the whole processed sequences at http://iselab.cvc.uab.es/?q=agent_motion

[16] This is applied to the ISE lab database, see http://iselab.cvc.uab.es/?q=tools. The aim of these is to develop new techniques, technology, and algorithms for the automatic evaluation (i.e. motion detection, tracking, recognition and interpretation) of human behaviours in image sequences. ISE Lab is involved in an ongoing effort to develop datasets of synchronised videos, and ground-truth data. Consequently, the provided datasets are meant to aid research in developing, testing and evaluating algorithms for human-behaviour understanding.


Motion Detected
Frame: 109
Target1: Tracked (observed). Entering. Updated. (New)
1
Frame: 115
Target1: Tracked (observed). Grouping (2). Updated.
Target2: Tracked (observed). Grouping (1). Updated.
2
1
Frame: 217
Target1: Tracked (ABT). Grouped (3).Updated.
Target2: Tracked (ABT). Grouped (3).Updated.
Target3: Group (1 2) (observed).
2
3
1
Frame: 227
Target1: Tracked (ABT). Splitting (2). Updated.
Target2: Tracked (ABT). Splitting (1). Updated.
Target3: Group (1 2) (Dissolving).
1
2
Frame: 236
Target1: Tracked (observed). Splitting (2). Updated.
Target2: Tracked (observed). Splitting (1). Updated.
Target3: Group (1 2) (Dissolving).
2
1
Frame: 237

(a)

(b)

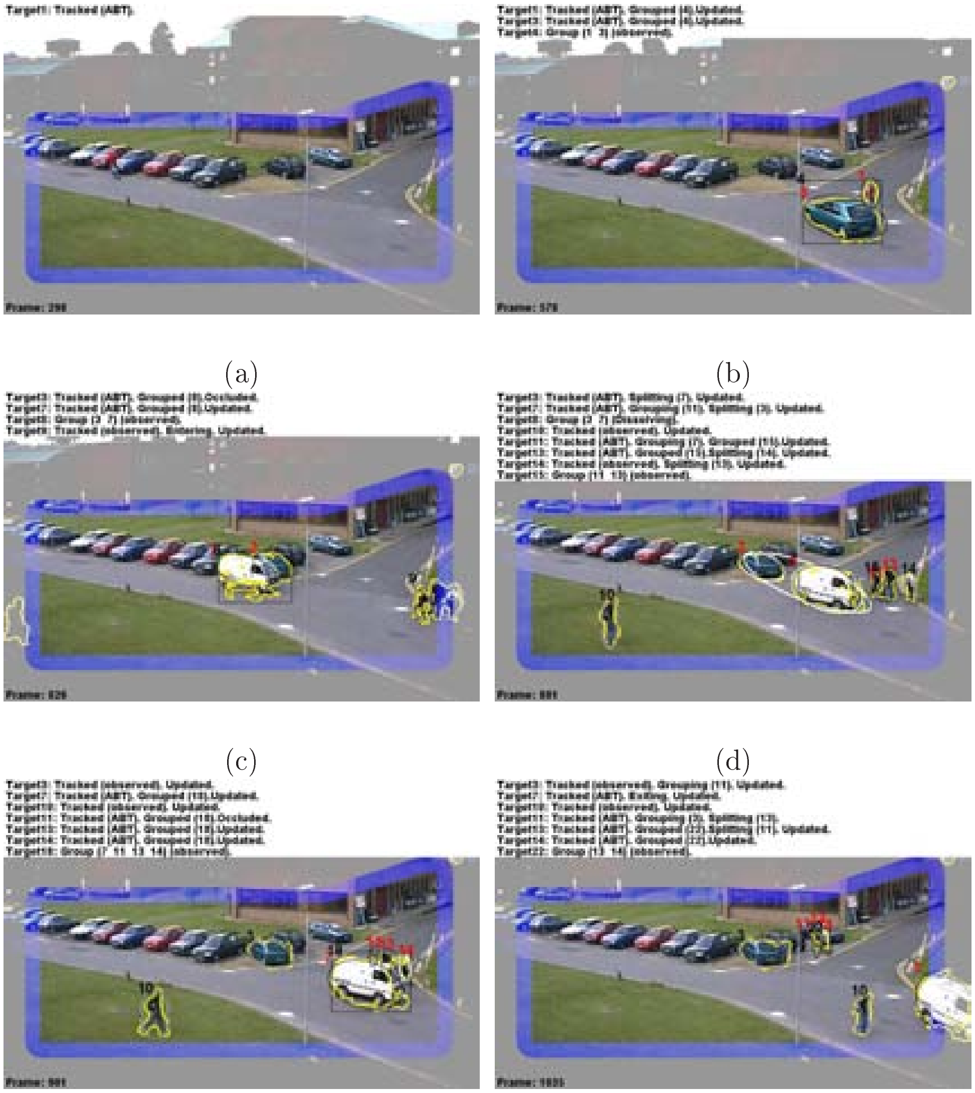
(c)

(d)

(e)

(f)

Figure 6.14: Sample tracking results on the *CAVIAR_ OneLeaveShopReenter1cor* sequence. Two targets are tracked; a LLT is instantiated in (a), and a HLT in (b); interaction events are correctly detected: (c) grouping, (d) grouped in (3), and (e) splitting; during the merging both targets are tracked using ABT; after it, new trackers are instantiated and correctly associated (f).



Target1: Tracked (ABT).

Frame: 298

(a)

Target1: Tracked (ABT). Grouped (4).Updated.
Target3: Tracked (ABT). Grouped (4).Updated.
Target4: Group (1 3) (observed).

Frame: 579

(b)

Target3: Tracked (ABT). Grouped (8).Occluded.
Target7: Tracked (ABT). Grouped (8).Updated.
Target8: Group (3 7) (observed).
Target9: Tracked (observed). Entering. Updated.

Frame: 829

(c)

Target3: Tracked (ABT). Splitting (7). Updated.
Target7: Tracked (ABT). Grouping (11). Splitting (3). Updated.
Target8: Group (3 7) (Dissolving).
Target10: Tracked (observed). Updated.
Target11: Tracked (ABT). Grouping (7). Grouped (15).Updated.
Target13: Tracked (ABT). Grouped (15).Splitting (14). Updated.
Target14: Tracked (observed). Splitting (13). Updated.
Target15: Group (11 13) (observed).

Frame: 891

(d)

Target3: Tracked (observed). Updated.
Target7: Tracked (ABT). Grouped (18).Updated.
Target10: Tracked (observed). Updated.
Target11: Tracked (ABT). Grouped (18).Occluded.
Target13: Tracked (ABT). Grouped (18).Updated.
Target14: Tracked (ABT). Grouped (18).Updated.
Target18: Group (7 11 13 14) (observed).

Frame: 991

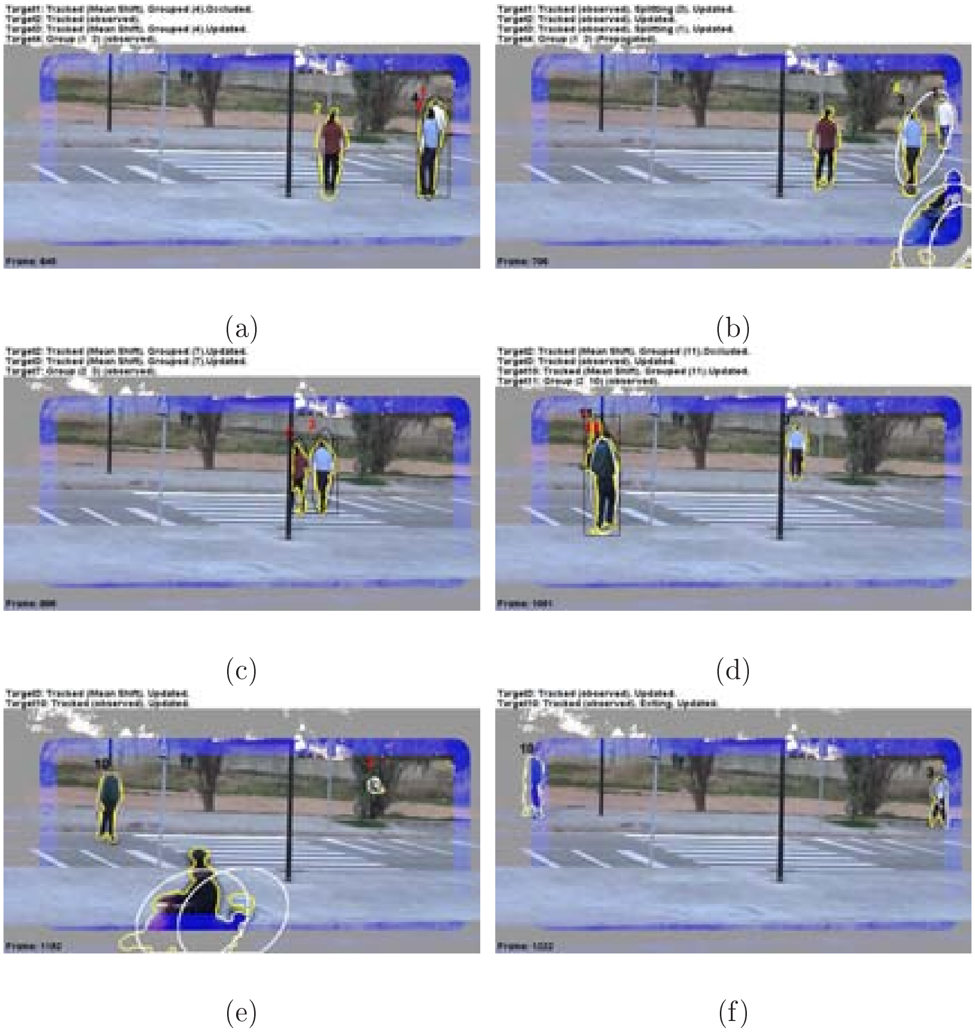
(e)

Target3: Tracked (observed). Grouping (11). Updated.
Target7: Tracked (ABT). Exiting. Updated.
Target10: Tracked (observed). Updated.
Target11: Tracked (ABT). Grouping (3). Splitting (13).
Target13: Tracked (ABT). Grouped (22).Splitting (11). Updated.
Target14: Tracked (ABT). Grouped (22).Updated.
Target22: Group (13 14) (observed).

Frame: 1035

(f)

Figure 6.15: Sample tracking results on the *PETS_DATASET1_TESTING_CAMERA1* sequence. Targets are tracked despite no segmentation is available in (a), a single blob is obtained for the group in (b), (d), or they are heavily occluded in (e); multiple simultaneous events are correctly inferred, such as target 13 is grouped in group 15 while splitting from 14 in (d).



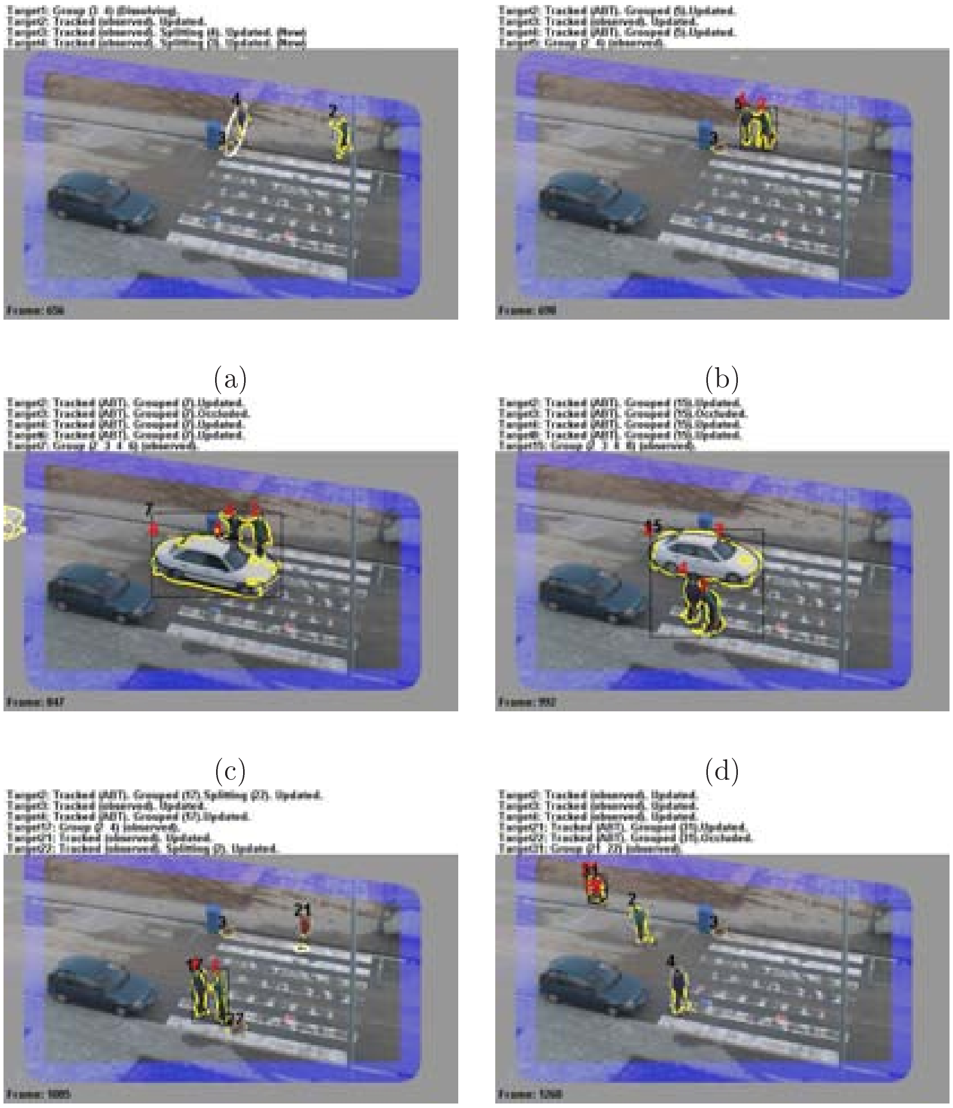
(a) (b)

(c) (d)

(e) (f)

Figure 6.16: Sample tracking results on the *CVC_Zebra1* sequence. Targets are successfully tracked despite mutual occlusions in (a) and (d), or occlusions with the background in (c) and (e); interaction and scene events are correctly inferred.

The *Hermes_Outdoor_Cam1* sequence (HERMES database, 1612 frames @ 15 fps, 1392 x 1040 px) presents a great diversity of situations. Three people and three cars act on a robbery sequence, where suitcases and bags are carried, left and picked from the floor. Multiple interaction events can be seen, in which several agents are involved in different simultaneous grouping, grouped and splitting events, while they are partially or completely occluded. Among the sequence difficulties, it must be also remarked that objects from the initial background are removed, several targets suffer from heavy background camouflage, and strong clutter is caused by similar group



Target1: Group (3 4) (Dissolving).
Target2: Tracked (observed). Updated.
Target3: Tracked (observed). Splitting (4). Updated. (New)
Target4: Tracked (observed). Splitting (3). Updated. (New)

Frame: 656

(a)

Target2: Tracked (ABT). Grouped (5).Updated.
Target3: Tracked (observed). Updated.
Target4: Tracked (ABT). Grouped (5).Updated.
Target5: Group (2 4) (observed).

Frame: 690

(b)

Target2: Tracked (ABT). Grouped (7).Updated.
Target3: Tracked (ABT). Grouped (7).Occluded.
Target4: Tracked (ABT). Grouped (7).Updated.
Target6: Tracked (ABT). Grouped (7).Updated.
Target7: Group (2 3 4 6) (observed).

Frame: 847

(c)

Target2: Tracked (ABT). Grouped (15).Updated.
Target3: Tracked (ABT). Grouped (15).Occluded.
Target4: Tracked (ABT). Grouped (15).Updated.
Target8: Tracked (ABT). Grouped (15).Updated.
Target15: Group (2 3 4 8) (observed).

Frame: 992

(d)

Target2: Tracked (ABT). Grouped (17).Splitting (22). Updated.
Target3: Tracked (observed). Updated.
Target4: Tracked (ABT). Grouped (17).Updated.
Target17: Group (2 4) (observed).
Target21: Tracked (observed). Updated.
Target22: Tracked (observed). Splitting (2). Updated.

Frame: 1085

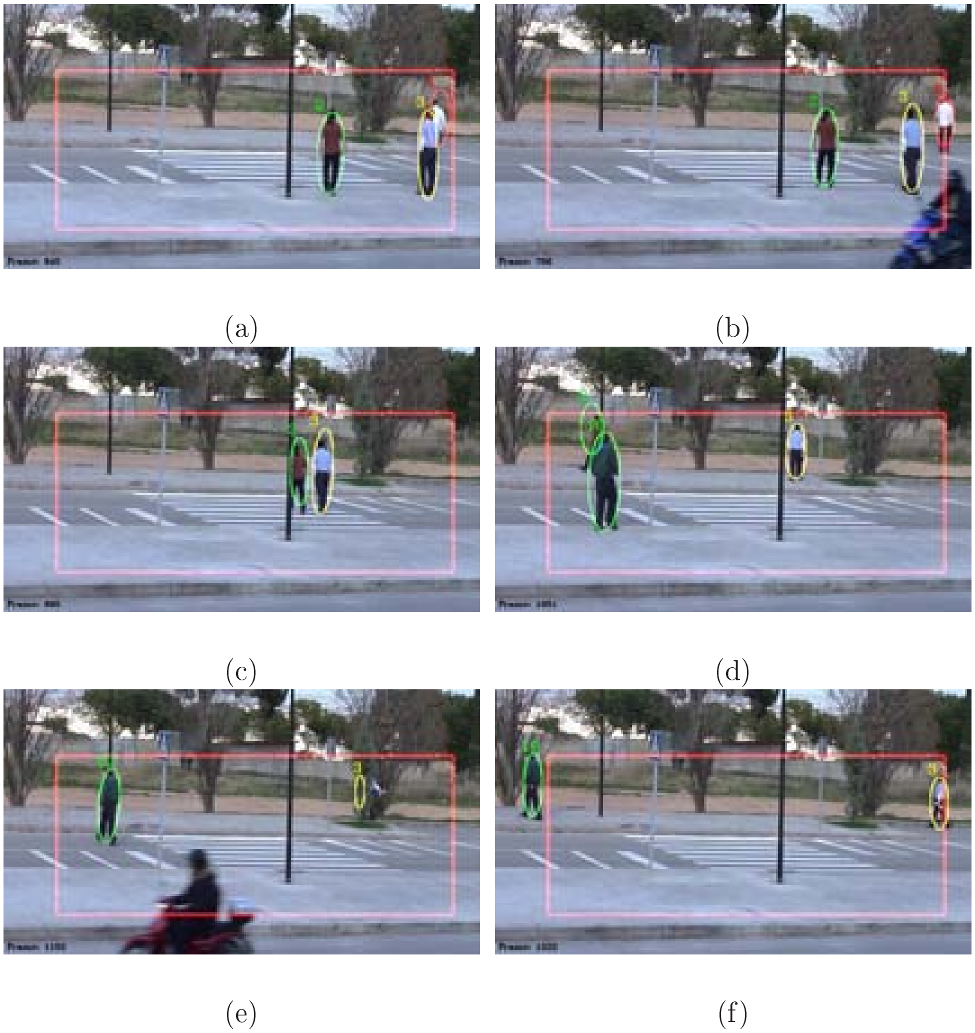
(e)

Target2: Tracked (observed). Updated.
Target3: Tracked (observed). Updated.
Target4: Tracked (observed). Updated.
Target21: Tracked (ABT). Grouped (31).Updated.
Target22: Tracked (ABT). Grouped (31).Occluded.
Target31: Group (21 22) (observed).

Frame: 1260

(f)

Figure 6.17: Sample tracking results on *HERMES_ Outdoor_ Cam1* sequence. The dissolution of a non-detected group form by target_4 —the man— and 3 —the bag— is correctly detected in (a), (e); targets are successfully tracked through groups in (b), partial occlusions in (c), and complete occlusions in (d); left objects are detected in (e), an correctly tracked after being picked up in (f).



(a) (b)

(c) (d)

(e) (f)

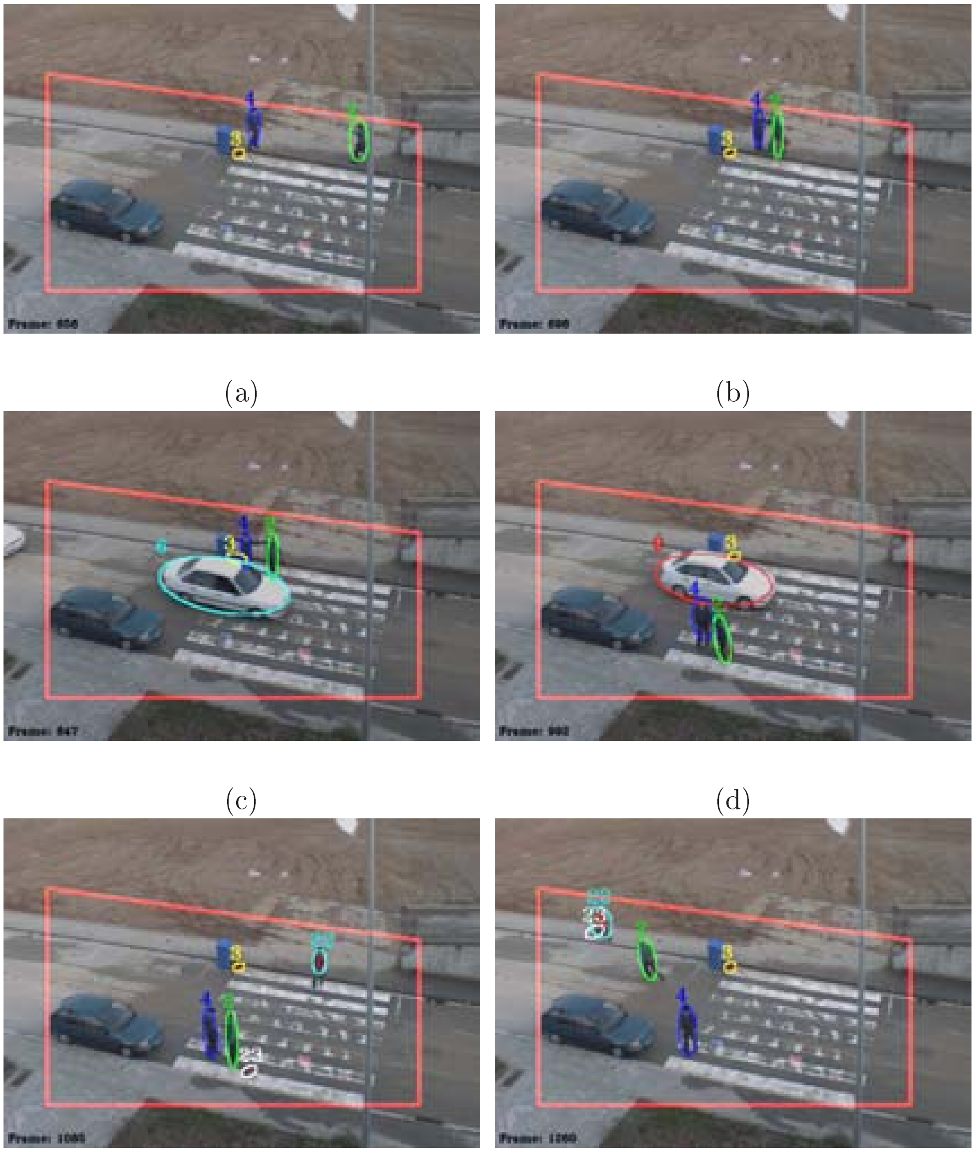
Figure 6.18: Demo tracking results on the *CVC_ Zebra1* sequence.

partners. Significant frames are shown in Fig. 6.17.

Finally, a football matched is recorded in the sequence *VS_ PETS_ Testing_ Camera4* (VS_PETS[17] database, 1570 frames analysed @ 25fps, 720 x 576 px). These sequence entails special difficulties given the high number of targets in the scene, and the fact that the appearance of all players from each team is identical.

Similar results are given in a demo mode, where all annotated information, scene regions, and intermediate results are removed for the sake of clarity. Thus, Fig. 6.18 shows some frames from *CVC_ Zebra1* sequence, whereas Fig. 6.19 does the same

[17] Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2003. http://www.cvg.rdg.ac.uk/VSPETS/



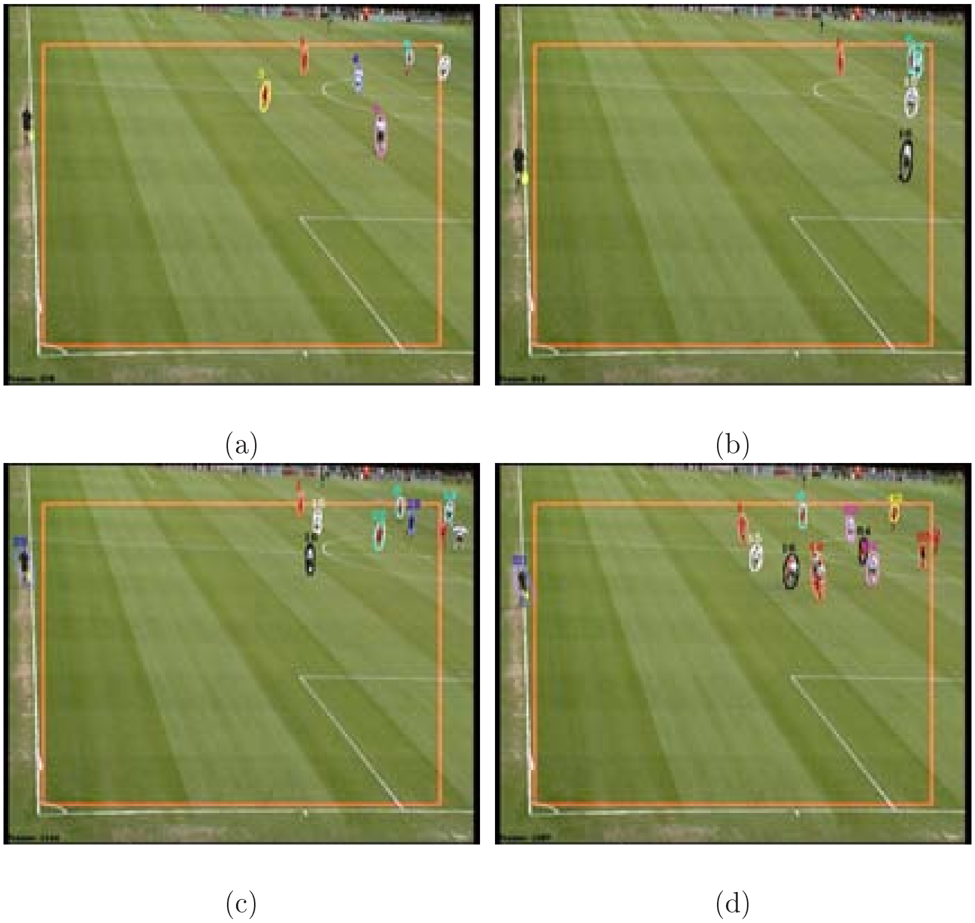
(a) (b)

(c) (d)

(e) (f)

Figure 6.19: Demo tracking results on the *HERMES_ Outdoor_ Cam1* sequence.

for *HERMES_ Outdoor_ Cam1* sequence, and Fig. 6.20 for the VS_PETS sequence. In this last sequence, the system fails after a thousand frames to accurately track all targets. Multiple facts entail this fact. First of all, the system must face a high



(a) (b)

(c) (d)

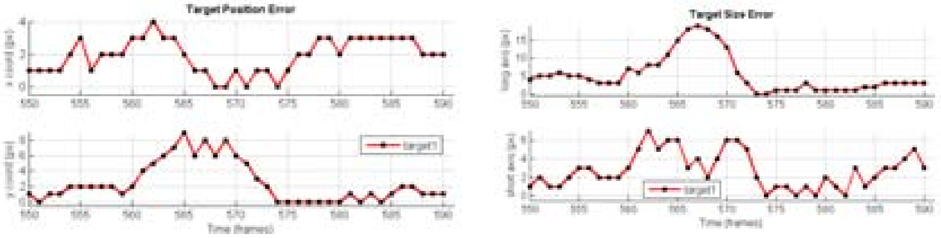
Figure 6.20: Demo tracking results on *VS_PETS_Outdoor_Cam1* sequence. (a) Multiple targets are simultaneously tracked; target 6 is successfully tracked in ABT mode, despite no segmentation is obtained —due to an wrong background model in the zone because a player was there during the initialisation; (b) target 4 partially occludes target 6; both are tracked in ABT mode; (c) the linesman is being tracked despite being out of the ROI since he has once stepped on it; (d) the system fails to track targets under tracker 16 and 49, see text for details.

number of targets in a low-resolution region. This issue can be solved by using a mosaic from registered multiple cameras. Secondly, no context constrains have been deliberately introduced into the system, but in a practical application the known appearance of the targets and background can be used[18]. In the third place, all the

[18] The aim has been designing a general system in order to cope with the maximum number of different scenarios. It should be easier to particularise the system later on, depending on

	Mean Error	Error Std. Dev.
x- position	1.93	1.06
y- position	2.46	2.62
Major axis	5.39	5.27
Minor Axis	2.77	1.90

Table 6.4: Error statistics.



(a)

(b)

Figure 6.21: Sample tracking evaluation. (a) Position and (b) size error of target1 in *Hermes_ Outdoor_ Cam1* sequence. Non-visible body-parts are also manually annotated; Major estimation errors correspond to frames with partial occlusions with the rubbish bin.

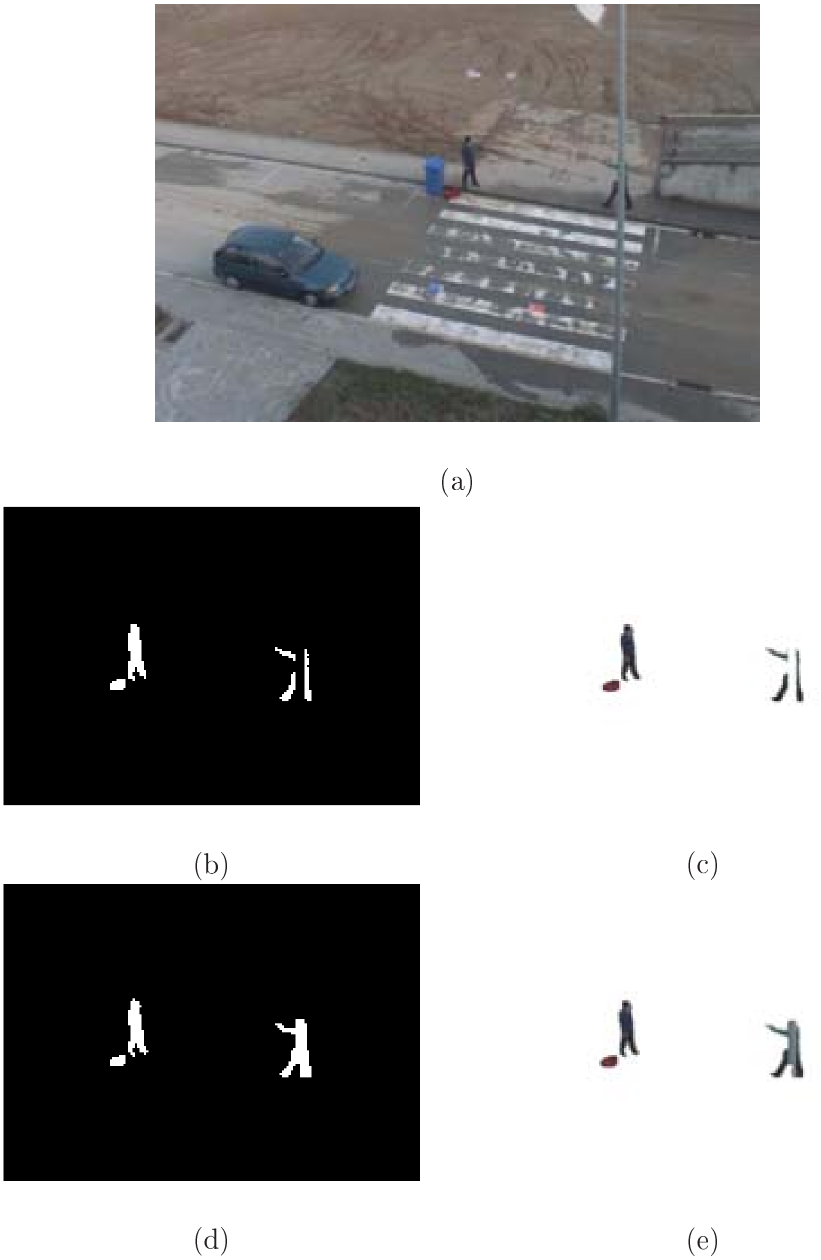
players of each team have the same colour appearance. As stated in the discussion on page 138, this issue is going to be addressed by introducing shape descriptors —as SIFT— to enhance disambiguation. Finally, many problems are caused by the fact that a single tracker is assigned to a group of targets that enter the scene together. As stated in the aforementioned discussion, this issue is being currently addressed within the HERMES project by the development of a target classification module.

Several of the above stated performance measures are here used to evaluate the system results, according to the data provided by the manual annotation tool. Thus, Fig. 6.21 shows the position and size error over time of target 1 in a *Hermes_ Outdoor_ Cam1*. Error statistics are shown in Table 6.4. Sample annotation frames are shown in Fig. 6.22.

Events are manually annotated and confronted with computed ones, see Table 6.5. Thus, events are correctly detected, albeit hardly ever occur at the exactly same time instant. This issue is of course sensitive to location estimation errors of a few pixels.

However, some errors due to the subjective component of the annotation remain. For example, in Fig. 6.17.(a) target 1 does not keep its ID after leaving the bag, due to major shape and appearance changes, and two new trackers are instantiated —target 3 and target 4. Hence, target 1 is referred as target 4 after bag —target 3— is left. Consequently, subsequent tracker instantiations have the labels shifted. Thus, the group is referred as target 4 in the annotated events and target 5 in the computed

a given scenario.



(a)

(b)

(c)

(d)

(e)

Figure 6.22: Sample annotation frames in *Hermes_ Outdoor_ Cam1* sequence. (a) Example of input frame; (b) manual annotation and (c) marked image without occluded target parts for segmentation-evaluation purposes; and (d) manual annotation and (e) marked imaged including occluded parts for tracking-evaluation purposes.

ones. Nevertheless, this ID change is desirable in other cases, such as a man leaving a child.

Further, several trajectory indicators over the tracked targets are computed and presented in Table 6.6. It must be remarked that just targets which enter completely in the scene are taken into account[19]. Thus, every time a new blob is detected, a LLT is instantiated. This usually happens when targets merge into groups, they dissolve themselves, or targets undergo significant changes due to camouflage, occlusions, etc.

Annotated event (t)	ID	Attrib.	Computed event (t)	ID	Attrib.
observed (550)	1	–	observed (550)	1	–
entering (629)	2	–	entering (629)	2	–
—			dissolving (655)	1	–
splitting (662)	1	3	splitting (655)	4	3
splitting (662)	3	1	splitting (655)	3	4
grouping (681)	1	2	grouping (682)	4	2
grouping (681)	2	1	grouping (682)	2	4
grouped (689)	1	4	grouped (697)	4	5
grouped (689)	2	4	grouped (697)	2	5
group (689)	4	1&2	group (697)	5	2&4

Table 6.5: Annotated and computed events on Hermes sequence. The attribute denote the targets involved.

Measure\Sequence	CAVIAR	PETS	CVC	HERMES
Targets	2	8	4	8
LLT	8	78	138	86
HLT (targets)	4	28	11	36
HLT (groups)	1	13	3	11
Temporarily Broken ID	0	0	1	2
Permanently Broken ID	0	0	0	2
False Positive	0	0	0	2
False Negative	0	0	0	0

Table 6.6: Trajectory measures.

[19] PETS results corresponds to the first 1300 sequence frames.

Module\Measure	Temporarily Broken ID	Permanently Broken ID	False Positive	False Negative
Normal operation	2	2	2	0
No use of bin weighting*	2	2	2	0
No ABT updating	4	4	2	0
No motion cues in ABT	3	6	2	0
Combined removal	3	9	2	0

*The indicators do not show a worse performance since the redundancies provided by the different modules make the errors no catastrophic enough to cause a target loss. However, a poor target localisation is obtained, as shown in Fig. 6.23.

Table 6.7: Effect of the different modules on tracking performance in the Hermes sequence.

Thus, the number of LLT's is much higher than the number of targets in every analysed sequence. When a LLT become stable, a HLT is created and associated with it. These are hopefully subsequently associated with the HLT that is already tracking the target. In this case, the target identity is not broken. When this process last more than one frame, the identity is temporarily broken. Since a HLT is created after the event is over, together with the fact that HLT are also instantiated to track groups, the number of HLT's is higher than the actual number of targets, even if the identities are correctly kept. Temporarily broken ID in CVC_Zebra sequence is due to an important partial occlusion of target 3 with a tree, see Fig. 6.16.(e). In the Hermes sequence this fact happens when the suitcase is picked up, due to significant segmentation errors. The permanent broken ID, and the false positives are due to *ghosts* yielded by a non-detected motionless car which starts motion[20].

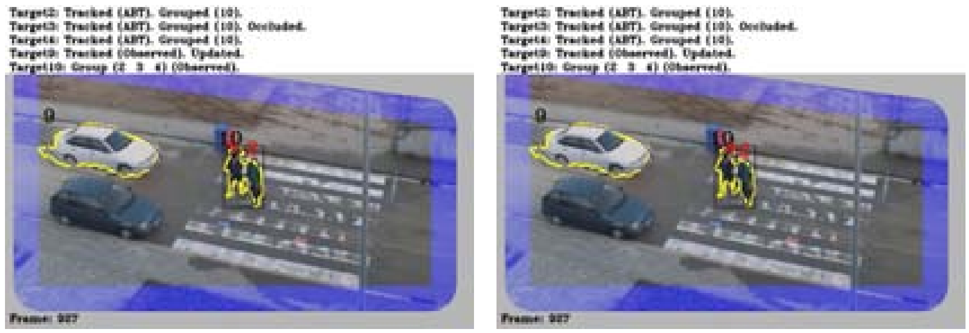
In order to to experimentally explore the effect of the different modules, several tests on the Hermes sequence have been carried out using the previous indicators. Thus, as shown in Table 6.7, the removal of any of this modules cause make the performance worse. Nevertheless, it should be remarked that these modules work in cooperation to maximise the target disambiguation from potential distracters. Therefore, since they provide some redundancy for the sake of robustness, the effect of removing only some of them may be not significantly noticeable.

Finally, it worth to say some remarks on the current implementation, and resulting real-time performances. As it have been above stated, multiple-people tracking in unconstrained and dynamic scenarios are one of the most computationally-speaking demanding task in Computer Vision.

The current system is implemented as a *Matlab* prototype. The focus has been placed on achieving robust and accurate performance, instead of on a careful code optimisation.

Significant speed improvements can be achieved by processing pixel-wise oper-

[20] This issue is commented in next section


Target2: Tracked (ART). Grouped (10).
Target3: Tracked (ART). Grouped (10). Occluded.
Target4: Tracked (ART). Grouped (10).
Target9: Tracked (Observed). Updated.
Target10: Group (2 3 4) (Observed).
9
10
Frame: 327
Target2: Tracked (ART). Grouped (10).
Target3: Tracked (ART). Grouped (10). Occluded.
Target4: Tracked (ART). Grouped (10).
Target9: Tracked (Observed). Updated.
Target10: Group (2 3 4) (Observed).
9
10
Frame: 327

(a) (b)

Figure 6.23: Bin-weighting effect on target localisation. Example of a poor localisation of target 4 due to the fact of no using the bin-weighting module in (b) in comparison with (a).

ations in parallel. In addition, many systems can benefit from specific hardware implementations like FPGA, DSP, GPU, etc. Low-level languages which give more direct access to the underlying machine allow faster computation as the expense of less readability and maintainability. On the contrary, interpreted languages are executed from source form, and are consequently slower. However, the code is often more flexible, allowing a faster prototyping.

Subsequent implementations of bottleneck modules[21] in C++ have yielded speed improvements which reduce 25 times the computation time of these particular functions. This would allow the system to process the above sequences at an average rate around 10 fps in a *Pentium V* @ 3200Mhz.

The computational complexity will be given by the complexity of each of the algorithms run at each module. For instance, the cost of the mean-shift algorithm is given by [14]:

$$C_O \approx N_i \left(c_h + P c_s\right), \tag{6.2}$$

where N_i is the mean number of iterations per frame an target, c_h the cost of computing the candidate histogram, P the number of target pixels, and c_s the cost of an addition, a squared root, and a division.

[21] The main bottleneck was located at the computation of the weighted histograms, given the huge number of evaluation required for selecting the best M feature out of 49, perform the appearance gating and association, the iterative mean-shift with multiple targets and group partners, multiple target models, the evaluation of background weights, and the final appearance updating.

Chapter 7

Concluding Remarks

In this thesis, the main goal has been achieving a robust and accurate Multiple-Target Tracking in human-populated scenarios. These should be as generic as possible, thereby limiting the number of assumed premises. The environment may be open, dynamic and uncontrolled. Towards this aim two approaches have been designed, implemented and experimentally verified.

The first proposal is founded on a particle-filter framework. PF algorithms has been widely used, specially between 1999 and 2003. These have been considered fast and efficient, and able to represent multi-modal density functions. Thus, with a fixed sample-set size —thereby, with bounded computational resources— multiple hypotheses could be simultaneously considered, in order to tackle background clutter.

Our approach was initially based on the PF algorithm implemented by Varona in his PhD at this Institute [90]. Subsequently, the focus has been placed in coping with the inherent drawbacks of SIR methods, and other common tracking difficulties, such as model drift. Then, after the evaluation of the obtained results and the feasibility of new enhancements, a second approach is developed and presented.

This second proposal –which constitutes the main contribution of this thesis— is based on a principled and structured framework. Thus, the system is implemented as a hierarchical but collaborative architecture, where each level is composed of several modules which are devoted to specific tasks. Therefore, this framework combines in a principled way both bottom-up and top-down tracking approaches.

7.1 Discussion and Contributions

The first approach has been widely explored, and finally discarded. Although significant advances have been obtained the approach is far from being appropriate to carry out multiple target tracking in unconstrained environments, specially in long sequences. Thus, the following issues have been explored and tested:

- Different appearance models and likelihood functions have been implemented. Thus, the approach has been evolved from using gray-scale templates computed from bounding boxes to colour histograms calculated from elliptical regions.

The former uses likelihood functions computed from the probability of each pixel value, whereas the latter relies on the Bhattacharyya distance.

- The dynamics updating stage has been modified to reduce sample wastage. The estimated speed is fed back into the prediction stage. All estimates are regularised.
- Model updating has been designed in order to overcome the model drift phenomenon. This is performed by taking into account the target likelihood, the evolution of this indicator, and the potential interactions with other targets.
- Sampling impoverishment have been tackled by redefining the weight normalisation. The proposed sample-weight normalisation avoids losing the targets due to the lack of samples.
- Occlusions have been dealt with by predicting collisions, and evaluating the target likelihood.

However, common problems of SIR filters have being inherited:

- High-dimension spaces cannot be densely populated, and estimations are often performed from a very limited number of samples. This results in poor state approximations when dealing with multi-modal pdf's.
- Top-down approaches require extremely constrained models, which is not feasible in generic applications. Errors in the estimation are propagated, thereby causing model drift.
- This problem is magnified by the fact that likelihood functions are usually not discriminative enough.
- An independent observation process from prediction is required to cope with estimation errors with a finite number of samples. This entails the necessity a bottom-up process.
- Finally, any generation of new samples from detection would just mask tracking misbehaviours. Survival rates are very low, and propagated samples would come from the newly generated ones.

The results obtained from the extensive experimental work carried out with the approach based on particle filters have led to the following preliminary conclusions:

- A bottom-up approach is required.
- Models must remain as simple as possible.
- The system must profit from all available sources of information.
- Multiple stages of hierarchical processing are desirable.
- More complex models at the higher levels need to be on-line built by using the information provided by the lower levels.
- These will be used to act on the lower levels.

Therefore, in order to take these issues to practice, a hierarchical but collaborative architecture has been designed. Each level feeds the higher one with its computed results, and is itself fed back with high-level results. In this way, by taking advantage of both approaches, the system is allowed to benefit from bottom-up capabilities; but also, high-level analysis is performed, granting accurately tuned models, and proper operation-mode selection:

- Three levels have been defined to perform each of the main system tasks: target detection, low-level tracking, and high-level tracking. Further, a remarkable characteristic of this architecture is that the tracking task is split into two levels. This fact is crucial to perform tracking without the need of previous detailed knowledge by introducing simultaneous modelling and tracking capabilities.

- Each level is fed with lower and higher level computed results. Further, each level has an internal feed-back loop.

- These levels can work according to two operation modes: Motion-Based Tracking (MBT) and Appearance-Based Tracking (ABT). These are independent and automatically selected according to each target particular conditions.

- A principled event management module is proposed and embedded in the architecture. Thus, a remarkable characteristic is its ability to manage multiple interactions among several targets. This allows the system to switch among different operation modes according to what situation is taking place. This capability is critical to achieve successful performances in uncontrolled scenarios. Further, a valuable knowledge is provided in order to obtain high-level scene descriptions.

- Feature-selection and appearance-computation modules have been developed, by paying special attention to the particular characteristics of grouping situations. Appearance is represented by means of multiple colour histograms. Histogram features are selected by considering not only the best distinction between the local background and the target, but also between the target and its group partners.

- A model pool is built, and long-run features are kept and smoothed. The use multiple features —including long-run ones— provide the system with recovering capabilities after grouping or camouflage events. Further, by smoothing the histograms the representation is less sensitive to potential initialisation and subsequent localisation errors.

- A procedure which takes into account motion and appearance cues relative to potential distracters has been designed to enhance the ABT operation mode. Thus, an important contribution focuses on tracking several targets independently while they are grouped, thereby yielding an accurate and robust target localisation, where other algorithms just provide coarse one.

- A principled model updating scheme has been followed to avoid model drift. Thus, targets are updated by considering the events in which they are involved. Targets tracked using MBT are updated when the track is confirmed as stable

—what depends of the quality of the observation sequence. Targets tracked using ABT are evaluated using the computed appearance models and similarity indicators before deciding whether update them or not.

Hence, the architecture proposed as second approach follows the natural paradigm, where visual-stimuli analysis is performed by the combination of pre-attentive and attentive processes. Further, it makes use of first-order and second-order motion perception. This results in a successful tracking of multiple targets simultaneously:

- This is achieved even in hard conditions of cluttered background and uncontrolled illumination.
- Targets present a high appearance and shape variability.
- Complex tracking events —in which numerous targets are simultaneously involved in different grouping and splitting situations— take place.

In spite of these difficulties, experiments on complex indoor and outdoor scenarios have yielded robust and accurate results. These have been carried out using sequences taken from both public well-known databases, and own ones, thereby demonstrating the system ability to deal with unconstrained and dynamic scenes:

- No a-priori knowledge about either the scene or the targets, based on a previous training period, is required.
- The method is adaptive in the sense of the background model, the number of targets, the best appearance representation, or the most appropriate tracking algorithm according to the events which are taking place.

The architecture itself must be seen as the main contribution, since it introduces the necessary synergies between the different modules and methods to tackle such an inherently complex problem. Therefore, each module task is performed by a particular algorithm, but they can be substituted by enhanced ones without modifying the architecture itself, thereby enhancing the system capabilities.

7.2 Open Issues and Future Work

As it has been stated multiple times, tracking success depends on the ability of distinguishing the target from potential distracters. An important effort has been made on this direction in all involved modules. Still, many cases remain in which no positive discrimination can be obtained using colour and intensity cues. Thus, target segmentation can be enhanced by making use of new cues:

- For instance, gradient-change detection can be used to attenuate target camouflage[1].

[1] This is currently being developed and tested. Promising results have already been achieved [36].

- Further, shadow removal techniques could be very useful to address those false detections due to changes in the illuminant chrominance.

An important remaining issue is caused by the background objects which are eventually removed. This fact leads to the so-called ghost detection problem:

- An analysis of the speed and contrast of newly created objects can be useful to tackle this open issue.
- Further, a multi-layered background can be built by including characteristics of left objects. Therefore, motion segmentation of new targets over former ones could be achieved, while ghost detection is mitigated.

Target classification was out of the scope of this work. However, a classification module can be easily inserted in the architecture, as shown in Fig. 5.1 on page 82:

- This would require a-priori learnt knowledge in order to distinguish among people, vehicles, and other objects in motion[2]. In addition, working in cooperation with detection modules, it would also help to segment targets who enter the scene within a group.

Target representation can be refined by including structure components and shape cues:

- For example, body-part histograms and salient points would enhance agent tracking during long-term partial occlusions, while SIFT descriptors would provide new ways of target discrimination.

The system is also prepared for taking advantage in the future of any high-level information about the context and current situations provided by cognitive levels of the HSE framework. Further, learning methods can be considered to tune algorithm parameters according to the particular conditions of a given scenario. The potential future use of multiple active cameras from several point of views is also feasible.

Finally, some remarks on what this system cannot do, or it is not intended to do. The premises taken in the design process assume that the background slowly changes with respect to the motion of the targets. Tracking is based on an initial motion segmentation in order to launch the LLT's. The issue maybe attenuated by modelling the background on a MoG basis, for instance. Still background motion should be limited.

Changes in both the target's dynamics and appearance are supposed to be smooth at the current frame rate. Very fast objects[3] cannot be tracked. The size of the targets in the image is assumed to be big enough in order to build a representative statistical appearance model, but small enough w.r.t the scene size to ensure that a coarse blob representation is feasible. Humans will essentially remain in upright posture. For instance, a single human recorded in a close-up image making fast movements cannot

[2] This issue is currently being addressed within our lab, according to the HSE framework.

[3] According to the scenario conditions, the selected frame rate, and the speed of the targets.

be tracked. Further, it is assumed that the size of the targets permits that they can completely lie within he ROI in order to perform event analysis.

In the current implementation —as a *Matlab* prototype— the focus has been placed on achieving robust and accurate results, instead of on real-time performances. It was not the aim of this thesis to design a system on a commercial platform. However, this system can be easily exported by taking into account the consideration discussed in Chapter 6.

Therefore, the system is designed to carry out trajectory analysis applications, such as people counting, video-surveillance, video-safety, extraction of sport match statistics, etc. Other further use requires the combination of this proposal with other systems which perform detailed human-body action analysis, or face tracking and facial expression analysis, etc. This is assumed to be performed by the remaining two channels within the HSE framework.

Acknowledgements. This work has been supported by the Catalan Research Agency (AGAUR), by the Spanish Ministry of Education (MEC) under projects AHNA (TIC2003-08865), SYSIPHUS (TIN2006-14606), and DPI-2004-5414, and by the EC under projects HERMES (IST-027110) and Vidi-Video (IST-045547).

Appendix A

Acronyms

Given the extensive use of acronyms through the text —related to the specific terms some already used in the literature, but most introduced in this work— we have found convenient to summarise them in Tables A.1, A.2.

Symbol	Description
ABT	Appearance-Based Tracking
ASL	Active-Sensor Level
BC	Bhattacharyya Coefficient
BCM	Background Colour Model
BIL	Behaviour-Interpretation Level
BIM	Background Intensity Model
CI	Confidence Interval
CIL	Conceptual Integration Level
CVS	Cognitive Vision System
EKF	Extended Kalman Filter
fps	frames per second
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HLT	High-Level Tracking
HMA	Human-Motion Analysis
HMM	Hidden Markov Models
HSE	Human-Sequence Evaluation
ISE	Image-Sequence Evaluation
ISL	Image-Signal Level
JPDAF	Joint Probabilistic Data Association Filter
KF	Kalman Filter

Table A.1: Acronyms (I).

Symbol	**Description**
LLT	Low-Level Tracking
MHT	Multiple-Hypotheses Tracker
MoG	Mixture of Gaussians
MTT	Multiple-Target Tracking
MSD	Mahalanobis Squared Distance
NL	Natural Language
NN	Nearest Neighbour
PDAF	Probabilistic Data Association Filter
pdf	Probabilistic Density Function
PF	Particle Filter
PDL	Picture-Domain Level
PTZ	Pan-tilt-zoom
ROI	Region of Interest
SDL	Scene-Domain Level
SIR	Sequential Importance Re-sampling
SIS	Sequential Importance Sampling
SPD	Spectral Power Distribution
UIL	User-Interface Level
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
WAGN	White Additive Gaussian Noise

Table A.2: Acronyms (II).

Appendix B

Symbol List

Due to the fact that numerous collaborative algorithms have been presented, this work has required the use of a large number of symbols. In order to aid the reader comprehension, these symbols are here summed up. They are split through several manageable tables, according to the symbol category. Thus, functions are described in Table B.1; indexes in Table B.2, constants in Table B.3, scalars and vectors in Table B.5, and matrices and data-structures in Table B.4.

For the sake of clarity, symbols exclusively related to the particle-filer approach are split in Table B.6.

Symb.	Description	Symb.	Description
$L(\lambda)$	illuminant SPD	$b(\bullet)$	bin-indexing function
$\mathcal{N}(\bullet)$	Gaussian pdf	$g_E(\bullet)$	Epanechnikov kernel profile
$R(\lambda)$	object reflectance distrib.	$\delta(\bullet)$	Kronecker delta
$S^c(\lambda)$	sensor sensitivity	$\phi(\bullet)$	general discrete distrib.
		$\chi^2_d(\bullet)$	Chi-squared pdf with d degrees of freedom

Table B.1: Functions.

Symb.	Description	Symb.	Description
a	pixel index (sub)	q	background index (sup)
c	channel index (sup)	t	time index (sub)
i	feature index (sup)	B	blue channel
j	entity index (sup)	G	green channel
k	bin index (sub)	I	intensity
l	particular entity index (sup)	R	red channel

Table B.2: Sub- and super-index symbols. Lowercase denote variables, while uppercase denote constants.

Symb.	Description	Symb.	Description
C	normalisation constant	ε	small positive quantity
J	number of group partners	κ_{ABT}	confidence factor for ABT updating
K	number of histogram bins	κ_D	conf. factor for dark foreground
M	current best features	κ_L	conf. factor for light foreground
N	best long-run features	κ_m	factor for the outer margin of the basin of attraction
P	number of target pixels	τ_m	minimum sensor sensitivity
T	number of window frames	τ_n	saturation sensor point
Δ_t	sampling period	τ_{σ^2}	covariance ellipsoid variance threshold

Table B.3: Constant symbols. They are represented by non-bold Latin uppercase, and non-bold Greek lowercase.

Symb.	Description	Symb.	Description
$\mathbf{A}$	transition matrix	$\mathbf{R}$	observation noise covariance
$\mathbf{C}$	output matrix	$\mathbf{S}_t$	innovation covariance
$\mathbf{G}$	noise matrix	$\mathcal{C}_t$	conceptual data
$\mathbf{I}_t$	current frame	$\mathcal{S}_t$	HLT data
$\mathbf{K}_t$	Kalman gain	$\mathcal{X}_t$	LLT data
$\mathbf{M}_t$	Segmentation map	$\mathcal{Z}_t$	observation data
$\mathbf{P}_t$	error covariance	$\Lambda_k^{i,l}$	log-likelihood ratio
$\widehat{\mathbf{P}}_t$	predicted error covariance	$V^{i,l}$	variance ratio
$\mathbf{Q}$	process noise covariance		

Table B.4: Matrices and data structures. Bold uppercase denotes matrices, while data structures are printed in calligraphic uppercase.

Symb.	Description	Symb.	Description
$d^{i,j}_{Bhat,t}$	Bhattacharyya distance	$\alpha_{a,t}$	brightness distortion
$d^2_{Mahal,t}$	Mahalanobis Squared Distance	$\overline{\alpha}_a$	brightness distortion RMS
$\widetilde{h}^j_t, \widetilde{w}^j_t$	observed major and minor axes	$\breve{\alpha}_{a,t}$	normalised brightness distortion
h^j_t, w^j_t	axes state-variable	$\alpha_{\mathbf{p}}$	histogram adaptation rate
$\dot{h}^j_t, \dot{w}^j_t$	axes change-rate state-variable	$\beta_{a,t}$	chrominance distortion
$\mathbf{p}^{i,j}_t$	histogram	$\overline{\beta}_a$	chrominance distortion RMS
$n^{i,j}$	counter of updating times	$\breve{\beta}_{a,t}$	normalised chrominance distortion
p_k	histogram bin	$\widetilde{\theta}^j_t$	observed angle
$\overline{\mathbf{p}}^{i,j}_t$	smoothed histogram	θ^j_t	angle state-variable
$\widehat{\mathbf{p}}^{i,j}_t$	histogram at estimated location	η^q	rate of exclusion for background bins
s^c	sensor response	λ	wavelength
$\mathbf{u}_t$	control signals	$\boldsymbol{\mu}_a$	colour-channel mean
w^c	channel weight	μ^I_a	mean intensity
w^i_k	bin weight	$\mu^{i,j}_t$	mean Bhattacharyya distance
$\tilde{w}^i_k$	equalised bin weights	ρ	Bhattacharyya coefficient
w_a	pixel weight	$\boldsymbol{\sigma}_a$	colour-channel std. dev.
$\mathbf{x}_a$	pixel location	σ^I_a	intensity standard deviation
$\mathbf{x}^j_t$	LLT state	$\sigma^{i,j}_t$	Bhattacharyya dist. std. dev.
$\hat{\mathbf{x}}^j_t$	LLT predicted state	σ^2_x	process variance due to acceleration
$\widetilde{x}^j_t, \widetilde{y}^j_t$	observed centroid	$\sigma^2_{\boldsymbol{\nu}}$	observation noise variance
x^j_t, y^j_t	centroid state-variable	τ_D	dark-foreground threshold
$\dot{x}^j_t, \dot{y}^j_t$	speed state-variable	τ_L	light-foreground threshold
$\mathbf{y}_t$	innovation	$\tau_{\alpha 1}$	low-brightness threshold
$\mathbf{z}^j_t$	observation	$\tau_{\alpha 2}$	high-brightness threshold
$\hat{\mathbf{z}}^j_t$	predicted observation	τ_β	chrominance threshold
		$\boldsymbol{\nu}_t$	observation noise
		$\boldsymbol{\omega}_t$	process noise

Table B.5: Scalar and vector symbols. Scalars are printed in non-bold lowercase, whereas vectors are denoted by bold lowercase.

Symb.	Description	Symb.	Description
c_t^i	cumulative prob. for sample i	$\mathbf{I}_t$	scene image at time t
$\mathbf{e}_t$	evidence variable instance	$\mathbf{I}_t^{l,p}$	predicted image region of target l
i	sample index	L	number of targets
l	target label	N	number of samples
$n_{\mathbf{e}}$	evidence-space dimension	M	number of pixels of the appearance model
$n_{\mathbf{s}}$	state-space dimension	$P(\bullet)$	probability density function
$p(\bullet)$	probability	$\tilde{P}(\bullet)$	simulated pdf
$\mathbf{p}_t^l$	target histogram	$Q(\bullet)$	proposal distribution
t	time index	$\mathbf{S}_t$	state random variable
$\mathbf{s}_t$	state variable instance at time t	$\alpha_{\mathbf{A}}$	appearance adaptation rate
$\mathbf{s}_t^l$	state estimate of target l	$\alpha_{\mathbf{p}}$	histogram adaptation rate
$\mathbf{s}_t^{i,l}$	sample i of target l	$\alpha_{\mathbf{u}}$	speed adaptation rate
$\widehat{\mathbf{s}}_t^{i,l}$	sample temporal prior	$\alpha_{\mathbf{x}}$	position adaptation rate
$\mathbf{u}_t^{i,l}$	speed of sample i	λ_t^l	expected likelihood
$\hat{\mathbf{u}}_t^{i,l}$	speed temporal prior of sample i	$\pi_t^{i,l}$	sample weight
$\mathbf{u}_t^l$	speed estimate of target l	$\overline{\pi}_t^{i,l}$	normalised sample weight
$\mathbf{w}_{t-1}^{i,l}$	size of sample i	ρ_t^l	occlusion status
$\hat{\mathbf{w}}_t^{i,l}$	size temporal prior of sample i	τ	time offset
$\mathbf{w}_t^l$	size of target l	$\xi_{\mathbf{u}}^i$	speed diffusion vector
$\mathbf{x}_t^{i,l}$	position of sample i	$\xi_{\mathbf{w}}^i$	size diffusion vector
$\hat{\mathbf{x}}_t^{i,l}$	position temporal prior of sample i	$\xi_{\mathbf{x}}^i$	position diffusion vector
$\mathbf{x}_t^l$	position estimate of target l	$\mathbf{\Sigma}_{\mathbf{u}}^l$	speed covariance matrix
$\mathbf{A}_t^l$	l-target appearance matrix	$\mathbf{\Sigma}_{\mathbf{w}}^l$	size covariance matrix
$\mathbf{E}_t$	evidence random variable	$\mathbf{\Sigma}_{\mathbf{x}}^l$	position covariance

Table B.6: Symbols related to the particle-filtering approach. Notation is consistent with the one used in the previous tables, and further, in a probabilistic context, uppercase denotes pdf's and random variables; while lowercase denotes probabilities and variable instances.

Appendix C

Basic Statistics

Probability theory provides a principled way of reasoning under the uncertainty derived from the impossibility of accessing to the whole truth about the environment. A probability model is based in four main elements. Thus, given an experiment with an uncertain result, the set of all possible ones is called *outcome set*; a subset of this is called *event* x; each event has a long-term relative frequency which is its *probability* p; and finally, a *random variable* X is a real function whose domain is the probability space S defined by the outcome set, all possible events, and their probabilities. Expressions involving random variables represent possible events in the probability model.

An *atomic event* is a complete specification on the state of the model by assigning a value to each defined random variable. Atomics events are mutually exclusive (two of them cannot be the case simultaneously), and the set of all of them is exhaustive (one must be the case).

An *unconditional* or *prior probability* is a statement about the probability of the event given by the expression on the random variable in the absence of any other information.

The probability theory is build from the *Kolmogorov axioms*:

1. all probabilities are between 0 and 1: $0 \leq p(x) \leq 1$,

2. the probability of the *true* event is 1, and the probability of the *false* event is zero,

3. the probability of the disjunction is given by:

$$p\left(x \vee y\right) = p\left(x\right) + p\left(y\right) - p\left(x \wedge y\right) \tag{C.1}$$

The probability of all possible outcomes for a random variable is given by its *probability distribution*, P. The distribution function of a discrete variable is called *probability mass function*, whereas it is called *probability density function* —since probabilities will be integrals— in case of continuous variables. A *joint probability distribution* of some variables provides the probabilities of all possible combined values of the involved random variables. The *full joint probability distribution* gives the joint distribution for the complete set of random variables.

The *conditional or posterior probability* is used when some knowledge is available. They are defined as:

$$p\left(x|y\right) \quad = \quad \frac{p\left(x \wedge y\right)}{p\left(y\right)}, \tag{C.2}$$

this give place to the so-called *product rule:*

$$p(x \wedge y) \quad = \quad p(x|y)p(y), \tag{C.3}$$

which can be defined in terms of probability distributions:

$$P\left(X,Y\right) \quad = \quad P\left(X|Y\right)P\left(Y\right). \tag{C.4}$$

Marginals probabilities are obtained by extracting the probability distribution of some subset of variables in a process called *marginalisation*:

$$P(X) \quad = \quad \int P\left(X,y\right)dy. \tag{C.5}$$

Making use of the product rule, the *conditioning* rule can be derived:

$$P(X) = \int P\left(X|y\right)p\left(y\right)dy. \tag{C.6}$$

The *Bayes' theorem* is deduced from the product rule:

$$P\left(X|Y\right) \quad = \quad \frac{P\left(Y|X\right)P\left(X\right)}{P\left(Y\right)}, \tag{C.7}$$

which can be also conditionalised:

$$P\left(X|Y,z\right) = \frac{P\left(Y|X,z\right)P\left(X|z\right)}{P\left(Y|z\right)}. \tag{C.8}$$

Random variables are said to be *independent* if the following equivalent expression hold:

$$\begin{aligned} P\left(X\left|Y\right.\right) &= P\left(X\right), \\ P\left(Y\left|X\right.\right) &= P\left(Y\right), \\ P\left(X,Y\right) &= P\left(X\right)P\left(Y\right). \end{aligned} \tag{C.9}$$

On the other hand, random variables are *conditionally independent* if:

$$P\left(X, Y|Z\right) \quad = \quad P\left(X|Z\right)P\left(Y|Z\right), \tag{C.10}$$

which allows to decompose large probability models into manageable sub-models. In addition, conditional independence assumptions are much more realistic than absolute independence ones. Thus, it leads to the so-called *naïve Bayes* model in which a full joint distribution concerning a cause and its effects is decomposed considering that the effects are independent, given the cause:

$$P\left(Cause, Effect_1, ..., Effect_n\right) \quad = \quad P\left(Cause\right)\prod_{i=1}^{n}P\left(Effect_i|Cause\right) \tag{C.11}$$

The expected long-term average observed value of a distribution, called the population *mean*, is given by:

$$\mathbb{E}\left[X\right] \quad = \quad \int xp(x)dx. \tag{C.12}$$

The values given by:

$$\mathbb{E}\left[X^k\right] \quad = \quad \int x^k p(x)dx, \tag{C.13}$$

are called *raw moments.* The *central moments* are given by:

$$\mathbb{E}\left[(X-\mu)^k\right],$$

where μ is the population mean, or the first order raw moment. The second order central moment, commonly denoted by σ^2, is called *population variance.* The following relation holds:

$$\sigma^2 = \mathbb{E}\left[\left(X-\mu\right)\left(X-\mu\right)^T\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}^2\left[X\right]. \tag{C.14}$$

The covariance of two random variables X, Y is defined by:

$$\operatorname{cov}\left(X, Y\right) = \mathbb{E}\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right]. \tag{C.15}$$

Two variables are said to be uncorrelated if their covariance is zero, which implies:

$$\mathbb{E}\left[XY\right] = \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right]. \tag{C.16}$$

Appendix D

Kalman Filter

The Kalman filter [48] is a stochastic state estimator developed by Rudolph E. Kalman in 1960. It implements a recursive algorithm which works in a prediction-correction way, estimating the system state from noisy measures. The estimator is optimal in the sense that it minimises the steady-state error covariance:

$$\mathbf{P} = \lim_{t \to \infty} \mathbb{E}\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right]. \tag{D.1}$$

However, strong assumptions are required: the transition model must be linear Gaussian, and the sensor model must be Gaussian. Nevertheless, albeit these conditions rarely exist, the filter still works reasonably well for many applications, and it has been widely used[80].

It works as follows. The process is assumed to be governed by a linear stochastic difference equation:

$$\mathbf{x}_t \quad = \quad \mathbf{A}\mathbf{x}_{t-1} + \boldsymbol{\omega}_t, \tag{D.2}$$

where

- $\mathbf{x}_t \in \mathcal{R}^n$ is the system state, n the state-space dimension, and t a discrete time index,
- $\mathbf{A}$ is a $n\, x\, n$ matrix describing the linear transition model,
- $\boldsymbol{\omega}_t \sim \mathcal{N}(0, \mathbf{Q})$ is the process noise, and $\mathbf{Q}$ the noise covariance. Hereby, zero-mean white additive Gaussian noise is assumed to represent modelling uncertainties and disturbances.

The measure process is assumed to be governed by the next equation:

$$\mathbf{z}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\nu}_t, \tag{D.3}$$

where,

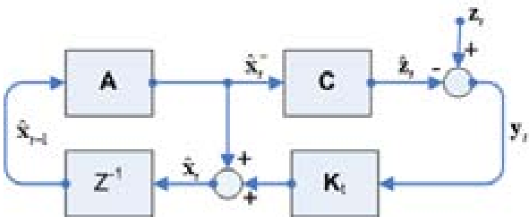


Figure D.1: Diagram block of a Kalman state estimator. See text for details.

- $\mathbf{z}_t \in \mathcal{R}^m$ is the measure vector, and m the measure-space dimension,
- $\mathbf{C}$ is a $m\,x\,n$ matrix relating the state to measure,
- $\boldsymbol{\nu}_t \sim \mathcal{N}(0, \mathbf{R})$ is the sensor noise, and $\mathbf{R}$ the noise covariance. Hereby, zero-mean white additive Gaussian noise is assumed to represent measurement noise.

It is also assumed that both process and measurement noise are uncorrelated:

$$Cov\left(\boldsymbol{\nu}_t \boldsymbol{\omega}_t^T\right) \quad = \quad 0. \tag{D.4}$$

The initial state is unknown, but it is assumed that it follows a normal law:

$$\mathbf{x}_0 \sim \mathcal{N}(\mu_0, \mathbf{P}_0), \tag{D.5}$$

where

- $\mathbf{x}_0$ is the system initial state,
- μ_0 is the initial distribution mean,
- $\mathbf{P}_0$ is the initial distribution covariance.

Independence of process noises $\boldsymbol{\omega}_t, \boldsymbol{\nu}_t$ and initial state $\mathbf{x}_0$ is assumed.

The filter works in two steps which are recursively performed —a block diagram is shown in Fig. D.1. In the first one, a prediction is made: the expectation and covariance are propagated according to the dynamic model, thereby obtaining the temporal prior:

$$\begin{aligned}\hat{\mathbf{x}}_t^- &= \mathbb{E}\left[\mathbf{A}\mathbf{x}_{t-1} + \boldsymbol{\omega}_t\right] \\ &= \mathbf{A}\hat{\mathbf{x}}_{t-1},\end{aligned} \tag{D.6}$$

and the prior covariance matrix:

$$\begin{aligned}\mathbf{P}_t^- &= \mathbb{E}\left[\left(\mathbf{x}_t - \mathbb{E}\left[\mathbf{x}_t\right]\right)\left(\mathbf{x}_t - \mathbb{E}\left[\mathbf{x}_t\right]\right)^T\right] \\ &= \mathbb{E}\left[\left(\mathbf{A}\left(\mathbf{x}_{t-1} - \mathbb{E}\left[\mathbf{x}_{t-1}\right]\right) + \mathbf{w}_t\right)\left(\mathbf{A}\left(\mathbf{x}_{t-1} - \mathbb{E}\left[\mathbf{x}_{t-1}\right]\right) + \boldsymbol{\omega}_t\right)^T\right] \\ &= \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q}. \qquad \text{(D.7)}\end{aligned}$$

After obtaining the new measurement $\mathbf{z}_t$, the second step is carried out, and values are updated according to the observation likelihood:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\mathbf{y}_t, \qquad \text{(D.8)}$$

$$\mathbf{P}_t = \mathbf{I} - \mathbf{K}_t\mathbf{C}\mathbf{P}_t^-, \qquad \text{(D.9)}$$

where:

$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{C}\hat{\mathbf{x}}_t^-, \qquad \text{(D.10)}$$

is called the *innovation* or the *residual,*

$$\mathbf{S}_t = \mathbf{C}\mathbf{P}_t^-\mathbf{C}^T + \mathbf{R}, \qquad \text{(D.11)}$$

is called the *innovation covariance*, and

$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{C}^T\mathbf{S}_t^{-1}, \qquad \text{(D.12)}$$

is known as the *Kalman gain.*

Appendix E

Biological Foundations of the Proposed Hierarchical Architecture

In the following, a brief depiction of a biological paradigm —which can be seen as a natural inspiration for the proposed architecture— is presented. Stress is laid on (i) the capabilities of a natural vision system, and (ii) at which level and how the decisions are taken. This section complements the exposition made in section 5.2 on page 84.

E.1 Natural Vision System

According to Urtubia [88], vision is the capacity of processing information about the environment by means of light stimuli incident on the retina. The retina is a layer of neural cells that generate visual neural signals. These cells contains a protein responsible for photo-reception: the opsin. Two kind of opsin are present in human retina, namely *rod* opsins and *cone* opsins. The conjunction of both provide the different visual capabilities.

Thus, rods are mainly located in the periphery of the retina, while cones have a higher concentration in the *fovea*, at the centre of the retina. There are three subtypes of cones which differ in the light wavelength to which they are receptive. They are consequently called red, green and blue cones.

Rods are used to see at low levels of light, while cones allow to distinguish colour and other features at normal light intensities. Hence, rods are responsible for *peripheral vision*, which occurs outside the centre of gaze, that is, outside the *macula*. Due to the lower density of cells, peripheral vision is less accurate in humans. This along with the fact that these cells are mainly rods, cause poor peripheral vision capabilities in distinguishing color and shape. However, the peripheral vision present another significant feature: the ability of motion detection. Thus, it provides good motion detection capabilities. Further, it is predominant in the dark, since the lack of light makes cones useless, whereas on the contrary rods get easily saturated.

Motion perception is the process of inferring the speed and direction of any object

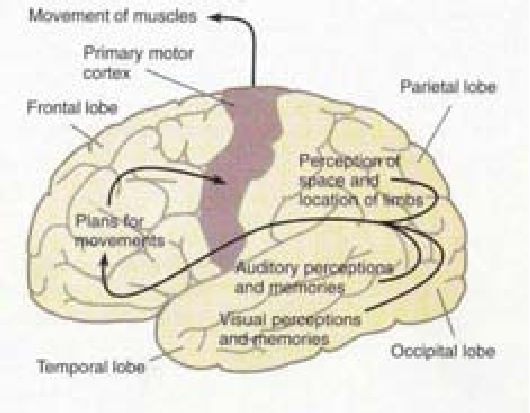


Figure E.1: Motor command generation. Signals involved in voluntary motor responses are generated at the cortex, according to current sensory perceptions and memories. (Figure from Psychology 465/665 notes, Nawrot, 2002).

that moves in a scene, given some visual input. Several kind of motion perception processes can be carried out. First-order motion perception is performed by detecting luminance changes in a particular point of the retina, and correlating it with a delayed change at a neighbouring point. On the contrary, second-order motion perception depends on moving contours defined in terms of contrast —difference in the color and brightness with the surroundings— or texture.

Visual signals are processed through three layers of neurons, namely photo-receptors, bipolar cells, and ganglionic cells. Then, neural signals are relayed to the brain through the optic nerve: the biggest ganglionic cells relay information related to motion and intensity from the periphery through the *magnocellular system*, whereas the smallest ones transmit colour and acuity information from the macula through the *parvocellular system.*

Complex visual information is processed in the *visual cortex*, which is the most massive system in the human brain. It is, thereby, responsible for a high-level processing of the acquired image sequences. Signals are first transmitted to the *Primary Visual Cortex* (V1), in where cells respond to particular chromatic stimuli and edge orientation. Then, signals are relayed to the associative areas (V2, V3, V4 and V5) where various analyses are carried out on motion, dynamic shapes, colour, and shape associated to colour. All these analyses converge in the *inferior temporal cortex* (IT), where pattern recognition is accomplished. See [68, 88] for details.

E.2 Natural Motor Responses

The cerebral *cortex* is the outer layer of a vertebrate brain. Mammals have developed a top cortex layer called *neocortex*, which is itself composed of six layers[1]. These

[1] The fact of including this section may surprise the reader, since the thesis is devoted to Computer Vision. However, two issues should be considered. In the first place, although the

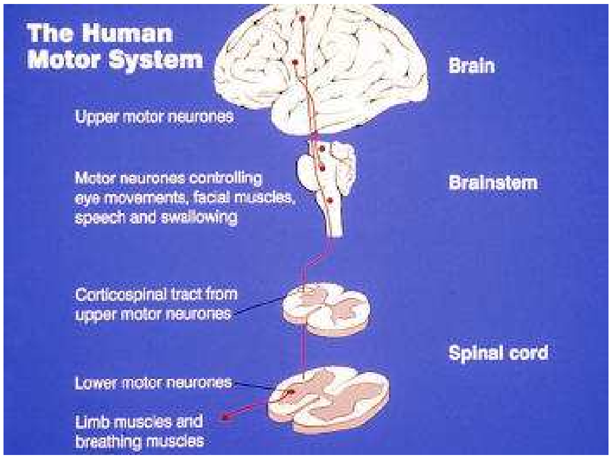

The Human Motor System
Brain
Upper motor neurones
Motor neurones controlling eye movements, facial muscles, speech and swallowing
Brainstem
Corticospinal tract from upper motor neurones
Spinal cord
Lower motor neurones
Limb muscles and breathing muscles

Figure E.2: Human motor system. Multiple hierarchical levels are involved in the different loop in which decisions are taken according to sensory inputs. (Figure from Biomedical Science notes, Shaw, 2007)

are labelled from I, the outermost, to VI, the innermost. In mammal species, the neocortex is the part of the cerebral cortex responsible for higher functions, such as sensory perception, and the generation of motor commands, see Fig E.1.

Thus, the neocortex region involved in planning, controlling, and executing voluntary motor responses is called *motor cortex*. This is divided into the *primary motor cortex* (M1), and the *secondary motor cortices*. The former is responsible for generating the neural impulses which control the execution of movements. Among the latters, the *posterior parietal cortex* is involved in transforming visual information into motor commands; and the *pre-motor cortex* plays an important role in sensory guidance of movement.

Further, other brain regions outside the cortex are also strongly related to motor functions. Among these, the most notably ones are the cerebellum, the pons, and the medulla oblongata. The cerebellum —located at the inferior posterior part of the brain— provides a feed-back loop in order to tune motor movements according to sensory perception of body posture. It sends this information to the motor cortex

part of the HSE framework addressed in this thesis uses stationary cameras, the ultimate aim of HSE is to benefit from the obtained results at this stage to act on multiple cameras. These will be able to focus on the scene region where tracking is being done by panning, tilting and zooming in order to provide results from the two remaining information channels: body pose and face expression. Further, the system can be provided with the capability of acting on the scene, for example, by opening doors, or switching lights. Both aspects can be considered as motor responses.

Secondly, the aim of HSE is to emulate human skills in inferring other human behaviour, and acting in consequence. In this task, multiple decisions are taken at every level, and the information flow follows both bottom-up and top down pathways, thereby creating numerous loops. The paradigm is clearly represented in the Natural Motor System itself, and in its relations with the Sensory Systems.

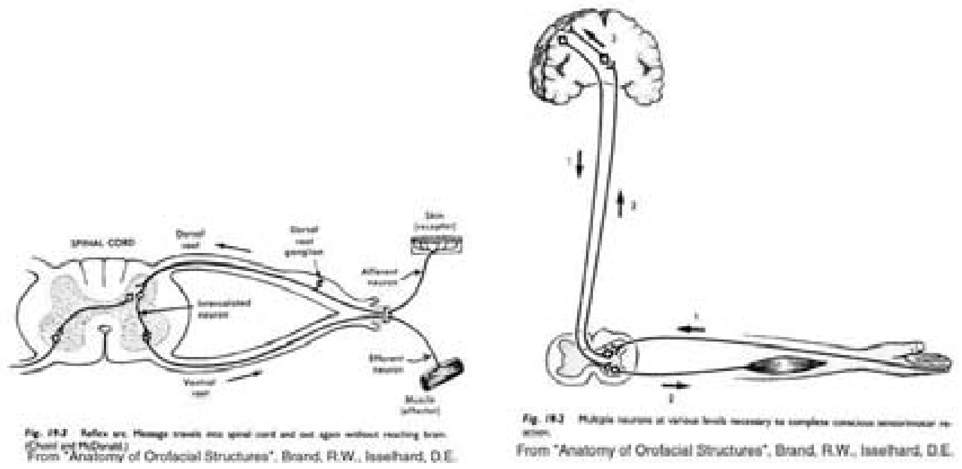


Fig. 19-2 Reflex arc. Message travels into spinal cord and out again without reaching brain. (Chusid and McDonald.)
From "Anatomy of Orofacial Structures", Brand, R.W., Isselhard, D.E.

Fig. 19-3 Multiple neurons at various levels necessary to complete conscious sensorimotor reaction.
From "Anatomy of Orofacial Structures", Brand, R.W., Isselhard, D.E.

(a) (b)

Figure E.3: Signal pathways. (a) Reflex arc: decisions are locally taken. (b) Conscious motor reaction: decisions are taken in the brain.

through the pons, which is frontal to the cerebellum. Then, the neural signals are relayed to the muscles, thereby moving them accordingly. Below the pons and above the spinal cord is the medulla oblongata. In addition to transmitting neural signals between the spinal cord and the brain, it also directly controls many involuntary muscular and glandular activities.

Efferent neurons are responsible for transmitting motor neural signals, see Fig E.2. Two categories can be distinguished: on the one hand, upper motor neurons, which are located in the brain, connect this to the spinal cord; on the other, lower motor neurons, carry the neural impulses from the upper motor neurons to muscle fibers.

Thus, upper motor neurons propagate neural signals through the central nervous system. For instance, a direct pathway from the layer V of the primary motor cortex to lower motor neurons —located in the ventral horn of the spinal cord— sends fine voluntary motor control signals and also controls voluntary body posture adjustments; another pathway from the motor cortex to the pons and medulla is involved in involuntary maintenance of body posture; or a pathway from the superior colliculus to lower motor neurons is responsible for involuntary adjustment of head position in response to visual information.

Lower motor neurons innervate two types of muscle fibers, and are therefore accordingly classified. On the one hand, *alpha motor neurons* innervate extrafusal muscle fibers, which are involved in contracting the muscle. On the other, *gamma motor neurons* innervate intrafusal muscle fibers, which are related to muscle spindles and the sense of body position. A muscle spindle is a specialised muscle structure innervated by both sensory and motor neuron axons. It is related to the capability of sensing the position, orientation and movement of the different parts of the body.

Thus, alpha motor-neurons —located in the anterior horn— effect the muscles, while sensory neurons, at the posterior horn, receives sense information. The latter are the nerve cells responsible for converting the external organism stimuli into internal electrical signals, thereby being a part of the reflex loops. These are usually located in the spinal cord.

A reflex arc is a neural pathway that allows reflex or involuntary actions, see Fig E.3.(a). By synapsing in the spinal cord, these pathways do not pass through the brain, and therefore can occur relatively quickly since the delay of routing the signal through the brain is avoided. Nevertheless, the brain receive the sensory signals for further cognitive processing, but this happens simultaneously to the reflex action, see Fig E.3.(b). Reflex arcs can be mono-synaptic or poly-synaptic. The former involves just a motor and a sensory neuron, while in the latter inter-neurons connect both afferent and efferent signals. This allows to process or inhibit the reflexes at spinal cord. See [29] for details.

Appendix F

Publications

F.1 Journals

1. On Tracking into Groups. D. Rowe, I. Rius, J. Gonzàlez and J. J. Villanueva. *Accepted with changes* in Pattern Recognition Letters, Elsevier, 2007.

2. Understanding Dynamic Scenes for Advanced Human-Computer Interaction. J. Gonzàlez, D. Rowe, X. Varona and X. Roca. *Accepted in* Image and Vision Computing, Elsevier, 2007.

3. Towards Robust Multiple-Target Tracking in Unconstrained Human-Populated Environments. D. Rowe, J. Gonzàlez, X. Roca and J. J. Villanueva. *Submitted to* Pattern Analysis and Machine Intelligence, IEEE, 2007.

4. A Hierarchical Taxonomy for Reviewing Detections and Tracking Approaches. D. Rowe, J. Gonzàlez, and J.J. Villanueva. *Submitted to* International Journal of Pattern Recognition and Artificial Intelligence, World Scientific Publishing, 2007.

5. Understanding Human Behavior from Image Sequences. P. Baiget, D. Rowe, X. Roca and J. Gonzàlez. *Submitted to* Machine Vision and Applications, Springer, 2007.

6. Multi-Cue Image-Segmentation by Fusing Colour, Intensity and Edges Cues. I. Huerta, D. Rowe, and J. Gonzàlez *To be submitted to* Machine Vision and Applications, Springer, 2008.

F.2 Conferences

1. On Reasoning over Tracking Events. D. Rowe, I. Huerta, J. Gonzàlez and J. J. Villanueva. In 15th SCIA, Aalborg, Denmark, pp. 502-511, Springer LNCS 4522, 2007.

2. Robust Multiple-People Tracking Using Colour-Based Particle Filters. D. Rowe, I. Huerta, J. Gonzàlez and J. J. Villanueva. In 3rd IbPRIA, Gerona, Spain, vol. 1, pp. 113-120, Springer LNCS 4477, 2007.
3. Improving Background Subtraction based on a Casuistry of Colour-Motion Segmentation Problems. I. Huerta, D. Rowe, J. Gonzàlez and M. Mozerov. In 3rd IbPRIA, Gerona, Spain, vol. 2, pp. 475-482, Springer LNCS 4478, 2007.
4. Unconstrained Multiple-people Tracking. D. Rowe, I. Reid, J. Gonzàlez and J. Villanueva. In 28th DAGM, Berlin, Germany, pp. 505-514, Springer LNCS 4174, 2006.
5. Efficient Incorporation of Motionless Foreground Objects for Adaptive Background Segmentation. I. Huerta. D. Rowe, J. Gonzàlez and J. J. Villanueva. In 4th AMDO, Mallorca, Spain, pp. 424-433, Springer LNCS 4069, 2006.
6. Improving Tracking by Handling Occlusions. D. Rowe, I. Rius, J. Gonzàlez, and J. J. Villanueva. In 3rd ICAPR, Bath, UK, vol. 2, pp. 384-393, Springer LNCS 3687, 2005.
7. 3D Action Modelling and Reconstruction for 2D Human Body Tracking. I. Rius, D. Rowe, J. Gonzàlez, and X. Roca. In 3rd ICAPR, Bath, UK, vol.2, pp. 146-154, Springer LNCS, 2005.
8. Robust Particle Filtering for Object Tracking. D. Rowe, I. Rius, J. Gonzàlez, and J. J. Villanueva. In 13th ICIAP, Cagliari, Italy, pp. 1158-1165, Springer LNCS 3617, 2005.
9. Probabilistic Image-based Tracking: Improving Particle Filters. D. Rowe, I. Rius, J. Gonzàlez, X. Roca, and J. J. Villanueva. In 2nd IbPRIA, Estoril, Portugal, vol. 1, pp. 85-92, Springer LNCS 3522, 2005.
10. A 3D Dynamic Model of Human Actions for Probabilistic Image Tracking. I. Rius, D. Rowe, J. Gonzàlez, and X. Roca. In 2nd IbPRIA, Estoril, Portugal, vol. 1, pp. 529-536, Springer LNCS, 2005.

F.3 Minor Conferences, Workshops and Technical Reports

1. Event-based Tracking Evaluation Metric. D. Roth, E. Koller-Meier, D. Rowe, T.B Moeslund, L. Van Gool.
   In press. IEEE Workshop on Motion and Video Computing, Colorado, USA, 2008.
2. A Hierarchical Architecture to Multiple Target Tracking. D. Rowe, I. Huerta, J. Gonzàlez, J. J. Villanueva. 2nd CVCRD, Computer Vision Centre, Barcelona, Spain, ISBN 84-93521-4-6, pp 111-116, 2007.
3. Background Subtraction by Fusing Colour, Intensity and Edges Cues. I. Huerta, D. Rowe, M. Mozerov, and J. Gonzàlez. 2nd CVCRD, Computer Vision Centre, Barcelona, Spain, ISBN 84-93521-4-6, pp 105-110, 2007.

4. Towards Robust Multiple-People Tracking in Unconstrained Environments. D Rowe. TR 100, Computer Vision Centre, Barcelona, Spain, 2007.
5. Detection and Tracking of Multiple Agents in Unconstrained Environments. D. Rowe, I. Huerta, J. Gonzàlez, J. J. Villanueva. 1st CVCRD, Computer Vision Centre, Barcelona, Spain, ISBN 84-933652-8-9, pp 69-74, 2006.
6. Improving Foreground Detection for Adaptive Background Segmentation. I. Huerta, D. Rowe, J. Gonzàlez, J. J. Villanueva. 1st CVCRD, Computer Vision Centre, Barcelona, Spain, ISBN 84-933652-8-9, pp 63-68, 2006.
7. Efficient Management of Multiple Agent Tracking through Observation Handling. J. Gonzàlez, D. Rowe, J. Andrade and J. J. Villanueva. In 6th IASTED VIIP, Mallorca, Spain, pp. 585-590, ACTA PRESS, 2006.
8. Probabilistic Image-based Tracking in Cluttered Human-populated Environments. D Rowe, TR 92, Computer Vision Centre, Barcelona, Spain, 2005.
9. Articulated Object Modelling Using Neural Gas Networks. S. García, D. Rowe, J. Gonzàlez, J. J. Villanueva. In 3rd IASTED BioMech, Benidorm, Spain, pp. 581-586, ACTA PRESS, 2005.

Bibliography

[1] J.K. Aggarwal and Q. Cai. Human Motion Analysis: A Review. *CVIU*, 73(3):428–440, 1999. (Cited on pages xi, 2, 15, 17, 18, and 22)

[2] B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, 1979. (Cited on page 30)

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on PFs for On-line Non-linear/Non-Gaussian Bayesian Tracking. *Signal Processing*, 50(2):174–188, 2002. (Cited on pages 31, 53, and 155)

[4] Y. Bar-Shalom and T. Fortran. *Tracking and Data Association*. A. Press, 1988. (Cited on page 30)

[5] A. Bhattacharyya. On a Measure of Divergence Between Two Statistical Populations Defined by Probability Distributions. *Bull. Calcutta Math. Soc*, 35:99–109, 1943. (Cited on page 33)

[6] C. Bibby and I. Reid. Visual Tracking at Sea. In *International Conference in Robotics and Automation*, pages 1841– 1846. IEEE, 2005. (Cited on pages 2 and 23)

[7] C. Bregler. Learning and Recognising Human Dynamics in Video Sequences. In *CVPR, Puerto Rico*, pages 568–574. IEEE, 1997. (Cited on pages 29 and 30)

[8] H. Buxton. Learning and Understanding Dynamic Scene Activity: A Review. *Image and Vision Computing*, 125-136(1):125–136, 2002. (Cited on pages 2 and 15)

[9] C. Cédras and M. Shah. Motion-based Recognition: A Survey. *Image and Vision Computing*, 13(2):129–155, 1995. (Cited on pages 2 and 15)

[10] R. Collins. Mean-shift Blob Tracking through Scale Space. In *CVPR, Madison, WI, USA*, volume 2, pages 234–240. IEEE, 2003. (Cited on page 34)

[11] R. Collins, A. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring. In *8th International Topical Meeting on Robotics and Remote Systems, Pittsburgh, USA*, pages 1–15. American Nuclear Society, 1999. (Cited on pages 2 and 28)

[12] R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631–1643, 2005. (Cited on pages 34, 35, 110, 111, and 115)

[13] R.T. Collins, A.J. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring: VSAM Final Report. Technical Report TR00-12, CMU, 2000. (Cited on pages 28 and 30)

[14] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564–577, 2003. (Cited on pages xi, 34, 35, 75, 76, 97, 110, 111, 118, 120, 121, 123, and 173)

[15] N. de Freitas, A. Gee M. Niranjan, and A. Doucet. Sequential Monte Carlo Methods for Optimisation of Neural Network Models. Technical Report TR 328, Cambridge University, 1998. (Cited on pages 32 and 53)

[16] J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185–205, 2005. (Cited on pages 12, 33, 34, 53, and 155)

[17] A.K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001. (Cited on page 4)

[18] A. Doucet. On Sequential Simulation-Based Methods for Bayesian Filtering. Technical Report TR310, Cambridge University, 1998. (Cited on pages 31, 53, 56, 62, and 65)

[19] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlang, first edition, 2001. (Cited on page 53)

[20] J. Elder, s. Prince, Y. Hou, M. Sizintsev, and E. Olevskiy. Pre-Attentive and Attentive Detection of Humans in Wide-Field Scenes. *IJCV*, 72(1):47–66, 2007. (Cited on page 85)

[21] G.D. Finlayson, S.D. Hordley, C. Lu, and M.S. Drew. On the removal of shadows from images. *Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006. (Cited on pages 27 and 140)

[22] D. Fox. Adapting the Sample Sieze in Particle Filters Through KLD-Sampling. *Int. Journal of Robotics Research*, 22(12):985–1004, 2003. (Cited on page 68)

[23] D. M. Gavrila. The Visual Analysis of Human Movement: A Survey. *CVIU*, 73(1):82–98, 1999. (Cited on pages xi, 2, 10, 15, 17, 18, 19, and 22)

[24] R. Gnanadesikan. *Methods for the Statistical Data Analysis of Multivariate Observations*. Wiley, 1977. (Cited on page 102)

[25] J. Gonzàlez. *Human Sequence Evaluation: The Key-frame Approach*. PhD thesis, UAB, Spain, 2004. (Cited on pages 2, 16, 39, and 50)

[26] J. Gonzàlez, D. Rowe, J. Andrade, and J.J. Villanueva. Efficient Management of Multiple Agent Tracking through Observation Handling. In *6th VIIP, Mallorca, Spain*, pages 585–590. IASTED, 2006. (Cited on page 84)

[27] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian Bayesian state estimation. In *IEE Proceedings-F*, volume 140, pages 107–113, 1993. (Cited on pages 31, 53, and 62)

[28] U. Grenander, Y. Chow, and D. M. Keenan. *HANDS. A Pattern Theoretical Study of Biological Shapes.* Springer-Verlang, 1991. (Cited on page 62)

[29] D. Haines. *Neuroanatomy: an Atlas of Structures, Sections and Systems.* Lippicott, Williams and Wilkins, 6th edition, 2004. (Cited on pages 86 and 199)

[30] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, 2000. (Cited on pages xi, 2, 26, 30, and 91)

[31] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. In *Third International Conference on Automatic Gesture and Face Recognition, Nara, Japan*, pages 222–227. IEEE, 1998. (Cited on page 26)

[32] C. G. Harris and M. Stephens. A Combined Corner and Edge Detector. In *4th Alvey Vision Conf. Manchester, UK*, pages 147–151, 1988. (Cited on page 23)

[33] J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In *2nd Workshop on Visual Surveillance, Washington DC, USA*, pages 74–81. IEEE, 1999. (Cited on pages 2 and 25)

[34] T. Horprasert, I. Haritaoglu, D. Harwood, L. Davis, C. Wren, and A. Pentland. Real-Time 3D Motion Capture. In *2nd Workshop Perceptual Interfaces*, 1998. (Cited on page 26)

[35] T. Horprasert, D. Harwood, and L. Davis. A Robust Background Subtraction and Shadow Detection. In *4th ACCV, Taipei, Taiwan*, volume 1, pages 983–988, 2000. (Cited on pages xi, 26, 27, and 30)

[36] I. Huerta, D. Rowe, M. Mozerov, and J. Gonzàlez. Background Subtraction by Fusing Colour, Intensity and Edges Cues. In *2nd CVCRD, Computer Vision Centre, Barcelona, Spain*, pages 105–110, 2007. (Cited on page 178)

[37] I. Huerta, D. Rowe, M. Mozerov, and J. Gonzàlez. Improving Background Subtraction based on a Casuistry of Colour-Motion Segmentation Problems. In *3rd IbPRIA, Gerona, Spain*, volume 2, pages 475–482. Springer LNCS, 2007. (Cited on pages 84 and 87)

[38] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *4th ECCV, Cambridge UK*, volume 1, pages 343–356. Springer-Verlang, 1996. (Cited on pages xi, 12, 32, 33, 53, and 63)

[39] M. Isard and A. Blake. A Mixed State Condensation Tracker with Automatic Model Switching. In *6th ICCV, Bombay, India*, pages 107–112, 1998. (Cited on page 33)

[40] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998. (Cited on pages 33, 35, 53, 61, 63, 68, and 155)

[41] M. Isard and A. Blake. Icondensation Unifying Low-level and High-level Tracking in a Stochastic Framework. In *5th ECCV, Freiburg, Germany*, volume 1, pages 893–908, 1998. (Cited on pages 35, 53, 70, and 159)

[42] M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001. (Cited on pages 34 and 53)

[43] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Workshop on Motion and Video Computing*, pages 22–27. IEEE, 2002. (Cited on pages 28 and 30)

[44] B. Julesz. Early Vision and Focal Attention. *Rev. Modern Phys*, 63:635–772, 1991. (Cited on page 84)

[45] S. Julier and J. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *11th AeroSense, Orlando, Florida*, volume 3068, pages 182–193, 1997. (Cited on page 30)

[46] R. Kahn, M. Swain, P. Prokopowicz, and R. Firby. Gesture Recognition Using the Perseus Architecture. In *CVPR, San Francisco, USA*, pages 734–741. IEEE, 1996. (Cited on pages 2, 22, and 35)

[47] R. E. Kahn, M. J. Swain, P.N. Prokopowicz, and R.J. Firby. Real-time Gesture Recognition with the Perseus System. Technical Report TR96-04, University of Chicago, 1996. (Cited on page 35)

[48] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *ASME–Journal of Basic Engineering*, 82(D):35–45, 1960. (Cited on pages 12, 30, 53, and 191)

[49] T. Kanade. Region Segmentation: Signal vs Semantics. *Computer Graphics and Image Processing*, 13:279–297, 1980. (Cited on page 2)

[50] M. Karaman, L. Goldmann, D. Yu, and T. Sikora. Comparison of static background segmentation methods. In *Visual Communications and Image Processing*, volume 5960, pages 2140–2151. SPIE, 2005. (Cited on page 25)

[51] D. G. Kelly. *Introduction to Probability*. Mac Millan, 1994. (Cited on page 54)

[52] O. King and D. Forsyth. How Does CONDENSATION Behave with a Finite Number of Samples? In *6th ECCV, Ireland*, volume 1, pages 695–709, 2000. (Cited on pages 31, 53, and 68)

[53] E.B. Koller-Meier and F. Ade. Tracking Multiple Objects Using the Condensation Algorithm. *Robotics and Autonomous Systems*, 34:93–105, 2001. (Cited on page 33)

[54] R. Llinas. *I of the Vortex: From Neurons to Self*. MIT Press, 2002. (Cited on pages 1 and 15)

[55] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. (Cited on page 23)

[56] J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. In *7th ICCV, Kerkyra, Greece*, volume 1, pages 572–578. IEEE, 1999. (Cited on page 33)

[57] J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV*, 39(1):57–71, 2000. (Cited on pages 33, 53, 69, 125, and 155)

[58] J. MacCormick and M. Isard. Partioned Sampling, Articulated Objects, and Interface-quality Hand Tracking. In *6th ECCV, Dublin, Ireland*, volume 2, pages 3–19. Springer-Verlang, 2000. (Cited on pages 33, 53, 155, and 159)

[59] D. Mackay. *Introduction to Monte Carlo Methods*, chapter 7, pages 175–204. MIT Press, 1998. (Cited on pages 53 and 54)

[60] T. Matsuyama and V. Hwang. *SIGMA A Knowledge Based Aerial Image Understanding System*. Plenum Press, 1990. (Cited on pages 83 and 144)

[61] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *CVIU*, 80(1):42–56, 2000. (Cited on pages 23, 26, 30, 91, and 110)

[62] T. Moeslund, A. Hilton, and V. Krüger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. *CVIU*, 104:90–126, 2006. (Cited on pages 2, 15, 17, and 19)

[63] T. B. Moeslund and Erik Granum. A Survey of Computer Vision-Based Human Motion Capture. *CVIU*, 81(3):231–268, 2001. (Cited on pages xi, 1, 10, 15, 17, 19, 20, and 22)

[64] J. Moustakas, K. Marias, S. Dimitriadis, and S. Orphanoudakis. A Two-Level CBIR Platform with Application to Brain MRI Retrieval. In *International Conference on Multimedia and Expo*, pages 1278–1281. IEEE, 2005. (Cited on page 85)

[65] H. Nagel. Image Sequence Evaluation: 30 years and still going strong. In *15th ICPR, Barcelona, Spain*, volume 1, pages 149–158. IEEE, 2000. (Cited on pages 2 and 15)

[66] S. Nishida, T. Ledgeway, and M. Edwards. Dual Multiple-scale Processing for Motion in the Human Visual System. *Vision Research*, 37:2685–2698, 1997. (Cited on page 86)

[67] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *Image and Vision Computing*, 21(1):99–110, 2003. (Cited on pages xi, 12, 33, 53, 75, 76, 97, 110, 111, and 118)

[68] T. V. Papathomas. *Early Vision and Beyond*. MIT Press, 1994. (Cited on pages 1, 15, 84, and 196)

[69] A. Pentland. Looking at people: Sensing for Ubiquitous and Wearable Computing. *PAMI*, 22(1):107–119, 2000. (Cited on pages 2, 10, 15, and 21)

[70] J. Philbina, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, Minneapolis, USA*, pages 1–8. IEEE, 2007. (Cited on page 2)

[71] M. Piccardi. Background subtraction techniques: a review. In *International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104. IEEE, 2004. (Cited on page 25)

[72] S. Pingali and J. Segen. Performance Evaluation of People Tracking Systems. In *3rd Workshop on Applications of Computer Vision, Sarasota, USA*, pages 33–38. IEEE, 1996. (Cited on page 146)

[73] F. Porikli. Achieving Real-time Object Detection and tracking under Extreme Conditions. *Real Time Image Processing*, 1:33–40, 2006. (Cited on page 144)

[74] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *7th ECCV, Copenhaguen, Denmark*, pages 661–675. Springer-Verlang, 2002. (Cited on pages 33, 53, and 69)

[75] A. Pujol, F. Lumbreras, X. Varona, and J.J. Villanueva. Locating People in Indoor Scenes for Real Applications. In *15th International Conference on Pattern Recognition*, volume 4, pages 632–635. IEEE, 2000. (Cited on page 2)

[76] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter*. Artech House Publising, first edition, 2004. (Cited on page 53)

[77] S. M. Ross. *Simulation*. Academic Press, 2nd edition, 1997. (Cited on pages 53 and 54)

[78] D. Rowe, J. Gonzàlez, I. Huerta, and J.J. Villanueva. On Reasoning over Tracking Events. In *15th SCIA, Aalborg, Denmark*, pages 502–511. Springer LNCS, 2007. (Cited on page 84)

[79] D. Rowe, I. Reid, J. Gonzàlez, and J. Villanueva. Unconstrained Multiple-people Tracking. In *28th DAGM, Berlin, Germany*, pages 505–514. Springer LNCS, 2006. (Cited on page 84)

[80] R. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach*, chapter 13-15. Prentice Hall, 2nd edition, 2003. (Cited on pages 12, 53, 54, 55, 56, and 191)

[81] A. Senior, A. Hampapur, Y.L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance Models for Occlusion Handling. *Image and Vision Computing*, 24:1233–1243, 2006. (Cited on pages 22, 35, 91, and 146)

[82] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio. A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex. Technical Report AI Memo 2005-036, MIT, 2005. (Cited on page 85)

[83] J. Shen. Motion detection in color image sequence and shadow elimination. In *Visual Communications and Image Processing, California, USA*, volume 5308, pages 731–740. SPIE, 2004. (Cited on pages 23, 29, and 30)

[84] H. Sidenbladh, M.J. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *7th ECCV, Copenhagen, Denmark*, volume 1, pages 784–800. Springer-Verlang, 2002. (Cited on page 2)

[85] C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3d body Tracking. In *CVPR, Hawaii, USA*, pages 447–454. IEEE, 2001. (Cited on page 155)

[86] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR, Fort Collins, CO, USA*, volume 2, pages 246–252. IEEE, 1999. (Cited on pages 27, 30, and 91)

[87] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Bayesian Object Localisation in Images. *International Journal of Computer Vision*, 44(2):111–135, 2001. (Cited on page 33)

[88] C. Urtubia. *Neurobilogía de la Visión*. Edicions UPC, 2nd edition, 1999. (Cited on pages 195 and 196)

[89] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The Unscented Particle Filter. Technical Report TR380, Cambridge University, 2000. (Cited on pages 31, 32, 53, and 155)

[90] X. Varona. *Seguimiento Visual Robusto en Entornos Complejos*. PhD thesis, Universitat Autònoma de Barcelona, Spain, 2001 (in Spanish). (Cited on pages 70 and 175)

[91] X. Varona, J. Gonzàlez, X. Roca, and J. Villanueva. iTrack: Image-based Probabilistic Tracking of People. In *15th ICPR, Barcelona, Spain*, volume 3, pages 1110–1113. IEEE, 2000. (Cited on pages 12, 53, 70, and 159)

[92] E. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Adaptive Systems for Signal Processing, Communication and Control, Lake Louise, Canada*, pages 153–158. IEEE, 2000. (Cited on page 30)

[93] L. Wang, W. Hu, and T. Tan. Recent Developments in Human Motion Analysis. *Pattern Recognition*, 36(3):585–601, 2003. (Cited on pages xi, 2, 15, 17, 20, 21, and 22)

[94] C. R. Wren, A. Azarbayejani, T. Darrell, and A.Pentland. Pfinder: Real-Time Tracking of the Human Body. *PAMI*, 19(7):780–785, 1997. (Cited on pages xi, 23, 25, and 30)

[95] Y. Wu, T. Yu, and G. Hua. Tracking Appearances with Occlusions. In *CVPR, Wisconsin, USA*, volume 1, pages 789–795. IEEE, 2003. (Cited on pages 34 and 53)

[96] M. Xu, J. Orwell, L. Lowey, and D. Thirde. Architecture and algorithms for tracking football players with multiple cameras. *Vision, Image and Signal Processing*, 152(2):232–241, 2005. (Cited on page 2)

[97] T. Yang, S. Li, Q. Pan, and J. Li. Real-time Multiple Object Tracking with Occlusion Handling in Dynamic Scenes. In *CVPR, San Diego, USA*, volume 1, pages 970–975. IEEE, 2005. (Cited on page 35)

[98] T. Zhao and R. Nevatia. Tracking Multiple Humans in Complex Situations. *PAMI*, 26(9):1208–1221, 2004. (Cited on page 36)

[99] T. Zhao and R. Nevatia. Tracking Multiple Humans in Crowded Environments. In *CVPR, Washington, USA*, volume 2, pages 406–413. IEEE, 2004. (Cited on page 146)

Natural Vision Systems have reached incredible performances in detecting and tracking multiple moving objects simultaneously. Accurate and robust multiple-target tracking is also a key task in many promising Computer-Vision applications. Practical usages of proposed algorithms can now be tackled in real time thanks to recent technological advances. Further, this represents a huge challenge because of the numerous particular problems involved in such a task. Thus, proposals must deal with multiple highly non-rigid targets which move in an unforeseeable manner through unconstrained dynamic open-world scenarios.

In this thesis, a principled hierarchical architecture which fulfills multiple-target tracking is presented. Further, another tracking approach —based on *particle filtering*— is previously developed and evaluated. Thus, a modular and hierarchically-organised system is designed. It is conformed by a detection level which feeds a two-level tracking subsystem. Co-operating modules, distributed through this architecture, work following both bottom-up and top-down approaches. Contributions include both the architecture itself, and the development, improvement and integration of the different modules. The proposed architecture introduces the necessary synergies which allow the system to tackle such a problem as unconstrained multiple-target tracking.

With respect to the different modules, the main focus is placed on high-level tracking algorithms. Since a careful analysis of motion events is a critical issue for tracking successful, a module for principled event management is proposed, and embedded in the system. Multiple-target interaction events, and a proper scheme for tracker instantiation and removal according to scene events, are considered. Thus, the system is allowed to switch among the two different operation modes implemented, motion-based tracking and appearance-based tracking. This entails another remarkable characteristic of the system: its ability to continuous and independently track numerous targets while they group and split. Multiple appearance models are built and constantly updated. A special attention is paid to maximise the discrimination between the target and potential distracters by means of an appropriate feature selection, and a wise combination of all available sources of information.

It works as a stand-alone application in a non-friendly, complex and dynamic scenario. No a-priori knowledge about either the scene or the targets, based on a previous off-line training period is needed. No camera calibration is required since tracking is achieved without the need of 3D information.

Successful tracking has been demonstrated in multiple sequences of both indoor and outdoor scenarios. Accurate and robust localisations have been yielded even during long-term target clustering and occlusions. Results are comprehensively analysed.