Philosophy Doctorate Thesis

FEATURE-BASED HUMAN TRACKING: FROM COARSE TO FINE

Jesús Martínez del Rincón

PhD. Supervisor
Carlos Orrite Uruñuela

October 16, 2008

For my parents Jesús and María Paz,
for my sister Mónica.

"Keep your dreams alive. Understand to achieve anything requires faith and belief in yourself, vision, hard work, determination, and dedication. Remember all things are possible for those who believe."

-Gail Devers-

Resumen

Un sistema de seguimiento robusto es la base sobre la que se sustentan las aplicaciones de análisis de movimiento humano. La presente tesis describe la problemática del seguimiento de personas desde tres puntos de vista: dominio puntual, regional y modelado de la figura humana. En el primer dominio, se considera al sujeto como un objeto compacto, pequeño y rígido. En el segundo, es posible modelarlo como un conjunto de regiones interrelacionadas, cada una de las cuales con sus propias propiedades, y cuya unión identifica al sujeto unívocamente. El último punto de vista nos permite identificar a la persona como un objeto no rígido, con propiedades morfológicas y dinámicas intrínsecamente humanas. Nuestro objetivo consiste en el estudio y entendimiento de estos tres diferentes estados en los cuales el ser humano puede ser modelado, así como en la selección del más adecuado, dependiendo de la aplicación y la problemática a resolver. La problemática fijará las necesidades e impondrá las limitaciones para cada momento, bien sean debidas a factores internos (múltiples objetos, oclusiones, posición de la cámara) o externos (coste, viabilidad técnica). Por tanto, el énfasis de este trabajo reside en el desarrollo de un sistema capaz de seguir una o múltiples personas en secuencias de vídeo, con diferentes grados de entendimiento dependiendo de la aplicación específica a la que vaya dirigido, las necesidades y los medios disponibles. Para alcanzar esta meta, el proyecto se centrará en seguimiento de personas, modelado de la figura humana y extracción de características. El sistema debe ser capaz de trabajar en escenarios reales, tales como vídeo-vigilancia, análisis deportivo o diagnosis médica.

En el dominio puntual, se ha implementado un sistema capaz de seguir personas satisfactoriamente, a pesar de malas medidas o sensores de baja calidad, gracias a la combinación de un detector de objetos estáticos, un estimador de la altura y un algoritmo de fusión de múltiples cámaras. El sistema ha sido diseñado para aplicaciones de vídeo-vigilancia, e incluso la calibración de las cámaras ha sido simplificada todo lo posible para satisfacer dicho requisito.

Se ha hecho especial énfasis en la extracción de características. El modelado del color, usando técnicas paramétricas y no paramétricas, ha sido la clave para el seguimiento de los objetivos, debido a la flexibilidad e invarianza que esta característica proporciona. Además, se ha implementado un sistema robusto de actualización del modelo de color, el cual es capaz de adaptarse tanto a cambios rápidos como progresivos de las condiciones de iluminación.

Se ha propuesto un algoritmo eficiente de seguimiento basado en color para acelerar el cálculo de los algoritmos basaos en múltiples hipótesis. La inclusión de técnicas tales como *partitioned sampling* o *importance sampling* reduce el número de muestras a evaluar, ya que aquellas hipótesis con baja probabilidad son descartadas. Además, el uso de la imagen integral en el proceso de evaluación minimiza significativamente el tiempo de cómputo para evaluar cada hipótesis.

Una incursión en el seguimiento de múltiples objetivos idénticos ha sido realizada. Técnicas de asociación y modelado de interacciones han sido propuestas para el tratamiento de la coalescencia y ayudar a la resolución de ambigüedades por parte del seguimiento. La gran complejidad del seguimiento multi-objeto también ha demandado la creación de un entorno integrado donde la información proveniente de múltiples sensores es conjugada.

Por último, se han empleado modelos articulados para el seguimiento, no solo del movimiento global, sino también de los movimientos relativos de las extremidades. En este ámbito, modelos 2D han sido propuestos debido a su mayor adecuación para propósitos de vigilancia, siendo capaces de trabajar en secuencias monoculares, tener una menor carga computacional y requerir una simple inicialización. El principal inconveniente que ha relegado a estas técnicas, es decir, su dependencia del punto de vista, ha sido tratado en profundidad. Información morfológica y biomecánica, introducida bien como parte del modelo o por medio de restricciones, permite la consecución de esta meta. Las dos posibles metodologías a aplicar, generativas o discriminatorias, han sido testeadas y comparadas.

Palabras clave: Seguimiento, objetivos múltiples, multi cámara, extracción de características, filtro de partículas, modelos articulados, análisis deportivo, vídeo-vigilancia, análisis del movimiento humano.

Abstract

A robust tracking algorithm is the cornerstone of human motion analysis applications. This thesis describes the tracking problem from three points of view: punctual, regional and human pose modelling. In the first one, the subject is considered as a compact object, small and rigid. In the second one, it is possible to model it as a set of connected regions, each one with its own properties, and which junction identifies the subject. The last point of view allows us to identify the person as a non-rigid object, with morphologic and dynamic properties which are intrinsically human. Our goal consists in the study and understanding of these three different stages in which the human being can be modelled, as well as the selection of the most suitable stage depending on the application and the problem to be solved. The problem will set the necessities and impose the limitations in each time, due to internal factors (multiple target, occlusions, camera location) or external factors (cost, technical feasibility). Therefore, the emphasis in this work lies on producing a system capable of tracking one or multiple people in video sequences, with different degrees of understanding depending on the specific application, the necessities and the available means. In order to achieve this goal, the project will focus on human tracking, human modelling and feature extraction. The system should be able to work in real life scenarios, such as video surveillance, sport analysis or medical diagnosis.

In punctual domain, a system capable of tracking people successfully has been developed, in spite of bad measurements or poor quality sensors, thanks to the combination of a static object detector, a height estimator and a multicamera conjugation algorithm. The system has been designed for surveillance applications and even the camera calibration has been simplified as much as possible to fulfil this requirement.

Regarding the feature extraction field, special emphasis on feature extraction has been done. Colour modelling, using parametric and non-parametric methodologies, has been the basic clue to track the targets due to the generality and invariance that this features provides. In addition, a robust colour update technique has been presented, which is able to adapt itself to both fast and slow changing illumination conditions.

An efficient colour tracking algorithm, based on particle filter, has been proposed to speed up the computation of the conventional version of this multi-hypothesis algorithm. The inclusion of techniques such a partitioned or importance sampling reduces the number of samples since they discard those hypotheses with low probability. Furthermore, the usage of the integral image in the evaluation procedure minimises significatively the computational time of evaluating each hypothesis.

An incursion in tracking of multiple identical targets has been done. Association techniques and interaction modelling have been proposed to deal with the coalescence and help the tracking to solve ambiguities. The high complexity of multi-target tracking has also demanded the creation of a integrated framework where multiple sensor information is conjugated.

Finally, articulated models have been employed to track not only the global motion of the target but also the relative motion of the limbs. In this field, 2d models have been proposed due to the fact that they are much more adequate for surveillance purposes, being able to work in monocular sequences, have a lighter computational load and require a simple initialistaion. The main drawback that has relegated this techniques, i.e the viewpoint dependence, has been tackled in depth. Morphologic and biomechanical information, introduced as part of the model or by means of constraints, allows the achievement of this goal. The two possible methodologies, discriminative and generative approaches, have been tested and compared.

Keywords: Tracking, multiple targets, multi camera, feature extraction, particle filter, articulated models, sport analysis, video-surveillance, human gait analysis

Acknowledgements

I would like to thank my supervisor at University of Zaragoza Dr. Carlos Orrite for his guidance and support. He has been always opened for discussions and thanks to him I have grown not only professionally but also personally. I should also like to extend my deepest gratitude to Dr. Jose Elías Herrero, who has been a second mentor for me, for all his help over the years beyond the call of duty.

I would also like to express my thanks to my supervisors at Kingston University, Dr. Jean Christophe Nebel and Dr. Dimitrios Makris. I gratefully acknowledge the precious opportunity I got to collaborate with the DIRC group to Prof. Graeme A. Jones. I highly appreciate to have taken part of the DIRC group as one of you. Thanks to all the members.

I want to mention my colleagues at the University of Zaragoza, who have offered me their support and help. Thanks Andrés, Carlos, Dominique, Fabienne, Fernando, Greg, Jorge, María José, Mario, Miguel, Nacho, Nico, Rubén and Tere. This work without coffee breaks would have been much harder.

I specially want to mention my friends in Zaragoza for their never ending and unconditional support and encouragement. Thanks for being there in the good and in the not so good moments, thanks Adrian, Fran, Jesús, Juan Diego, Juan Manuel, Paco, Rubén and Roberto. I do not want to forget my friends in London, they have already become a part of my family, especially Alberto, Alicia, Antonin, Amin, Aurelie, George, Gonzalo, Joanna, Norbert, Marina, Pawel and Zhen.

Thanks Mila for being the light at the end of the tunnel, for giving me another reason to finish.

Pero por encima de todo, gracias a mi familia, a mis padres Jesús y María Paz así como a mi hermana Mónica, sin vosotros no habría llegado hasta aquí, sin vosotros ni siquiera sería el hombre que soy.

The author would also like to thank I3A and Electronics and Communications Engineering Department at University of Zaragoza for their confidence. Thanks to the Goverment of Aragón and Real Zaragoza S.A.D for their colaboration and economical support in ASTRO Project. This thesis has been partially supported by grant TIN2006-11044 and FEDER from Spanish Ministry of Education. Jesús Martínez del Rincón was supported by a FPI grant BES- 2004-3741 from the Spanish Ministry of Education.

Contents

1

Introduction

"A narrow vision is divisive, a broad vision expansive. But a divine vision is all-inclusive."

-H. H. Swami Tejomayananda-

Detection and tracking of human beings using computer vision techniques require different approaches and degrees of understanding depending on the necessities and the technical feasibility. Thus, in this thesis, we propose to deal with the problem from three points of view: punctual, regional and human pose modelling (see Fig. 1.1). In the first one, the subject is considered as a compact object, small and rigid. In the second one, it is possible to model it as a set of connected regions, each one with its own properties, and which junction identifies the subject. The last point of view allows us to identify the person as a non-rigid object, with morphologic and dynamic properties which are intrinsically human.

Our goal consists in the study and understanding of these three different stages in which the human being can be modelled, as well as the selection of the most suitable stage depending on the application and the problem to be solved. The problem will set the necessities and impose the limitations in each time step, due to internal factors (multiple target, occlusions, camera location) or external factors (cost, technical feasibility).

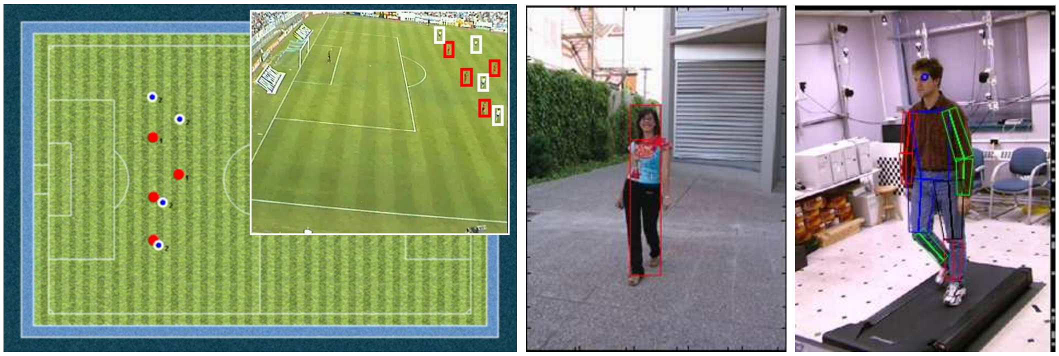


Figure 1.1: Tracking examples for each one of the different levels in which we have divided the tracking domain.

Because human tracking has significant intersection with human motion analysis, feature segmentation and human modelling, concepts related to these fields are also introduced and developed in this thesis. Specially the feature segmentation, being tackled in depth, since the type of measurements sets the requirements and characteristics of our tracking algorithm, and therefore, the level of understanding in which our tracking algorithm must operate.

1.1 Historical Introduction and Future Perspective. Framework of the Thesis

Although computer vision bases were established a long time ago, and most of the algorithms currently employed were proposed in the eighties, only the recent development in the microelectronic field has made possible the generalisation of these techniques. The increasing of both interest and resources in this field has produced a growing in the number of publication and researches in the last twenty years.

Last years have produced not only a deeper study of the known techniques and a better understanding of the problem but also new viewpoints and approaches to solve the challenge that this scientific field involves.

The deep study that has been carried out has shown that the absolute solution is further than it could be thought when this technology started. However, each new approach, each new solution even though it produced disappointing results, brings us a little bit near.

Nowadays, no universal solution exists which can solve a problem in any scenario, but specific applications have been developed which work properly given extraordinary results under a set of constraints.

A big effort must be applied to provide the current systems with the required robustness and accuracy as well as capacity to adapt themselves to new context and new necessities. Predictions point to a bigger impact of technology and specifically computer vision in our daily living, in the way in which we live and work. This interaction will be each time more and more intelligent, natural and imperceptible. The objective in the future is the transparency of all these devices and their complete automatisation, removing the participation of humans in the process.

This thesis is placed in this framework and it has required the specialisation in one of the parts that compose such a difficult problem. Thus, it tries to be a small contribution in this complex scientific field. The chosen topic is human tracking algorithms, which are a crucial stage as we will see through this work. To achieve this goal, an exhaustive study of the current methodologies has been done, and new solutions are proposed to improve the results, taking into account the limitations of the current computational systems.

1.2 Importance and Application Field

Tracking problem may imply the recognition and understanding of human being and human behaviour if we want to solve the problem at its highest level. Because of that, even for a human being, it is a difficult task in which confusions and misinterpretations have a high frequency of appearance.

In a globalised world, where the fear of crime and terrorist attacks and the concern about safety are more and more widespread, the number of cameras has been increased exponentially. However, nowadays the risk detection and video monitoring are responsibility of the human security personnel since automatic systems are not developed enough for general purpose use and only work for specific solution. Given the high level of concentration that the observation task requires as well as the huge number of cameras installed (over 5 millions only in England), it seems clear the motivation to develop more advanced automatic video-surveillance systems which aid the human personnel to monitor and detect dangerous situations.

In addition, the incorporation of technology in our daily living requires more comfortable and natural ways of communication with the computers. Vision, as well as speech recognition, are postulated as the future interfaces to establish a new order of understanding between humans and machines, removing the barriers of complexity that the use of technologies implies.

Using vision to track and analyse the human behaviour is one of the hottest issues in computer science, mainly due to its potential applications. This application filed covers different areas such as:

- Human-computer interaction
- Advance interface design
- Robot learning
- Video surveillance
- Sport analysis
- Medical diagnosis
- Traffic monitoring
- Ambient intelligence
- Patient remote assistance
- Elderly people video-attendance
- Gesture recognition and sign language
- Virtual reality and multimedia
- Etc...

Object tracking in video streams is one of the most popular topics in computer vision. The importance of tracking is crucial in human motion analysis since it is used for preparing and serving the data for subsequent pose estimation or action recognition applications. These high level applications requires a temporal coherence to work properly, being the goal of tracking to provide it. For instance, on surveillance applications, tracking is the fundamental component since subjects must be tracked before starting the action recognition. Human detection and human pose recovery are not solved problem yet and they can only work under specific conditions. For that, once the target has been detected, our objective will be to keep this target under control and this is the job of the tracking algorithms.

1.3 Aims and Objectives

A robust tracking algorithm is the cornerstone of human motion analysis applications. Our objective in this thesis is to take a step forward in the field of computer vision, specifically the development of tracking algorithms. By exploring this field at all the different levels in which it can be stated, a global vision of the problem is acquired and specific solutions for each one of the weak points can be proposed and tested.

The aim of this project is to produce a system capable of tracking one or multiple people in video sequences, at different degrees of understanding depending on the specific application, the necessities and the available resources. In order to achieve this goal, the thesis will focus on human tracking, human modelling and feature extraction. The system should be able to work in real life scenarios, such as video surveillance, sport analysis or medical diagnosis. The three understanding levels are tackled in depth and knowledge of each of them is fed back to the others in order to obtain a competitive advantage. However, an unified framework where the tracking domain is selected automatically is not provide in this thesis.

More specifically, the key issues which require addressing can be summarised as the following scientific objectives:

- To obtain a global vision of human tracking in its different understanding levels, adding innovation for each one.
- To develop feature detection algorithms suitable for detecting people in accordance with the quality of the camera.
- To develop multi-camera punctual tracking algorithms which allow us to track the target even with poor visual measurements.
- To obtain a tracking methodology which combines efficiently different kinds of features to identify and track one or more targets.
- To develop efficient multi-target tracking algorithms capable of dealing with occlusions and ambiguity.
- To implement articulated model tracking to recover the human pose in video sequences.

1.4 Methodology

Before giving the detail of the procedure to attain each one of the previous aims, we want to state the process that we consider correct in researching. This general methodology is based on the scientific method and it guarantees the progress of the science for any scientific research in any scientific field. Because we understand the science as a process of trial and error, the methodology should be iterative and be immersed in a refining procedure (see Algorithm 1).

1.4.1 General Methodology

Once we have described our philosophy of work, we proceed to specify the concrete milestones, grouped into phases, to achieve the different goals that have been proposed.

1.4.2 Specific Methodology

The project plan is splitted into 3 phases and in each one; milestones indicate the time plan and the deliverables of each phase.

1.4.2.1 Phase 1 - Punctual domain tracking

Punctual tracking was historically the beginning of object tracking [Aggarwal, 1987; Akita, 1984]. By making independent the tracking from the segmentation, tracking algorithms evolve towards more complex methodologies. However, although they are techniques of limited utility, they are useful yet. This utility is especially obvious in constrained scenarios, simple

Algorithm 1: Scientific Method

```
1. Problem Identification.
2. Approach, research and analysis of the problem.
    (a) Bibliographic revision.
    (b) Scrutiny of the existing algorithm in the state of the art, and selection of the most
        suitable ones for the particular problem.
    (c) Analysis of results, and extraction of limitations and drawbacks.
3. Goal description.
4. while (Hypothesis false) OR (No goal achievement)
    (a) Hypothesis proposal
    (b) Implementation and parameter optimisation. Experiment design.
    (c) Validation and comparison with the previous methods and assessment of
        compliance with goals.
    (d) Discussion and conclusions.
   end
5. Publication and dissemination of scientific knowledge.
```

applications, real-time process or bad quality, where more sophisticated techniques fail because they do not have at their disposal the required information.

Milestone M1: Develop a robust method to segment people under bad quality conditions

Under bad conditions, some features, such as colour, are not useful because of the similarity between targets or targets and background. In addition, colour requires an initial target model to be compared, which is not always possible (depending on the specific application). For this reason motion or gradients are much more adequate features. Additional improvements like shadow removal algorithms or background updating will be implemented. Even with the inclusion of the previous improvements, many false positive may appear. Given the nature of punctual tracking, in which the measurements should be reduced to a point and matched with the trackers, the presence of false positive is a serious problem which increases the complexity of matching algorithms, like auction algorithm [Bertsekas and Castanon, 1989]. In order to reduce the trouble, we propose the inclusion of knowledge about the scenario which allows us to discard a big amount of false detections. Calibration techniques and modelling of the scenario are important tools to carry out this approach.

Milestone M2: Implement a tracking framework suitable for bad quality scenarios

The specific domain of punctual tracking implies the splitting between the segmentation and filtering stages, where the responsibility in the performance of the algorithm relies on the segmentation. Nevertheless, several techniques can be implemented to cope with the limitation of this domain, like the extended or unscented versions of the conventional Kalman filter.

A different and complementary solution consists in the redundancy of hardware for solving the conventional problems. Multi-sensor systems have been shown as a way to solve tracking problems in overcrowded environments. Modifications of classic algorithms for accepting multi-camera recording systems will be proposed.

1.4.2.2 Phase 2 - Regional domain tracking

Due to the limitations of punctual tracking to deal with occlusions, specially in monocular sequences and non-calibrated environments, we propose a tracking based on Monte Carlo algorithms. In this way, non-Gaussian non-linear distributions can be modelled and difficult situations such as partial occlusions or multiple targets can be solved.

Milestone M1: Explore the feature extraction

In this stage, several features will be explored as characteristics to identify and track human bodies. Motion, colour, gradients or characteristic points will be explored to characterise the targets under different conditions and cameras. The initial feature model of the target in the first frame is assumed as known, but it should be evolved over time to deal with environmental condition changes. Thus, the correspondence with the hypothesis will be easily established.

Milestone M2: Improve the feedback and integration between features and tracking algorithm

The performance of the tracking depends, to a large extent, on the features or likelihood function employed in the process. The quality or discrimination of this measurement is not only important, but also the way in which this information is introduced in the tracking algorithm. Thus, when the measurement can produce confusing estimations, the prediction process and the prior information that the tracking algorithm introduces take on a higher relevance. By improving both mechanisms, and their feedback with the likelihood function, the resulting estimation will be more accurate and robust. We propose the introduction of a priori and rough measurement in the early stages of the tracking in order to guide the prediction of the hypotheses and remove clearly erroneous samples.

Milestone M3: Increase the efficiency and robustness of the tracking algorithm for multi-target

One of the mayor drawbacks of using particle filters and similar algorithms is the non-tractable computational time. This is due to the fact that the number of hypotheses to be evaluated grows exponentially with the dimensionality of the model and therefore with the number of targets. In order to reduce the computational cost, two ways will be explored: dividing the dimensionality of the model by iterative layers or partition techniques [MacCormick and Blake, 2000] and decreasing the number of required hypotheses by improving the prior information [Rui and Chen, 2001; Isard and Blake, 1998b]. Multi-target approaches will be explored during this phase.

1.4.2.3 Phase 3 - Articulated model tracking

Tracking articulated objects [Deutscher et al., 2000; Sminchisescu and Triggs, 2001] requires seeking in high dimensional spaces with complex and generally slow algorithms. Furthermore, this complexity implies some requirements which limit their relevance in specific applications. Since they usually rely on several cameras which need to be accurately calibrated, these models are unpractical in many fields such as video-surveillance, where the potential advantages that they could provide are more than evident. On the contrary, 2D models cannot deal by themselves with the intrinsic ambiguity of projected 3D postures, self occlusions and distortions introduced by camera perspective. Therefore, their usage is usually restricted to well defined motions and camera views, constraints which reduce their value in real applications, where only one camera is available most of the times.

For these reasons, we suggest to explore the world of the 2D models from 2 different approaches: one based on discriminative algorithms, which requires a previous training and therefore only valid for a known scenario, and another one based on generative tracking algorithms.

Our approaches are based on an advance 2D model (could also be called 2.5D) designed to tackle 3D motion patterns such as changes in the pose of the object with respect to the camera. Therefore, they are able to handle changes in rotation and depth. To achieve this without introducing strong motion constraints which would restrict the application of our system, we propose to use some specific knowledge about biomechanics and human gait analysis.

Milestone M1: Active contour and its extension to pose recovery

Discriminative approaches have a set of advantages which have been enumerated in the state of the art, such as real-time or robustness, even under poor quality conditions. In those conditions, a detailed model like articulated patches could not be suitable for a rough measurement. For that reason, active contours [Blake and Isard, 1998] are more adequate in that kind of situations. Because the activity of the subject must be trained before, we propose a scenario where the person is doing a simple but common action like walking, which we assume it is known. While most of the current systems assume the location of the person known, or obtained by an auxiliary and independent tracker, our proposal will try to take advantage of the relationship between the location and speed of the person in the image and the pose that this movement produce in the characteristic space (reduced space). The combination of both search spaces will be done combining both parameters in the same model. We propose a modification of the particle filter called Rao-Blackwellised particle filter [Khan et al., 2004], which enables to apply a different treatment to both subsets of parameter, but extracting a final global estimation. This technique makes possible the use of stochastic/deterministic or discriminative/generative methodologies for each subset.

Milestone M2: Articulated model

For a deeper analysis of the human motion, a more detailed model is required. In those conditions, the application implies better resolution cameras i.e. better measurements for the tracking algorithm. In addition, real time is less important than in surveillance applications. This kind of scenarios allows us the use of generative approaches.

We propose a generative approach in the 2D world. The 2D models limitation will be solved by the inclusion of biomechanical knowledge about the human gait, which simplifies the problem. Our tracking approach will be based on particle filter [Isard and Blake, 1998a]. An iterative procedure applies the filtering in three stages which refines the results by reducing the dimensionality of the problem. This reduction is achieved by the inclusion of two main biomechanic clues: the detection of the pivot point, that is, the location of the foot on which the person leans the weight of the body to do the movement, and the detection of the trajectory that the target is following. Both clues are extracted by means of the joint point extraction algorithm presented by Bouchrika [Bouchrika and Nixon, 2006].

1.5 Overview of the Thesis

This section summarises the content of each one of the chapters that compose this thesis. With the exemption of the two first chapters and the last one, which serve as introduction, summary and conclusion, the rest of chapter follow a similar structure:

1. Introduction
2. Description of the most relevant researches related with the topic
3. The explanation of the proposed method
4. Specific application and utility
5. Evaluation of results
6. Extraction of conclusions and discussion

The Thesis is composed of 7 chapters, which follow a logic order from the simplest tracking methodologies to the most complex ones.

> **Chapter 1** is a general introduction, where the field and the scope of the thesis are introduced. The goals to be fulfilled and the methodology employed to do that, as well as the phases and time plan, are also shown.
>
> **Chapter 2** describes the task of human tracking and puts it into the context of the thesis. An overview of the state of the art is given before presenting the original work in the following chapters.
>
> **Chapter 3** examines simple tracking approaches and evaluates their limitations for a better understanding of the problem and the future proposals but also their utility for a large set of conditions. Moreover, basic concepts about multi-camera and camera calibration are introduced. The applicability of the methodology is demonstrated by integrating the tracking system in a action recognition application, where it plays a crucial role.
>
> **Chapter 4** introduces feature extraction algorithms and their application for tracking systems. An improved integration of both fields is shown in order to obtain more robust and accurate results. Furthermore, techniques for increasing the efficiency of tracking and extending their scope to real-time scenarios are described.
>
> **Chapter 5** examines the influence of the novelties presented in the previous chapter in multi-camera multi-target scenarios. Several concepts to improve their effectiveness in those conditions are detailed. A real application which combines concepts related with the both previous domains is depicted.
>
> **Chapter 6** presents two different approaches to the human pose recovery, both based in 2D models. Several techniques to deal with the inherent problems of 2D applications and the advantages that these systems entail specially in video-surveillance applications are shown.
>
> **Chapter 7** discusses the presented work, extracts the conclusions and suggests the future lines.

As well, **Appendices** explain in detail some collateral aspects of the research which, because of their length, importance or complexity, could break the continuity of the reading. Finally, **Bibliography** contains the references to other works related with this thesis.

2

Human Tracking

"We are limited, not by our abilities, but by our vision."

-Anonymous-

Computer vision is a recent scientific field from the historical point of view. Nevertheless, current techniques can be considered mature in the formal aspect as the bases were established since 1980. In the last decades, the number of researchers in this field, as well as the scientific institutions, has suffered an exponential increase, and it is probably larger than the whole number of scientists a few centuries ago. Therefore, a development that had needed hundreds of years in the past has contributed with thousands of scientific papers and many applications of this knowledge are currently in our daily living.

In spite of that, many contributions must be done to generalise the application field of computer vision in order to go from specific system to robust and universal ones. The huge field that computer vision shapes makes impossible to deal with the problem in a global way and current researches specialise in a subset of knowledge. This thesis tries to address the particular problem of tracking.

2.1 Tracking

Tracking can be defined as the fact of matching objects in consecutive frames using features such as points, lines, blobs or regions. Another definition considers tracking as the equivalent of establishing temporal coherent relations between image features from different frames in accordance with position, velocity, shape, colour or texture, to name a few.

Therefore, tracking can be assumed as a temporal filter which uses past locations of the target and current and past observations to decide the most probable location of the target at current time. In this sense, tracking can be considered as a recursive estimator. In its simplest representation, only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state, in contrast to batch estimation techniques, where the history of observations and/or estimates are required.

However, the filter is not the only component of the tracking paradigm. Two additional blocks complete the framework being as crucial as the chosen filter for the quality of the estimation: the feature extraction and the target model.

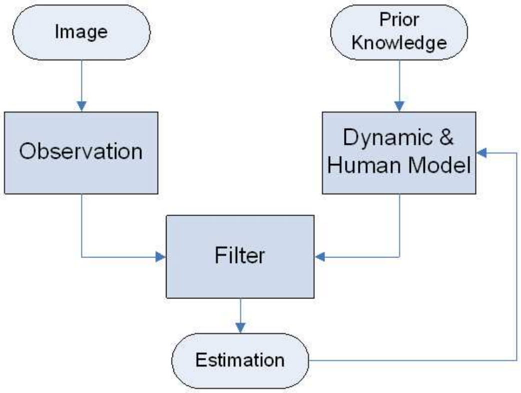


Figure 2.1: Human Tracking Paradigm.

Tracking algorithms require a good dynamic model in combination with the visual observation in order to accomplish a high robust tracking. While the feature extraction is an on-line process whose results depend mainly on the scenario, the target model and its evolution over time (dynamic) are previously learnt on the basis of a priori knowledge and therefore an off-line definition. However, the modelling of the human motion is not an easy task due to the high number of parameters that are involved and the unpredictability of human motion. In addition, the inclusion of the camera perspective effect makes this process even more difficult.

Many different tracking classifications have been made before [Aggarwal and Cai, 1999; Wang et al., 2003] according to different criteria. Human tracking algorithms can be divided into three main categories, which can be subdivided as well in other sub-categories. These categories are related with the target, the camera and the environment.

- With regard to the target, tracking can be divided into tracking of the whole body [Karaulova et al., 2000; Guo et al., 1994; Leung and Yang, 1995; Niyogi and Adelson, 1993; Ju et al., 1996; Rohr, 1994; Wachter and Nagel, 1999; Rehg and Kanade, 1995; Kakadiaris and Metaxas, 1996; Isard and Blake, 1996; Paragios and Deriche, 2000; Peterfreund, 1999; Polana and Nelson, 1994; Cai et al., 1995; Segen and Pingali, 1996; Rossi and Bozzoli, 1994; Gavrila and Davis, 1996; Utsumi et al., 1998] or tracking of body parts (hands, face) [Bernardo et al., 1996; Rehg and Kanade, 1995; Fieguth and Terzopoulos, 1997; Jang and CHoi, 2000], or into bounding-box/silhpuette models [Moeslund et al., 2006]. According to the number of targets, there are single person, multiple person or corwd tracking. Regarding to the dimension of model space, they can be grouped into 2D or 3D tracking.

- Considering the number of views, they can be classifies in monocular [Karaulova et al., 2000; Guo et al., 1994; Leung and Yang, 1995; Niyogi and Adelson, 1993; Ju et al., 1996; Rohr, 1994; Wachter and Nagel, 1999; Rehg and Kanade, 1995; Sidenbladh et al., 2000; Wu et al., 2003], multiple camera [Kakadiaris and Metaxas, 1996; Cai and Aggarwal, 1996; Gavrila and Davis, 1996; Utsumi et al., 1998] or omnidirectional camera [Boult et al., 1998]. Another criterion related to the state of camera divides the algorithms between moving or static camera.

- Finally, attending to the environment, tracking can be classified, for instance, as indoor versus outdoor or day versus night.

A classification based on the model of the target is proposed in our approach. In this way, tracking will be divided into punctual, regional or human pose. This classification enables to understand the tracking process at all the levels. The first level considers the subject as a compact object, small and rigid. The second one models it as a set of connected regions which identifies the subject, each one with its own properties. The last level models the person as a non-rigid object, with morphologic and dynamic properties which are intrinsically human. In connection with the three blocks which composed the tracking paradigm (see Figure 2.1), punctual tracking places emphasis on the filter block, region tracking on the likelihood function and human pose tracking stresses the importance of both human and dynamical models.

The three-level division allows us to create a framework where the subject can be tracked chosen the most adequate strategy on the basis of the necessities, limitation and conditions of the environment. While a low-quality camera with a poor feature extraction will suggest the use of a punctual tracking, a multi-target problem would require a region-based tracking in order to identify and distinguish among the candidates. In the same way, an ambitious application like medical diagnosis forces us to use a highly detailed tracking as human articulated models are.

An exhaustive bibliographic review for each point has been done as one of the goals of this thesis. Not to overwhelm the reader by a huge amount of references, the state of the art about each tracking domain will be detailed in its corresponding chapter. However, papers described there could be classified in other category or even in two categories simultaneously. This is due to the fact that the frontiers between them are quite fuzzy. In this manner, a simple region-based tracker can be defined as the sum of several punctual-based trackers, or a simple human model based tracker is a rigid articulated model with different regions for each part and a constant relationship between them.

2.1.1 Modelling

A model is a descriptor which characterises the target and describes directly or indirectly the appearance of that object. Following this statement, in human tracking, the model introduces a certain notion of the human target which allows us to distinguish people from other objects in the scenario.

The importance of the model is patent. An incorrect model, with an insufficient number of parameters to deal with the complexity that our application requires, causes a failure in the tracking process of the target, in spite of the simplicity of the sequence or the goodness of the measurement. An adequate selection of our model's complexity is therefore a decisive issue to be taken into account.

Human beings are complex articulated bodies whose appearance changes depending on the camera view and occlusions. Complicated 3D models are able to adapt to the changing appearance of humans under pose and viewing angle changes. However, that complexity has disadvantages. The computational cost that these models imply makes them non-suitable for real-time applications. In addition, the high dimensionality of the model space increases the probability of loosing the target due to the visual ambiguities like occlusions or unexpected situations. On the contrary, simple models, such a rectangle that model the bounding box of the person, have a simple implementation and are fast enough for real time, although they exhibit a lack of adaptability to changes in the appearance.

The design of the model is therefore a trade-off between the accuracy of complex models and the robustness and speed of simple ones. The advantages and disadvantages of each option will be decided by the demands of the particular application.

The level of complexity in human modelling varies over a wide range. Using a similar classification of human models to the one exposed in [Siebel, 2003], the tracking methods can be classified into four main categories of increasing complexity:

- **Category 1:** Methods using a single region or blob-based tracking. Although an implicit model is always used by definition, the simplicity of the model makes that some authors call them ”model free”.
- **Category 2:** Methods using multiple region tracking. The relations between regions are fixed and they move together.
- **Category 3:** Methods using an articulated 2D model in the two-dimensional space of the image. The target is not considered a rigid object, but an object with mobile parts. It permits more flexibility than the previous category since the spatial relationships between the regions which compose the model can change. It is suitable to model an accurate representation of the human appearance.
- **Category 4:** Methods using an articulated 3D model in a 3D space, usually, the real world. It enables modelling the real shape of the person removing the perspective effect due to the location of the camera.

The algorithms described in this work are in category 1 (Chapter 3), 2 (Chapters 4, 5) and 3 (Chapter 6).

Two other aspects can also be modelled. The first one is the way in which the model evolves over time, that is, the motion dynamic which characterises and predicts the movements of the model from its past locations and poses. The chosen dynamic model affects to our model, forcing us to include new parameters. For instance, a constant acceleration dynamic model forces the system to include the velocity in the model.

The second one is the evolution of the target appearance or its representation as time goes. Usually, it is assumed that the target appearance varies in a regular manner over time [Nummiaro et al., 2002; Pérez et al., 2002] and therefore, they can be modelled and predicted from the previous frames. In this manner, complex aspects like illumination changes can be tackled.

2.1.2 Tracking Filtering

Filtering is the block responsible of adjusting the model to the given data i.e. the observation. It can be considered as the engine of the tracking algorithm. The typical operation of tracking filters is based on a recursive two-stage process which estimates the state of a dynamic system from a series of potentially incomplete and noisy measurements. The state of the system is defined as a vector composed by the parameters which model the particular target.

The two distinct phases are the Prediction and the Correction (see Figure 2.1.2). The predict phase evolves the estimated state from the previous time step in order to produce a hypothesis about the new state at the current time step. This prediction is made according to the dynamic model. The correct stage introduces the measurement information at the current time step to refine the previous prediction. In this manner, a new estimation, presumably more accurate than the prediction, is obtained for the current time step. This final estimation is used at the next time step to generate the prediction.

Depending on the confidence assigned to the prediction or to the observation, the new estimation, which is a trade-off between both locations, will trust more in one or another, being the final location nearer to the most reliable information. Due to the recursive nature of the filter, an initial estimation should be given, which is not a trivial problem as we will see.

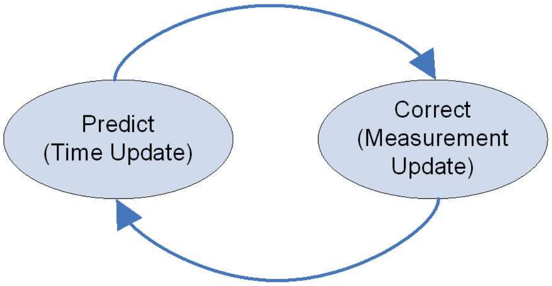


Figure 2.2: Predictor-Corrector cycle.

Regarding to the different filtering approaches, the most important mathematical tools are the Kalman filter [Welch and Bishop, 1995] and the Condensation algorithm [Isard and Blake, 1998a] (also called particle filter). Other available techniques are dynamic Bayesian networks, optical flow, mean shift algorithm or hidden Markov models.

Kalman filter has been used extensively and appears very often in the literature. Kalman filter [Welch and Bishop, 1995] is a state estimation method based on linear dynamical systems discretised in the time domain. They are modelled on a Markov chain built on linear operators perturbed by Gaussian noise. Thus, the state parameters are supposed to be a unimodal probability distribution. If this assumption is true, Kalman approach is the optimum filter. However, this assumption is not fulfilled in most real situations due to the presence of occlusions or cluttered backgrounds. A fact that must be remarked is that most applications use only a linear Kalman filter, which is specially limited, and they work mainly because the environment has been highly constrained (controlled indoor environments) and the motion can be modelled with a simple 2D dynamic model (zenital or lateral view).

The basic Kalman filter is limited to that linear assumption. However, most non-trivial systems are non-linear. The non-linearity can be associated either with the process model or with the observation model or with both. To cope with these situations, a non-linear version, called extended Kalman filter (EKF) [Welch and Bishop, 1995] was proposed. Unlike its linear counterpart, the extended Kalman filter is not an optimal estimator. Moreover, if the initial estimate of the state is wrong, or if the process is modelled incorrectly, the filter may quickly diverge, owing to its linearisation. Another problem with the extended Kalman filter is that the estimated covariance matrix tends to underestimate the true covariance matrix, becoming inconsistent in the statistical sense without the addition of "stabilising noise".

Mean shift [Comaniciu and Meer, 1999] is a non-parametric estimator of density gradient employed in the joint, spatial-range (value) domain of gray level and colour images for discontinuity preserving filtering and image segmentation. The filtering method associates with each pixel in the image the closest local mode in the density distribution of the joint domain. The convergence is guaranteed not requiring the intervention of the user to stop the filtering at the desired image quality. However, this convergence can occur in a local minimum which would not give the correct estimation. Two parameters control the resolution in the spatial and range domains and their values enable the filter to avoid local minima at the expense of a lack of accuracy.

Particle filter [Isard and Blake, 1998a] is a conditional density propagation method which allows us to deal with non-Gaussian distributions. The posterior distribution estimated in the previous frame is sampled with a set of samples, also called particles, which are propagated iteratively to successive frames given the distribution of the target in the sequence.

Unfortunately, it usually requires a large number of particles to ensure a good estimation of the likelihood of the current state.

Particle filters could degrade in performance as the dimensionality of the state space increases and the support of the likelihood decreases. As an alternative to particle filters, a variational approximation to the tracking recursion can be used [Vermaak et al., 2003b]. The variational inference is intractable in itself, and is combined with an efficient importance sampling procedure to obtain the required estimates.

Hidden Markov models (HMM) [Sitbon and Passerieux, 1995] are a statistical model in which the system being modelled is assumed to be a Markov process with unknown parameters where the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. A HMM can be considered as the simplest dynamic Bayesian network.

2.1.3 Observation

The third fundamental component of the tracking methodology is the observation, also called measurement or likelihood function. As it has been shown in the previous section, because of the correction phase, a good and reliable measurement is basic for obtaining a good estimation of the target state. The more reliable the observation is, the better estimation is obtained. Nevertheless, perfect measurement over time is never possible, otherwise the tracking filter would be completely unnecessary.

Many different types of measurements have been used for extracting the most reliable information about the state of the system. Thus, colour, motion, texture or gradients detectors have been applied. These observations represent a good deal between speed and reliability and allow evaluating a set of hypothesis simultaneously. More sophisticated techniques like corner detectors or optical flow can be employed when the situation or the complexity of the task requires it. It is worthy to remember that some of these clues need an initial model to be extracted, such as colour or texture, which furthermore can change over time. This can complicate the modelling of the target and the initialisation of the tracking filter.

Multiple visual clues can be combined to increase the reliability of the observation. To do that, a common practice consists in assuming independence between them. This independence is clear in some cases such as the combination of colour and motion, in which the colour of the target is uncorrelated with the fact that the target is moving (at least during slow motions or high frame rate cameras). However, it should be taken into account that this assumption is not always true and a certain correlation between them exits. This fact can produce that all the visual clues fail at the same time. For instance, a person dressed with cloth with a similar colour to the background makes useless the colour information, but in addition, because of the small difference between background and foreground, produces a slight gradient.

In the same manner that several observations can be combined, information from several sensors can also be mixed to improve the results of our tracking. Similar assumptions related with the independence of the likelihood function of each sensor are used. A good example is the use of muti-camera systems to obtain more robust tracking applications.

2.2 Field of Application

In section 1.2, a list of applications, where tracking is a crucial tool, has been listed to highlight the importance of tracking algorithm and justify the research made. In this section, we are going to show in depth those which have been explored in this thesis.

2.2.1 Video Surveillance

Video surveillance is probably one of the most popular areas for researching and development. The concern about the security that the last international events have produced among the population has implies a capital injection to improve safety in public areas. The huge number of surveillance cameras installed to monitor people means to handle a vast amount of data generated by thousands of cameras working 24 hours a day. Thanks to computer-based image processing systems this task becomes affordable.

Current systems do not have the capacity to automatise the whole surveillance task, and some human supervision is required nowadays. The responsibility of a surveillance system is to warn a human operator when a possible risk or potential dangerous situation is detected. In this way, only one operator can control and monitor a much larger number of camera and therefore, a bigger space with less effort and with a smaller probability of being distracted. The objective in this field consists in increasing the reliability of the existing application in order to reduce the false alarms. This can only be achieved if the system is able to understand the human behaviour, which means to detect, locate and track people as much accuracy and robustness as possible. Only with a good basis, the whole understanding of the scene is possible.

The main difficulties that the surveillance applications imply are:

- Low quality and low resolution images (even black and white cameras). High level of noise.
- Changing lighting conditions.
- Low contrast between people and background due to uncontrolled environment.
- Uncalibrated scenario.
- High variety of different scenarios.
- Partial and total occlusions among people.
- Overcrowded situations.
- Real-time is required.

Not all these conditions appear in all the applications. To reduce the dimension of the problem and simplify it, some assumptions can be done and some constraints can be introduced. For instance, installing cameras for gate access in controlled environments or in narrow corridors to avoid crowds. Another important issue is the necessity of real time applications; otherwise the utility of the system is to save and store the image or data for a posterior review. This fact, in addition to the enormous diversity of environments and conditions, makes that only simple or ad-hoc applications are feasible to date. No commercial product has been shown robust enough to provide accurate results under any scenario. Current researches try to solve these limitations. In this thesis, which is clearly oriented to video-surveillance application, we have tried to cope with this problem obtaining a good agreement between accurate results and robustness against diverse and difficult scenarios.

Some of the most habitual video surveillance scenarios are:

- Traffic monitoring.
- Gate access.
- Security in airports and train stations.

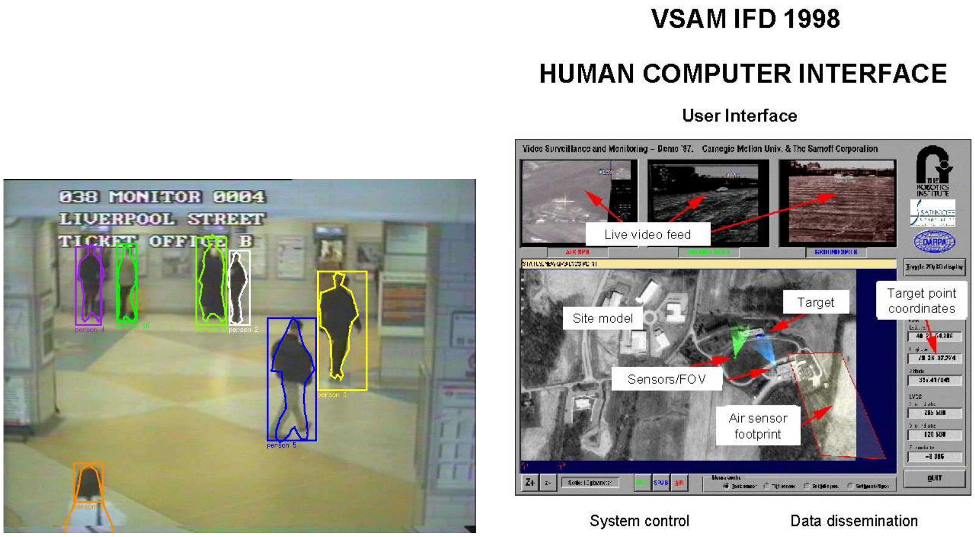
- Theft prevention in shopping mall.
- Crowd control in public events.
- Extraction of statistics.

There are several surveys about video surveillance papers. Regazzoni et al. [Regazzoni et al., 1999] present in their book second generation surveillance systems for performance monitoring tasks. It is centred on different surveillance architecture designs and their customisation for end-users, advances in the processing of imaging sequences, the understanding of the sequence, security systems, sensors, and remote monitoring projects. Foresti et al. [Foresti et al., 2000] cover in their text emerging surveillance requirements, including digital sensors for real-time acquisition of surveillance data, low-level image processing algorithms, and event detection methods. It also discusses problems related to knowledge representation in surveillance systems, wireless and wired multimedia networks, and a new generation of surveillance communication tools for transports and pedestrian monitoring. It is noteworthy the international workshops IEEE Workshops on Performance Evaluation of Tracking and Surveillance PETS (which include benchmark data) and IEEE Workshops on Visual Surveillance, which are focused on this specific topic. In addition to the vast number of papers about the issue, several research projects have tried to obtain a global solution to the surveillance problem.

The Defense Advanced Research Projects Agency (DARPA) Information Systems Office began a three-year program in 1997 to develop Video Surveillance and Monitoring (VSAM) technology [Collins and et al, 2000]. Carnegie Mellon University Robotics Institute and the Sarnoff Corporation were chosen to lead the technical area of the project. The objective of the VSAM project (see Figure 2.3.b) was to develop automated video understanding technology for surveillance applications. As requirement a single human operator should be able to monitor a broad area covered by a network of distributed video sensors. Technical areas include real-time moving object detection and tracking from stationary and moving camera platforms, recognition of generic object classes, object pose estimation with respect to a geospatial site model, active camera control and multi-camera cooperative tracking, human gait analysis, recognition of simple multi-agent activities, real-time data dissemination, data logging and dynamic scene visualisation, vehicle tracking and counting, airborne surveillance, novel sensor design, and geometric methods for graphical view transfer.

Another relevant project is European Union's research project Annotated Digital Video for Surveillance and Optimised Retrieval (ADVISOR IST-1999-11287) (see Figure 2.3.a). The time span was 3 years in length and it included three academic and three industrial partners and the goal was to build an integrated real time surveillance system for use in underground stations. The requirements of the project were the tracking of multiple people in multi camera scenarios, estimation of crowd density, analysis of people behaviour, alarm generation of dangerous situations, storage of images and event annotation in a database and the complete integration with the underground station camera system.

Project CAVIAR: Context Aware Vision using Image-based Active Recognition (IST 2001 37540) addressed two main applications: city centre surveillance, where unusual events, such as patterns of running people, converging people, or stationary people, are detected; and customer behaviour analysis to enable evaluation of shop layouts, changing displays and the effect of promotional materials. For these purposes, the main challenge to be tackled were: foveated and other feature extraction and grouping, integrating feature, object and top-down priming for spatial and temporal attention, representing and recognising objects, contexts and situations, learning instances of the representations from visual evidence and reactive and top-down control of the recognition process.



a) b)

Figure 2.3: Examples of surveillance projects: a) Advisor project. b) VSAM project (Image source: http://www-sop.inria.fr/orion/ADVISOR/ and http://www.cs.cmu.edu/∼vsam/)

The CASSANDRA project (Aggression Detection by Fusion Video and Audio) started in 2005 and with a length of 3 years is being developing in the University of Amsterdam. It pursues human activity recognition in dynamic environments, in particular, automatic aggression detection. Because events associated with the buildup or enactment of aggression are difficult to detect by a single sensor modality (e.g. shouting versus hitting-someone), CASSANDRA combines audio- and video-sensing.

A clear example of application will be given in section 3.5.1, where a robust system for detecting left luggage in train stations is presented. In addition, tracking people scenarios indoor (Appendix D) and outdoor (section 4.6.3) has been tackled, and an approach for maritime surveillance was proposed (section 4.6.1).

2.2.2 Sports Analysis

Competition occurs naturally between living organisms which coexist in the same environment. Animals compete over water supplies, food, and mates. Human beings, in addition, compete for attention, wealth, prestige, and fame. Thanks to the competition the life survives and the civilization steps forward. In the current society, in the same way that in former times during the peace periods, sports are a way out, where the sportsmen are heroes due to the extensive public attention. This produces a continuous flow of money, part of it is invested to improve the performance of the elite players, and therefore, their benefits. High-performance centres and specific applications on sports have been created to respond to this necessity, incorporating a conceptual ideal of a perfect performance, which incorporates measurable criteria and standards which are translated into numerical ratings and scores.

Sports are generally broken down into three categories: individual sports, such as athletics, dual sports, such as tennis, or team sports competition, such as football. Computer vision techniques applied to improve the performance in individual or dual sports have been used extensively. These techniques are based on biomechanics analysis and human pose estimation,

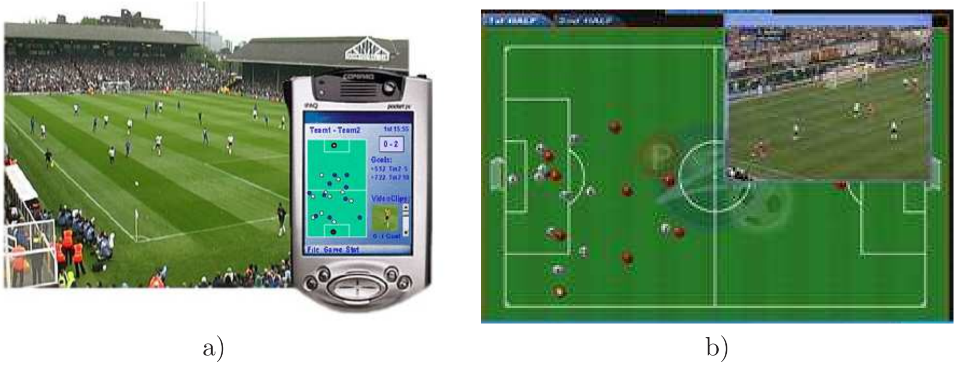


Figure 2.4: Examples of sports analysis projects: a) INMOVE project. b) SIGA project. (Image source: http://inmove.erve.vtt.fi and http://paginas.fe.up.pt/∼lpreis/Projects/SIGA.htm)

and they will be explained in detail in the next section.

Team sports are much more difficult to analyse automatically, due to the high degree of interaction that these activities imply. The importance of tracking in this category is crucial since it allows us to locate the player. The complexity of the algorithms required in cooperative sports grows exponentially, being necessary the use of multiple target tracking techniques and multiple camera systems. In this line two project should be remarked.

INMOVE project (IST 2001-37422) was a 2 year project started in 2001. This project falls into two application areas: Sports Viewing and Intelligent Monitoring. Regarding the first area, INMOVE chose football for the sports viewing application to be implemented into a concrete pilot system. This choice was based on the enormous popularity of this sport throughout Europe. Implementation of the pilot system required several cameras installed around the field. This was crucial for the technically most difficult task of the application, i.e. for detecting and tracking the players and the ball. Information about the location of players and ball is sent to a remote spectator equipped with a PDA terminal (see Figure 2.4.a). He/she has several means to follow the match in real-time or afterwards. In real-time it is possible to watch the whole football field with animated players and the ball, or a video view from any of the cameras. Off-line the user can request replays of his/her choice on specific situations, e.g. shots on goal, either as animated or video-based. In addition the user has accessed to various relevant information, like match statistics and team or player data. Applications and toolkit software were developed, with an emphasis on usability issues, relevant signal processing features, intelligent image/video analysis methods, video transmission technology, virtual and augmented reality tools, and wireless capacities and mobility.

Soccer Intelligent Game Analysis (SIGA) Project aims at building an intelligent soccer game analysis and simulation system (see Figure 2.4.b). An autonomous vision system based on six cameras installed on a real soccer stadium, with 3D player's detection and tracking capabilities, is the basis for the system information gathering, and artificial intelligence techniques in order to rectify and correct information coming from the image processing and analysis system. The agent is also capable of calculating complex individual and team performance statistics. Results are stored in an electronic format appropriate for its analysis and real-time visualisation using a graphical system.

The techniques and methodologies developed in this thesis have been employed in two projects ASTRO'05: Automatic System for Tactical Review and Optimization (DGA 0438-6 2005) and ASTRO'06. Both projects are the result of a collaboration agreement between the Government of Aragon, the University of Zaragoza and Real Zaragoza S.A.D, for the investigation of particular techniques in the sports field. The goal is to measure and store tactical and fitness data for team sports, watching the movements of all players at the same time and delivering an accurate analysis of the performance of all player during the whole match. The system must work with a minimum human interaction, and if it is possible, in real time. A multi-camera recording system installed in the stadium provides the images which are processing by a distributed cluster. As result, a 2d representation is obtained as well as the statistics of each player.

2.2.3 Human Motion Analysis

Human motion analysis consists in the observation of the motion of human beings. The goal is the pose estimations of a person during a video sequence and its posterior analysis to draw conclusions about the dynamic of the human being. To simplify the task, tracking of each body part in which a human can be segmented is applied. Human gait analysis is used for diverse applications such as:

- Medical diagnosis.
- Biomedical research.
- Sport performance.
- Rehabilitation.
- Ergonomics.
- Gait biometry.
- Computer graphics and animation.

To solve the problem, different approaches have been tested, existing even commercial solutions. Professional solutions are mainly based on markers, that is, small visual references attached to a suit which are captured by a multi-camera system and converted to 2D or 3D coordinates. VICON (see Figure 2.5)is one of the most famous motion capture systems which employs a system composed of several infrared cameras to detect reflective markers. In addition, visual markers are not the only possibility and suits based on gyroscopes or accelerometers are also available.

However, these techniques are intrusive, requiring the collaboration of the target and limiting the functionality only for voluntary application. Thus, current research is focused on non-marker pose estimation systems. Given the extraordinary complexity of the task, restrictions about the type of motion, the scenario or the environment are introduced. In this way, close environments or models of a simple and repetitive action are applied.

Project LEAR (Learning and Recognition in Vision) is an interdisciplinary research team based at INRIA (the French national institute for research in computer science and control). LEAR research focuses on machine learning and statistical modelling based approaches to visual object recognition, scene interpretation, and image and video indexing. The research in this area uses machine learning techniques and robust visual shape descriptors to characterise humans and their movements with little or no manual modelling. Particular focuses include robust human detection in images and videos, and reconstructing 3D human movement from monocular images.

The Human Pose Estimation project was supported by the National Science Foundation. The goal of this effort was to develop algorithms for articulated structure and motion estimation,

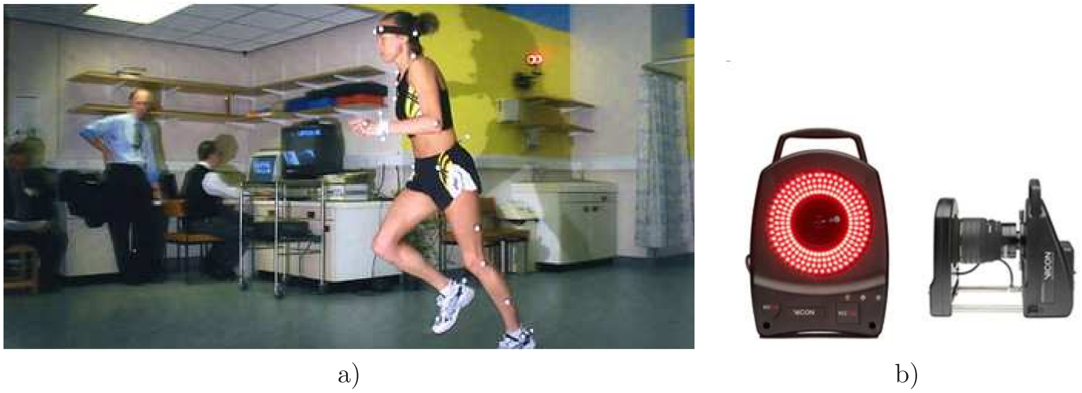


Figure 2.5: VICON Motion capture system. a) Example of application. b) Detail of infrared cameras used in the system. (Image source: http://www.vicon.com)

from one or more video streams. In this project, 3D articulated structures and motion estimation algorithms were developed so they could automatically initialise themselves, estimate multiple plausible interpretations along with their likelihood, and provide reliable performance over extended sequences. In order to achieve these objectives, concepts from machine learning, graphical models, multiple view geometry, and structure from motion were employed. The research effort was focused on two main areas: (1) 3D articulated pose estimation given video obtained from uncalibrated cameras, (2) statistical learning models that capture the dynamics of articulated motion. This project also investigated detection and segmentation methods for use in localising the articulated structure in the image.

Finally, the Research and Development in Ergonomics group, belonging to the Aragon Institute of Ingeenering Research, tries to develop analysis tools in order to find out if the person poses are ergonomic or they should be corrected.

In Chapter 6, two approaches to estimate the human pose in video sequences have been addressed. Both methods are based on 2D techniques and keep an agreement between accuracy and robustness against noise and low-quality signals.

2.2.4 Activity Recognition

Activity recognition addresses the problem of understanding vision from both computational and cognitive points of view. The intrinsic difficulty that it implies is due to the fact that a complete understanding of the scenario, the person and the environment is required. This complete understanding is only possible if the big gap between the human and the computer cognitive systems is reduced by introducing neural and psychological approaches like neural networks and fuzzy systems. Obviously, it also requires robust and mature tracking, detection and/or pose estimation algorithms to work properly. A failure in any of these stages would propagate the error through the scheme, making impossible the understanding of the action.

The most complex existing systems employ pose estimation techniques to obtain detailed information about people, and in this way, differentiate between a set of predefined activities. These systems are strongly related with the field seen in Section 2.2.3. The extraction of the relative motion between the limbs as well as the continuity of this motion is the key to identify the action, and it is in this area where tracking play a crucial role. Because of human pose problem is not solved yet, this systems are limited to close and controlled environments.

A different kind of action recognition systems are based on coarse clues, like global motion

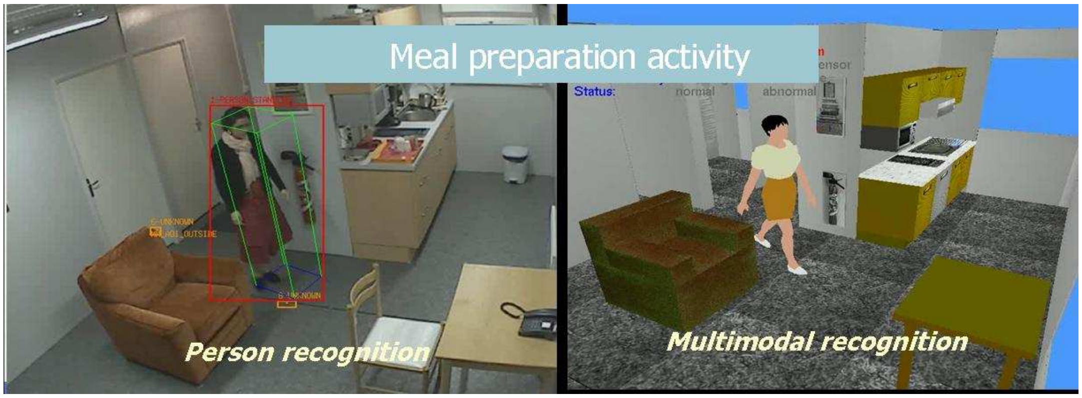


Figure 2.6: Activity Recognition example. (Image source: GER'HOME project http://www-sop.inria.fr/orion/personnel/Francois.Bremond/topicsText/gerhomeProject.html)

or trajectories (see Figure 2.6). They are not useful to distinguish between subtle actions, but on the other hand, the requirements that theses systems need regarding to detection and tracking stages are much more realistic. Therefore, this methodology is perfect for non-ambitious and specific applications, being much more robust and adaptable to the scenario than the previous techniques. In this line, a robust system capable of detecting a dangerous activity like abandoning luggage is presented (see Section 3.5.1).

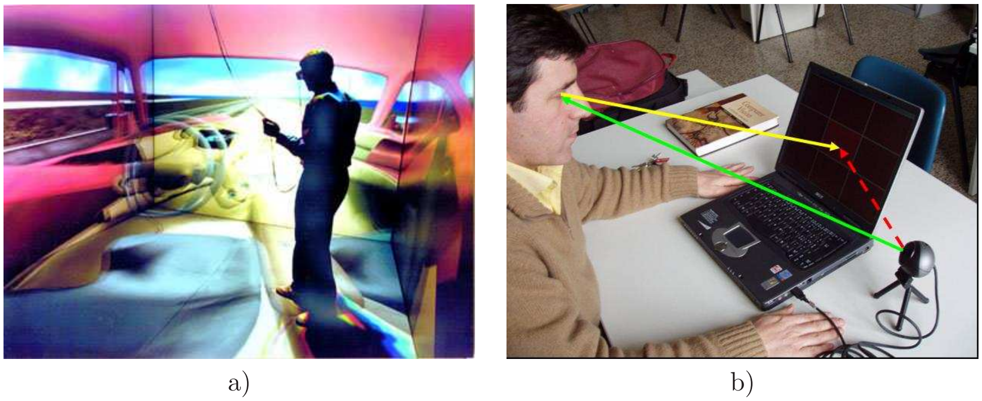
Action recognition is highly related with advanced visual-surveillance application, in which the action should be identified in order to reduce the number of false alarms. Therefore, most of the visual surveillance projects and researches can be included in this field. However, action recognition is a larger area, where other disciplines are enclosed. This field is, for instance, one of the cornerstones of the ambient intelligence systems.

The project MEDUSA: Multi Environment Deployable Universal Software Application is concerned with the identification of situations associated with gun related threats, based on behavioural interpretation of CCTV data and through combining psychological and image processing approaches. MEDUSA is a collaborative project funded by the EPSRC, and in which the Kingston University is involved, for three years which began in July 2006. The project aims to develop a new machine learning system for the detection of individuals carrying guns through the use of CCTV surveillance networks. The system will combine both human and machine-based factors in achieving this.

The European Project MonAmi: Mainstreaming on Ambient Intelligence is focused on ambient intelligence, specially on approaches to support people as they age in maintaining independence in daily living at home, at work and in the community. This application requires a robust action recognition system capable of detecting dangerous situation such as falls. MonAmi runs for four year and it started in 2006. The University of Zaragoza plays an important role being the responsible of the action recognition module.

2.2.5 Human Interface

Another interesting application field is the development of advanced user interfaces, in which human tracking is used for controlling a computational system. Although human communication is based on speech, visual clues are fundamental in order to understand the message as well as perceiving the feelings and context. To a large extent, many clues for a complete understanding can be obtained watching the pose, gestures, facial expressions,



a) b)

Figure 2.7: Examples of human-machine interfaces: a) CAVE virtual reallity system. b) Eye-mouse using webcam. (Image source: http://um3d.dc.umich.edu/hardware/CAVE/ and http://vcg.isti.cnr.it/∼corsini/research.html)

etc. Recent advances on computer vision are leading to the automatic gesture detection and recognition, introducing radical changes in the way humans and computers interact. A computer linked to a video camera is able to detect the presence of users, track faces, arms and hands in real time, and analyse expressions and gestures. The implications for interface design are immense. Special attention has received the automatic gesture recognition and sign language understanding for deaf and dumb people, because of social reasons, or the machine control in noisy environments, like factories or airports where oral communication is not possible.

Most researches are focused on hand and face detection and tracking as basic clues to locate the person and react to his/her movements (Figure 2.7). Currently, human-machine interfaces are not robust and accurate enough to replace conventional interfaces for the normal user, but systems for disable people or specific applications for specialised tasks or environments have been developed with successful results.

Several overviews on gesture recognition and interfaces are available. The book *"Computer Vision for Human-Machine Interaction"* [Cipolla and Pentland, 1998] collects several ideas and algorithms from different scientists, offering a glimpse of the radical changes that are around the corner and which will change the way we all interact with computers in the near future. The Gesture Recognition Home Page [Cohen, 2008] summarises the main groups and projects related with the gesture recognitions as well as the future events and scientific meetings. Content is classified into several categories: human hand gesture research, human body motion gesture research, pen- and mouse-based research and sign language recognition. Other webpages [Kohler, 2008; Howell and McKenna, 2008] also collect a set of links to papers, groups, projects and meetings on this topic.

The ENACTIVE Network is a multidisciplinary research community with the aim of structuring the research on a new generation of human-computer interfaces called Enactive Interfaces i.e. interfaces based on the active use of the hand for apprehension tasks. This type of knowledge transmission can be considered the most direct, in the sense that it is natural and intuitive, since it is based on the experience and on the perceptual responses to motor acts. Enactive Interfaces are based on the capability of recognising complex gestures. Intelligent interfaces recognise the gesture of the user at the beginning of the action and are able to interpret the gestures and to adapt to them in order to improve the user's performance.

Our little contribution to this vast field is a face tracker based on skin colour (Secttion 4.6.2). The tracking algorithm by itself is only a part of the system, but its role is basic. The algorithm can be employed not only as first step to apply a more complex algorithm like eigenfaces but also as a simple interface (mouse) using the movement of the head.

2.3 Future Lines of Human Tracking

In this chapter, we have been discussing about the current state of people tracking and the stages involved. An introductory idea about different applications which can profit from this field has been exposed. In the current section, we are going to detail the drawbacks that are unsolved yet and we will predict the future lines to address by the scientific community.

Current systems have achieved to solve automatic vision problems in all sort of applications, but only for specific scenarios under a set of fixed constraints. If those constraints are not fulfilled, both accuracy and robustness fall dramatically. Then, no universal solutions have been proposed, only fragile applications which could not work under different assumptions.

Under this framework, in order to achieve these ambitious goals, we should intend to reduce the gap that exists today between biological vision (which is by large not yet understood) and computer vision (which is biologically inspired and whose flexibility, robustness, and autonomy remain to be demonstrated). This implies the whole understanding of the content of everyday images and videos as one of the major outstanding challenges of computer vision.

This global vision forces to deal with the different fields which compose the computer vision, in general, and the human tracking, in particular, at the same time. Only combining the different disciplines to take advantage of the strong points reducing the influence of the weak ones, the emulation of the human brain will be a reality. A clear example of the influence and the relationship between the different areas can be seen reviewing the fields of applications: video surveillance applications are related with action recognition which is related with human motion analysis, and this last one is also related with sports analysis applications.

Consequently, multi-disciplinary systems combining different levels of understanding and introducing a feedback procedure between the algorithms which will compose this complex system are postulated as the future way to tackle the problem. Only in this manner, an approach to the biological system will be possible from the artificial cognitive systems.

In this line, our humble contribution in this direction consists in the integration of levels of understanding to tackle the tracking problem in a global way.

3

Punctual Domain Tracking

"Distance lends enchantment to the view."

-Mark Twain-

Punctual tracking is the most elemental tracking domain. Although this simplicity implies a set of drawbacks, such as weakness against multiple target tracking and occlusions, it also permits to reduce the complexity of the human body tracking to adjust the necessities to limited scenarios. This kind of systems has been used extensively in surveillance due to its robustness for simple tasks, light computational cost and good performance with rough measurements, which make it especially suitable in this field.

The subject is considered as a compact object, small and rigid. Usually, a simple rectangle which surrounds the target, called bounding box, is enough to model the person.

No restriction about the tracking filter to be used exists, being all of them adequate. However, it has no sense to employ complex filters or evaluate multiple locations to track a single object because the simplicity of the measurement discards automatically most of them or makes all of them redundant. For instance, it is not profitable to launch several particles around an area using Particle filter if all of them are associated to exactly the same measurement and therefore give the same estimation. This is due to the fact that observation can not be based on a prior model of the subject, which is usually unknown, and whose size or resolution makes difficult to apply complicated detection algorithms. The problem which appears in this type of observation is normally not to detect the target to provide the filter with enough information or associate the correct observation to its corresponding tracker.

In this chapter we will analyse the filtering and the observation process, such as was described in Figure 2.1. Modelling will not be described explicitly due to its simplicity. Unscented Kalman filter will be detailed as punctual filter as a good agreement between robustness and simplicity. The observation will be provided by a motion detector since non prior information about the target is required. Due to its simplicity, a set of mathematical tools will be described to overcome the limitations that this observation implies by means of multi-sensor conjugation. Finally, a robust and accurate application on video surveillance will be presented to prove the utility of punctual tracking in spite of its limitations.

3.1 State of the Art

Punctual domain assumes that the detection process is applied before the tracking process. It receives all the possible candidates as a list of observation vectors. This category is especially relevant when the target is so far from the camera than can not be modelled in a more complex way or the measurement has been simplified to a blob, which can be characterised as a point (the centroid, the point in contact with the floor, etc...). Kalman filter is the clearest example of this tracking domain.

These systems work well if the number of targets is small or the occlusions are short and not so often. Otherwise, the probability of loosing the target will be high for two main causes

- The absence of measurement during a long period of time will produce that the prediction tends to diverge from the reality.
- A consecutive number of bad measurements that move further away the tracker and make impossible that the tracking algorithm can recover the real location.

Due to its simplicity, usually only one person in a constrained environment is followed. Cai et al [Cai et al., 1995] and Okawa and Hanantani [Okawa and Hanatani, 1992] propose a system to track a single person in a monocular indoor environment. In [Cai et al., 1995] the person is extracted using difference between consecutive images and the possible motion of the camera system is estimated detecting the edges in the background. In [Okawa and Hanatani, 1992] a simple background subtraction is used. If we want to follow more than one objets, external interaction models should be included to achieve a successful tracking, as we will see in Chapter 5.

Multiple camera systems turn out more interesting. The combination of several sensors, even using really simple features, has been shown reliable in many different scenarios and it is helpful for reducing the ambiguity or handling occlusions. Thus, Cai and Aggarwal [Cai and Aggarwal, 1996; Cai and Aggarwal, 1999] use motion detection to extract the blob of the people in the scene. After that, once the target has been identified as a person using PCA, a Bayesian classifier makes the matching of the target between consecutive frames. The multi-camera extension is introduced translating all the features to the same coordinate system and being modelled with a multivariate Gaussian distribution. They also discuss a switching mechanism between neighbouring cameras.

As requirement, a calibration process [Tsai, 1987] must be done previously in order to obtain a shared reference for all the cameras. Establishing feature correspondences from multiple views is more challenging than from a single view because they are recorded in different spatial coordinates. Therefore, they must be adjusted to the same reference before performing the matching process. In the work by Sato et al. [Sato et al., 1998] a distributed vision agent model in which each agent has a monocular camera is proposed, an image processor and a communication link to other agents. A CAD solid model of a building is used as reference, and it makes possible track multiple objects harmoniously.

Multi-camera systems enable tracking several people using simple algorithms like Kalman filter or probabilistic matching. Utsumi [Utsumi et al., 1998] track multiple human motions with a multiple-view based system. Kalman filter follows the human positions and an automatic viewpoint selection mechanism reduces the impact of self-occlusion and mutual occlusion among people.

The crucial issue in a tracking system based on multiple cameras consists in handling the selection and data fusion between cameras. The system must be able to decide the cameras to be used at each time instant. In addition, the capability of selecting the best view to represent the data [Utsumi et al., 1998; Cai and Aggarwal, 1999] in each time step is a useful tool to simplify the task of the human supervisor. In this line, Thirde et al. [Thirde et al., 2006] present a multi-camera tracking surveillance system as part of the AVITRACK project. The aim of the project is to automatically recognise activities around a parked aircraft in an airport apron area to improve the efficiency, safety and security of the servicing operation. The multi-camera tracking module takes as input per-camera tracking and recognition results and fuses these into object estimates using a common spatio-temporal co-ordinate frame.

The paper [Meyer et al., 1998] remarks the problems of monocular tracking and its inability to measure the 3D-size and 3D-position of objects reliably, as object size and velocity are estimated within the 2D-image plane. A second camera is used to include 3D-information about the scene. An efficient 3D-scene model combines the evaluation of the measurement data from two cameras with an overlapping field of view. The logical conclusion of this work is that the combined evaluation of two cameras reduces the false detection rate in comparison with a pure monocular evaluation.

Finally, Black and Ellis [Black and Ellis, 2002] developed a multi-camera image tracking system in the context of a surveillance and monitoring task, principally targeted at tracking people through indoor and outdoor environments. A simple motion detector based on background subtraction is applied for each camera, and correspondences between viewpoints are established by a epipole-line analysis. A least square estimation extracts measurements on each corresponding object in the 3D world, and their error covariance is computed using a sensory uncertainty field. This information is used for interpreting the behaviour of the target and learning the relationships between each camera and the scenario.

The multiple-camera tracking methodology can be extended to combine other kind of sensors, like laser, ultra-sound and so on. As an example, in the work by Rossi and Bozzoli [Rossi and Bozzoli, 1994], a method to track and count people with laser range finders is introduced. Multi target model and Kalman filter based estimation are employed.

3.2 Filtering

Kalman filters are one of the most used approaches [Welch and Bishop, 1995]. In brief, Kalman filters are based on linear dynamical systems discretised in the time domain. At each discrete time increment, a linear operator is applied to the state of the system x_k to generate the new state x_{k+1}, with some noise v_k mixed in, and optionally some information from the controls on the system u_k if they are known (see Figure 3.1).

$$\mathbf{x}_k = F_k \cdot \mathbf{x}_{k-1} + B_k \cdot \mathbf{u}_{k-1} + \mathbf{v}_{k-1} \tag{3.1}$$

where F_k is the state transition model which is applied to the previous state $\mathbf{x}_{k-1}$, B_k is the control-input model which is applied to the control vector $\mathbf{u_k}$ and $\mathbf{v}_k$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance R_k^v.

$$\mathbf{v}_k \sim N(0, R_k^v) \tag{3.2}$$

At time k an observation (or measurement) $\mathbf{z}_k$ of the true state $\mathbf{x_k}$ is made according to

$$\mathbf{z}_k = H_k \cdot \mathbf{x}_k + \mathbf{n}_k \tag{3.3}$$

where H_k is the observation model which maps the true state space into the observed space and n_k is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_k^n.

$$\mathbf{n}_k \sim N(0, R_k^n) \tag{3.4}$$

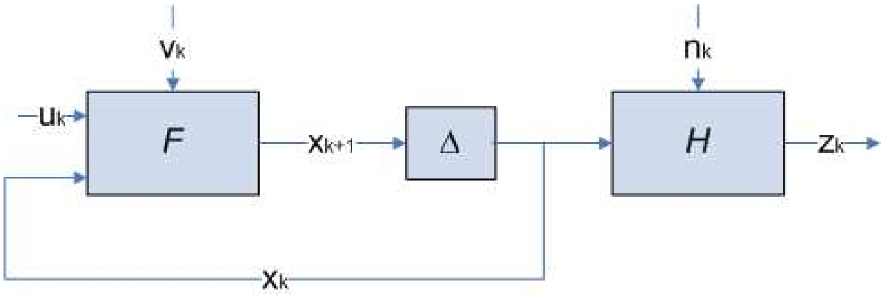


Figure 3.1: Dynamic system model underlying the Kalman Filter

The initial state, and the noise vectors at each step $\{\mathbf{x}_0, \mathbf{v}_1, ..., \mathbf{v}_k, \mathbf{n}_1 ... \mathbf{n}_k\}$ are all assumed to be mutually independent. F_k and H_k are assumed as known.

Although most real dynamical systems do not fit this model, the Kalman filter can approximate it given good enough results because it is designed to operate in presence of noise. Extended Kalman Filter (EKF) tries to solve part of this drawback removing the linear assumption that the Kalman filter uses. However, when the state transition and observation models, that is, the prediction and updating functions are highly non-linear, EKF gives poor performance. This is due to the fact that EKF uses first order terms of the Taylor series expansion of the nonlinear functions, that is, only calculates the posterior mean and covariance accurately to the first order. To solve that problem Unscented Kalman filter was proposed, which calculates the mean and covariance to the second order. A comparison between the different versions of Kalman filter can be seen in Figure 3.2. The left plot shows the true mean and covariance propagation over the sampled real distribution. The center plots show the results using the EKF which is not able to obtain a good estimation due to the highly non-linear function. Finally, the right plots show the superior UKF performance, where the distribution is correctly approximated thanks to the transformed sigma points.

The Unscented Kalman filter (UKF) [Julier and U., 1997][Wan and v. d. M., 2001] is a very popular tracking algorithm which provides a way of processing non-linear but Gaussian models. It uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the non-linear functions and the covariance of the estimate is then recovered. The result is a filter which more accurately captures the true mean and covariance. (This can be verified using Monte Carlo sampling or through a Taylor series expansion of the posterior statistics.) In addition, this technique removes the requirement to analytically calculate Jacobians, which for complex functions can be a difficult task in itself.

In this chapter, we propose a modified UKF, called multi-camera UKF (MCUKF) to extend its application to multi-sensor scenes, thus improving the global result. The combination of several independent sensors increases the precision and robustness of our tracking system, since it makes possible to solve difficult situations, such as occlusions or noise.

3.2.1 Multi-Camera Unscented Kalman Filter

In this section we explain a modification of the UKF which combines several measurements, provided by different cameras, for each tracked object. This algorithm has been developed by our research team at the university of Zaragoza. Due to the use of several sensors as measurement sources, we call the algorithm Multi-Camera Unscented Kalman Filter (MCUKF) [Gómez et al., 2006; Martínez et al., 2006].

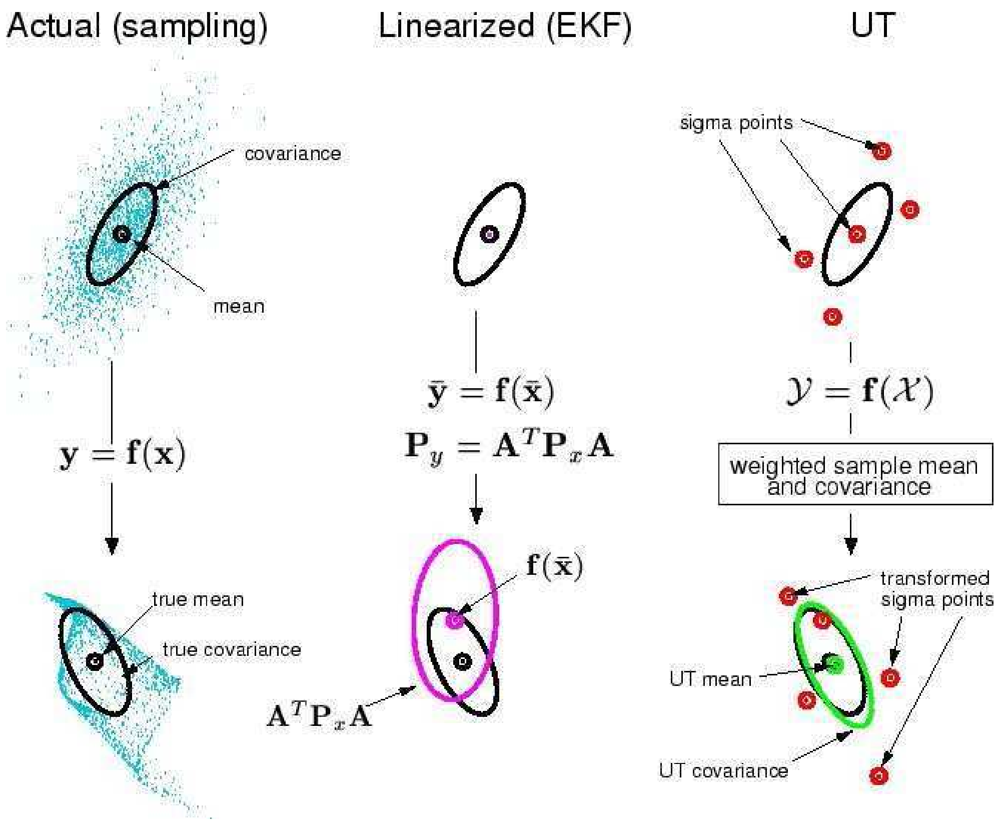

Actual (sampling)
Linearized (EKF)
UT
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T\mathbf{P}_x\mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T\mathbf{P}_x\mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UT mean
UT covariance

Figure 3.2: Graphical comparison between the unscented and the extended kalman filter. x is the random variable which is propagated through a nonlinear function f, with mean $\bar{x}$ and covariance P_x. $\mathcal{X}$ are the sigma vectors. (Image source: http://cslu.cse.ogi.edu/nsel/ukf/img84.gif)

The filter can be divided into three stages: state prediction, measurement prediction, and estimation. This process can be shown in the following scheme (Figure 3.3). An external matching process must be used in order to make correspondences between trackers and measurements.

3.2.1.1 State prediction

In the prediction stage, the tracker is initialised with the last estimation done in the previous time step. Hence, knowing the previous state $\hat{\mathbf{x}}_{k-1}$, with $e \times 1$ components, and its covariance $\hat{\mathbf{P}}_{k-1}$, with $e \times e$ components, both the extended covariance $\hat{\mathbf{P}}^a_{k-1}$ and state $\hat{\mathbf{x}}^a_{k-1}$ can be obtained concatenating the previous parameter and the state noise $\mathbf{v}_k$. This is a simplification of the UKF, in which state and measurement noises are used. The measurement noise will be used in the measurement prediction stage.

$$\hat{\mathbf{x}}^a_{k-1} = \left[\hat{\mathbf{x}}^\top_{k-1} \quad E\{\mathbf{v}_k\}\right]^\top \qquad \text{with} \quad E\{\mathbf{v}_k\} = [0\ 0\ ...\ 0]^\top \tag{3.5}$$

$$\hat{\mathbf{P}}^a_{k-1} = \begin{bmatrix} \hat{\mathbf{P}}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^v \end{bmatrix} \qquad \text{where} \quad \mathbf{R}^v \quad \text{is the state noise matrix.} \tag{3.6}$$

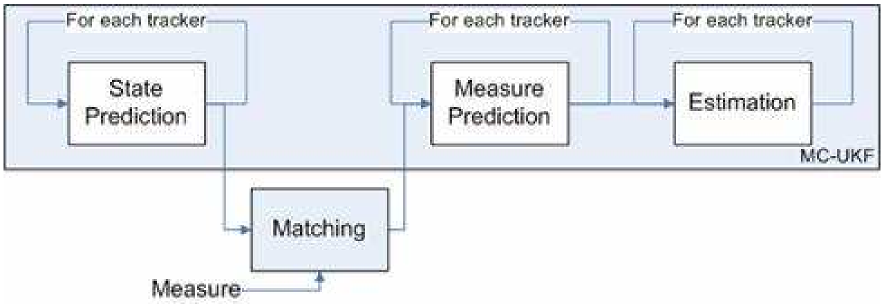


Figure 3.3: MCUKF algorithm.

The number of sigma points is $2n+1$, where n is the length of the extended state.

Following the classical equation of the Unscented Transform, the first sigma point corresponds with the previous frame estimation, the next n_{th} sigma points are the previous estimation plus each column of the previous estimation matrix, and the last n_{th} points are the previous estimation minus the same columns.

$$\mathcal{X}^a_{k-1} = \left[\hat{\mathbf{x}}^a_{k-1} \quad \hat{\mathbf{x}}^a_{k-1} + \sqrt{(n+\lambda)\hat{\mathbf{P}}^a_{k-1}} \quad \hat{\mathbf{x}}^a_{k-1} - \sqrt{(n+\lambda)\hat{\mathbf{P}}^a_{k-1}} \right] \tag{3.7}$$

The components of these sigma points can be divided into two groups: derived from the state $\mathcal{X}^x_{k-1}$ and from the state noise $\mathcal{X}^v_{k-1}$.

The weights assigned to each sigma point are calculated in the same way as in the unscented transformation. Therefore, the 0_{th} weight will be different to obtain the mean weight $W_i^{(m)}$ or the covariance weight $W_i^{(c)}$.

$$\begin{array}{ll} W_0^{(m)} = \lambda/(n+\lambda) & \\ W_0^{(c)} = \lambda/(n+\lambda) + (1+\alpha^2+\beta) & \\ W_i^{(m)} = W_i^{(c)} = 1/\left[2(n+\lambda)\right] & i = 1,2,\ldots n \end{array} \tag{3.8}$$

where $\lambda = \alpha^2 \cdot (n+k) - n$ is a scale parameter. Constant α involves the spread of sigma point around the mean $\bar{x}$ which has a small positive value (usually $1 > \alpha > 10^{-4}$). Constant k is a secondary scaling parameter, usually with values between 0 and $3-n$. Finally, β is used to incorporate a previous knowledge of the distribution of x.

In order to predict the sigma points in the k_{th} instant, knowing the previous points, the transition matrix $\mathbf{F}$ is firstly required. Using a constant velocity model, $\mathbf{F}$ can be calculated as:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F} \cdot \hat{\mathbf{x}}_{k-1} \tag{3.9}$$

The sigma point in the next instant of time:

$$\mathcal{X}^x_{k|k-1} = \mathbf{F} \cdot \mathcal{X}^x_{k-1} + \mathcal{X}^v_{k-1} \tag{3.10}$$

With these points and their weights, the predicted mean and covariance are given by

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{X}^x_{i,k|k-1}$$

$$\hat{\mathbf{P}}_{k|k-1} = \sum_{i=0}^{2n} W_i^{(c)} \left[\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right] \left[\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]^\top \tag{3.11}$$

A graphic representation of this process is depicted in Figure 3.4.

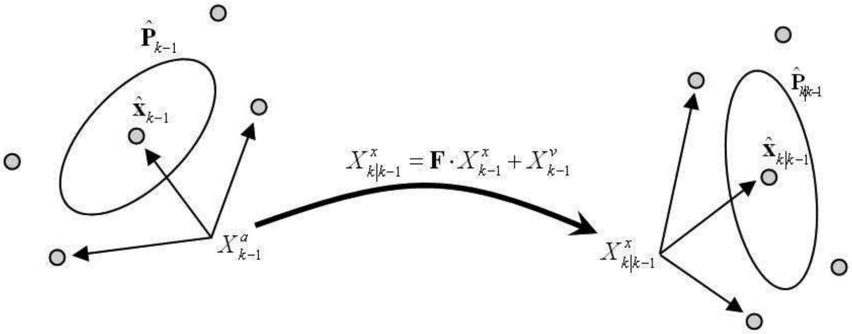

$\hat{\mathbf{P}}_{k-1}$
$\hat{\mathbf{x}}_{k-1}$
X^a_{k-1}
$X^x_{k|k-1} = \mathbf{F} \cdot X^x_{k-1} + X^v_{k-1}$
$X^x_{k|k-1}$
$\hat{\mathbf{P}}_{k|k-1}$
$\hat{\mathbf{x}}_{k|k-1}$

Figure 3.4: State prediction of mean and sigma points

3.2.1.2 Measurement prediction stage

The second contribution to original UKF consists in obtaining the measurement prediction taking into account measurements and measurement noises of each camera. In the measurement prediction stage, the first step consists of calculating the state predictions and the measurement-tracker matching. Next, using the predictions and the measurement noise, both the extended state $\hat{x'}^a_k$ and covariance $\hat{P'}^a_k$ can be developed. The concatenated measurement noise matrix R^n is built from measurement noise matrices of each camera R^n_i with $r \times r$ components, being r the dimensionality of the measurement and $i = 1, 2, ..., S$ with S the number of cameras.

$$\hat{\mathbf{x}'}^a_k = \left[\hat{\mathbf{x}}_{k|k-1}\ 0\ 0\ldots\right]^\top \quad \hat{\mathbf{P}'}^a_k = \begin{bmatrix} \hat{\mathbf{P}}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^n \end{bmatrix} \quad \mathbf{R}^n = \begin{bmatrix} \mathbf{R}^n_1 & \mathbf{0} & \\ \mathbf{0} & \mathbf{R}^n_2 & \\ & & \ddots \end{bmatrix} \tag{3.12}$$

In such a case, a tracker with S measurements, of S different cameras, will have a state vector with $n' = r \cdot S + e$ components, and $2(r \cdot S + e) + 1$ sigma points.

$$\mathcal{X'}^a_{k-1} = \left[\hat{\mathbf{x}'}^a_k \quad \hat{\mathbf{x}'}^a_k + \sqrt{(n+\lambda)\hat{\mathbf{P}'}^a_k} \quad \hat{\mathbf{x}'}^a_k - \sqrt{(n+\lambda)\hat{\mathbf{P}'}^a_k}\right] \tag{3.13}$$

$$\begin{array}{ll} W'^{(m)}_0 = \lambda/(n' + \lambda') & \\ W'^{(c)}_0 = \lambda/(n' + \lambda') + (1 + \alpha'^2 + \beta') & \\ W'^{(m)}_i = W'^{(c)}_i = 1/\left[2(n' + \lambda')\right] & \qquad i = 1, 2, \ldots n' \end{array} \tag{3.14}$$

The sigma point components can be divided into components derived from the state $\mathcal{X'}^x_k$, and components derived from the measurement noise $\mathcal{X'}^n_k$, which can be separated according to its measurement $1, 2, ..., S$: $\quad \mathcal{X'}^{n(1)}_k, \mathcal{X'}^{n(2)}_k, \ldots, \mathcal{X'}^{n(S)}_k$

Measurement matrix $\mathbf{H}$, which makes the transformation from state coordinates to measurement coordinates, is applied to obtain the measurement prediction sigma points $\mathcal{Y}$ from sigma points $\mathcal{X}$, and applies the gain.

$$\mathcal{Y}^{(s)}_{k|k-1} = \mathbf{H} \cdot \mathcal{X'}^x_k + \mathcal{X'}^{n(s)}_k \qquad s = 1, 2, \ldots, S \tag{3.15}$$

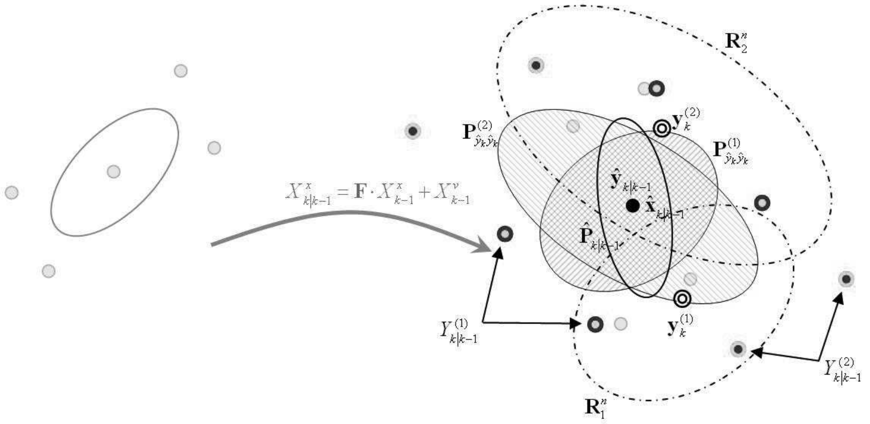

$\mathbf{R}_2^n$
$\mathbf{P}^{(2)}_{\hat{y}_k\hat{y}_k}$
$\mathbf{y}_k^{(2)}$
$\mathbf{P}^{(1)}_{\hat{y}_k\hat{y}_k}$
$X^x_{k|k-1} = \mathbf{F} \cdot X^x_{k-1} + X^v_{k-1}$
$\hat{\mathbf{y}}_{k|k-1}$
$\hat{\mathbf{x}}_{k|k-1}$
$\mathbf{P}_{k|k-1}$
$Y^{(1)}_{k|k-1}$
$\mathbf{y}_k^{(1)}$
$Y^{(2)}_{k|k-1}$
$\mathbf{R}_1^n$

Figure 3.5: Hypothetic sigma point distribution for measurements of two different cameras. These points adjust their positions to represent the measurement covariance placed on the prediction

Using these S sets of sigma points, we can obtain, for each measurement, the measurement prediction, the covariance prediction and the measurement-state cross-covariance.

$$\hat{\mathbf{y}}^{(s)}_{k|k-1} = \sum_{i=0}^{2n'} W'^{(m)}_i \mathcal{Y}^{(s)}_{i,k|k-1} \tag{3.16}$$

$$\mathbf{P}^{(s)}_{\hat{\mathbf{y}}_k\hat{\mathbf{y}}_k} = \sum_{i=0}^{2n'} W'^{(c)}_i \left[\mathcal{Y}^{(s)}_{i,k|k-1} - \hat{\mathbf{y}}^{(s)}_{k|k-1}\right] \left[\mathcal{Y}^{(s)}_{i,k|k-1} - \hat{\mathbf{y}}^{(s)}_{k|k-1}\right]^\top \tag{3.17}$$

$$\mathbf{P}^{(s)}_{\hat{\mathbf{x}}_k\hat{\mathbf{y}}_k} = \sum_{i=0}^{2n'} W'^{(c)}_i \left[\mathcal{X}'^{(s)}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right] \left[\mathcal{Y}^{(s)}_{i,k|k-1} - \hat{\mathbf{y}}^{(s)}_{k|k-1}\right]^\top \tag{3.18}$$

These equations are depicted in Figure 3.5, with a two camera example.

3.2.1.3 Estimation stage

First, a gain matrix for each measurement associated to the tracker is calculated.

$$\mathbf{K}^{(s)}_k = \mathbf{P}^{(s)}_{\hat{\mathbf{x}}_k\hat{\mathbf{y}}_k} / \mathbf{P}^{(s)}_{\hat{\mathbf{y}}_k\hat{\mathbf{y}}_k} \tag{3.19}$$

After that, measurements from different cameras must be combined to obtain a shared estimation (Figure 3.6). Weights $\beta^{(s)}$ play the role of combining the different measurements depending on their reliability. It is considered that weights are composed of two factors: the distance to the prediction, and the covariance of each measurement. Both factors are combined depending on the importance given to each one. Weights $\beta^{(s)}$ can be interpreted as a priori probability of each measurement.

The set of weights will be normalised, since the sum of the weights must be 1. Mean and covariance estimations will be:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \sum_{s=1}^{S} \beta^{(s)} \mathbf{K}^{(s)}_k \left(\mathbf{y}^{(s)}_k - \hat{\mathbf{y}}^{(s)}_{k|k-1}\right) \tag{3.20}$$

$$\hat{\mathbf{P}}_k = \hat{\mathbf{P}}_{k|k-1} + \sum_{s=1}^{S} \beta^{(s)} \mathbf{K}_k^{(s)} \mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k}^{(s)} \left(\mathbf{K}_k^{(s)} \right)^\top \tag{3.21}$$

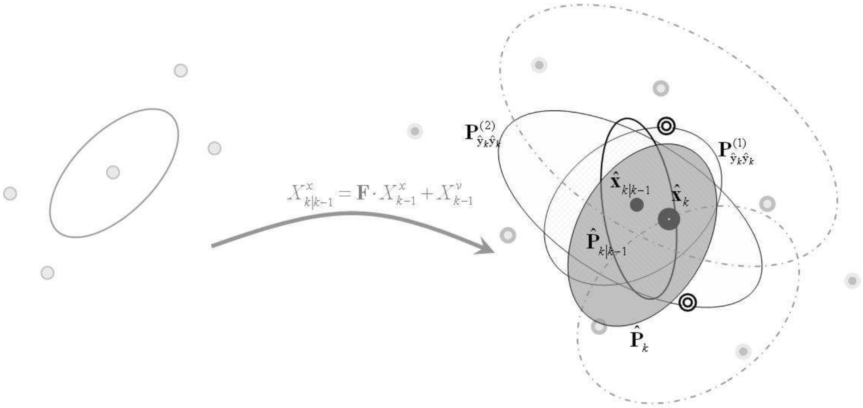

$\mathbf{P}^{(2)}_{\hat{\mathbf{y}}_k\hat{\mathbf{y}}_k}$
$\mathbf{P}^{(1)}_{\hat{\mathbf{y}}_k\hat{\mathbf{y}}_k}$
$X^x_{k|k-1} = \mathbf{F} \cdot X^x_{k-1} + X^v_{k-1}$
$\hat{\mathbf{x}}_{k|k-1}$
$\hat{\mathbf{x}}_k$
$\hat{\mathbf{P}}_{k|k-1}$
$\hat{\mathbf{P}}_k$

Figure 3.6: Graphic scheme which shows the estimation

3.2.2 MCUKF vs UKF Comparison

We can observe the MCUKF benefits in Figure 3.7, where a comparison between our multi-camera tracking algorithm and two independent UKFs is established. The comparison is made between the MCUKF resulting estimation and the mean of two independent estimations obtained with two single-camera UKFs. For low state noise R_v or measurement noise R_n levels, both algorithms give similar results. However, MCUKF obtains a lower mean square error when the noise level rises up.

3.3 Multi Camera Data Fusion on a Shared Reference

Algorithms like MCUKF allow us to combine measurements from different cameras. However, in order to integrate those measurements a shared reference is required. Once the single-camera detection algorithm has been selected, we need mechanisms for integrating the information of all cameras. Camera calibration allows obtaining the homographic transformation which relates each camera to the real world. It gives support to the multi-camera tracking algorithm but, in addition, it can also be used to reduce false positive of the observation process.

A usual reference is the groundfloor, modelled as a plan of the scenario, although other references such as one of the cameras can be used too. Thus, we can transform points from the image to the coordinate system of the plan. When the measurements from the cameras have been projected onto the plan, the tracking algorithm can be applied. The use of a plan as reference has several implicit advantages. The motion of an object in the image presents a distortion due to the perspective of the camera. The homographic transformation corrects this effect simplifying the search of an adequate motion model. Furthermore, the multi-camera tracking algorithm becomes more robust against occlusions in one or several cameras. Finally the combination of several noisy measurements gives us a more accurate estimation of the actual position. All these advantages are verified in Section 3.2.2.

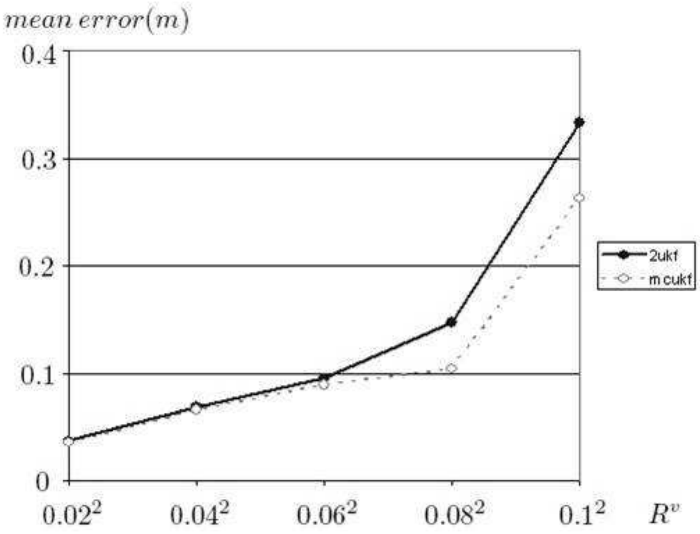


Figure 3.7: Mean square state error of MCUKF and two independent camera UKFs for different levels of state noise. A similar graphic is obtained for different levels of measurement noise.

3.3.1 Homographic Transformation

Taking a minimum of four points, the correspondence between the floor plane in the image and the plan of the field [Hartley and Zisserman, 2004] can be established (Figure 3.8). With this transformation, we can locate the position of people on the plan, assuming that the person is in contact with the floor. One homographic matrix must be calculated for each camera.

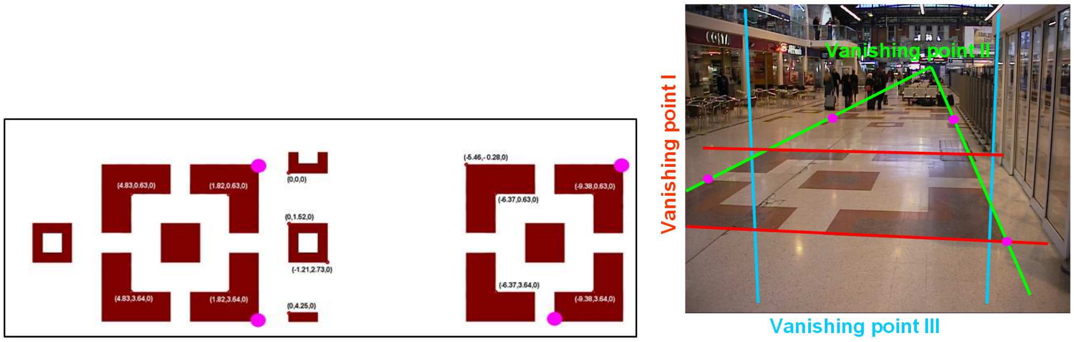


Figure 3.8: Camera Calibration and Vanishing Points Obtained.

Given the homographic matrices H, for each target location in the image, its location in the plan (x', y') can be extracted by transform the point of the target in contact with the floor, i.e. the foot point (x, y).

$$a = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \qquad x' = \frac{a(1)}{a(3)}, \qquad y' = \frac{a(2)}{a(3)} \tag{3.22}$$

Since measurements from different cameras have different reliabilities, we have to transform them too [Gómez et al., 2008]. We can define the reliability of a location in both axis using a covariance matrix $R = [R_{xx}, R_{xy}; R_{yx}, R_{yy}]$, which is easily transformed to the plan as follows:

$$Ax1 = H \cdot \begin{bmatrix} R_{xx} & R_{xy} & 0 \\ R_{yx} & R_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot H', \quad Ax2 = \frac{1}{a(3)^2} \cdot \begin{bmatrix} a(3) & 0 \\ 0 & a(3) \\ -a(1) & -a(2) \end{bmatrix}, \quad R' = Ax2' \cdot Ax1 \cdot Ax2 \tag{3.23}$$

This calibration methodology could seem simple in comparison with complex methods presented in the literature [Tsai, 1987]. Nevertheless, its accuracy is more than enough for punctual or regional tracking purposes, and its simplicity makes it suitable for surveillance application because it can be applied easily to different kinds and models of cameras working in the same scenario and whose parameters are unknown.

3.3.2 Multi Camera Tracking Evaluation

In addition to the advantage that our MCUKF has in comparison with conventional UKF (seen in Section 3.2.2), another advantage derived from the use of multi-camera algorithms is the compulsory use of a shared reference (plan). By tracking a target on the plan, we avoid the perspective effect that distorts the dynamic model, obtaining a more accurate prediction. In order to demonstrate this fact, we propose three tests. The objective of these experiments is to compare the performance of a tracker on the ground versus a tracker on the image. Two UKF trackers will be compared, using the same constant-velocity model. Although each one may have a different tuning, since coordinates in the image and coordinates in the ground represent different magnitudes, this tuning will be equivalent. Obtained results have been measured in the ground using Euclidean distance between prediction and measures. They can be seen in Table 3.1, in millimetres.

- **Test 1**: it compares the precision of short-term predictions D_s. In the test sequence, the object is moving towards and backwards the camera.

- **Test 2**: to compare the accuracy of medium-term predictions D_m, a test consisting in decimating the number of measurements has been carried out, being f the decimation factor.

- **Test 3**; to test the precision of long-time predictions D_l, a determined number n of consecutive measurements will be erased for both trackers. This test evaluates the capacity of recovering the object after an occlusion.

In this first test, the tracking on the ground has a great advantage over the tracking on the image, since the effect of the perspective deformation is avoided.

In the second test, mean distances grow as decimation factor increases, since the movement of the object is not totally predictable. The tracker on the image has an extra error originated by the deviation in velocity produced by the perspective effect. In the last test, the differences between both trackers are higher. The ground tracker does not loose the measurement even after an occlusion of 100 frames in length, maintaining short distances between prediction and measurement. However, the image tracker easily looses the measurement after an occlusion of 10 frames in length, giving very bad predictions.

Therefore, thanks to these tests, we have verified the advantages enumerated in this section. Two first tests confirm the profit due to correcting the perspective effect and applying an adequate motion model. Last test shows the capacity of the algorithm for dealing with occlusions.

Table 3.1: Mean distances in mm. between prediction and measure. D_s compares the precision of short-term predictions, D_m of medium-term and D_l to large-term.

	D_s		D_l				
	Fwd	Bkd	n=2	n=5	n=10	n=50	n=100
Image	54.90	44.02	79.17	107.3	260.7	855.9	3202.4
Ground	18.89	20.01	2.22	8.20	10.16	14.89	23.53

	D_m					
	f=1	f=2	f=3	f=5	f=8	f=12
Image	54.9	78.49	97.75	132.01	180.02	248.08
Ground	18.89	22.89	26.58	31.06	40.33	55.93

3.4 Observation: Motion Detection

The motion detector is based on a simple background subtraction followed by a thresholding. The reference background, that is, the appearance of the scene without people in it, is calculated and updated by a median filter [de la Escalera Hueso, 2001] over the last n frames, which allows adapting itself to changes in the lighting conditions. The resulting binary pixels compose blobs, which are regions of movement. A blob can be defined as a connected set of pixels in an image. In order to link those blobs, morphological operators such as erode and dilate [de la Escalera Hueso, 2001] are applied.

More complex background modelling methods can be used to address strong changes in the illumination, shadows or background movements, like it happens with trees in the wind. Multi-modal statistical motion detectors [C. Stauffer, 1999] models each pixel of the background with a mixture of several Gaussian distributions. However, they have the disadvantage of requiring a significant load of computational cost, which limits their use in real time applications.

3.4.1 Shadow Removal

Due to the application of the detector in indoor environments, where artificial illumination and multiple reflections may appear, and to improve the motion detection algorithm, a simple shadow removal algorithm based on hysteresis thresholding is used. The algorithm is inspired by [Rosin and Ellis, 1995] and it is a simplified version of the one introduced in [Herrero-Jaraba, 2005].

The basis of the algorithm is the calculation of the photometric gain. This gain is calculated as follows

$$Gain(x, y) = \frac{B(x, y)}{I(x, y)} \tag{3.24}$$

where $B(x, y)$ are the pixels in the background image and $I(x, y)$ are the pixels in the current frame.

By comparing the gain value of the pixel with three variance thresholds, the pixel is classified into four different categories: background, shadows, light objects and dark objects, as it is shown in Figure 3.9. All the pixels which composed the blob of the moving object and they have been classified as shadows are removed.

$$Class = \left\{ \begin{array}{cc} Lightobject & gain \leq \sum_1 \\ Background & \sum_1 \ < gain \leq \sum_2 \\ Shadow & \sum_2 \ < gain \leq \sum_3 \\ Darkobject & gain > \sum_3 \end{array} \right. \tag{3.25}$$

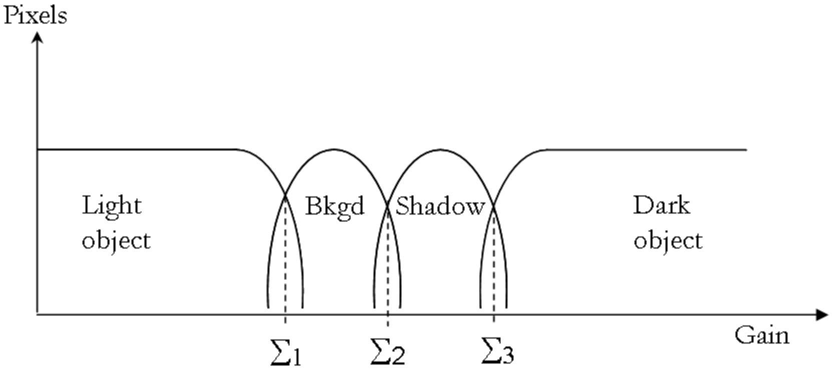


Figure 3.9: Pixel classification in an image on the basis of their variation with regard to the background

The thresholds are calculated by taking samples of background areas and possible moving objects. In this manner, and assuming background areas and objects as Gaussian distributions, thresholds are extracted calculating the mean and the standard deviation.

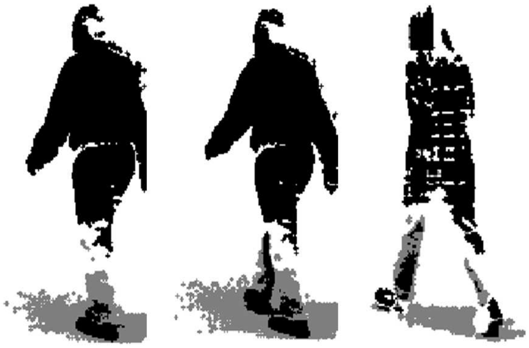


Figure 3.10: Results of applying the shadow removal. Shadows in gray

This algorithm permits to remove shadows and reflection in indoor or outdoor scenarios (see Figure 3.10). However, its performance is only good under soft illumination conditions. If the light conditions produce high saturated areas, such as bright reflections or high contrast between shadows and illuminated areas in a sunny day, which the automatic gain of the camera can not compensate, the algorithm is not able to distinguish between shadows and dark objects.

3.4.2 False Positive Reduction

As it has been mentioned, the motion detection algorithm returns a set of blobs which represent all the moving objects in the image. Nevertheless, some of them are not possible targets, but noise and distracters. While noise usually produces small blobs which can be eliminated by size, distracters are a more difficult issue.

By removing as many distracters as possible, the probability of tracking failures decreases drastically. Given that our main objective consists in human tracking, we propose two criteria based on human morphology to discard other moving objects, like cars or animals. The first one takes into account that the proportion between the human width and height is around one third. Assuming that the moving person is standing up, all the blob with a vertical size of three time the horizontal one are considered humans candidates. A percentage of tolerance is applied.

The second criterion is the knowledge of the height in pixels of a person in every place of the image. Knowing this information, we can compare the blob height in the scene with the height that a normal person (between 1.5 and 2 metres) would have in that pixel. If the difference is higher than a threshold, blobs are labelled as non-human. The algorithm developed to know the height of a human in pixel is called height estimator. As requirement, this method needs a simple camera calibration of the camera as shown in Section 3.3.1.

3.4.3 Height Estimator

Even a simple calibration of a scenario can provide a big amount of useful information which can be used to one's advantage to improve the least robust parts of the system. For instance, because of the poor quality of cameras in punctual tracking application, detection algorithm can produce mediocre results, with a lot of false positives. Since we are mainly interested on human tracking, a technique which allows us to discriminate between moving people and other dynamic objects or noise will simplify the filter task, increasing the global performance of the system. For this purpose, we propose the height estimation algorithm. Height estimator is a tool for obtaining the number of pixels which represents the average height of a person. Due to the perspective effect, this number is different depending on the location of the person in the image. We are able to ascertain this due to scene calibration [Criminisi et al., 2000]. In addition, the height estimator is also used to split blobs corresponding to two different people which have been linked due to the perspective of the camera.

First, we have to obtain a perpendicular plane to the floor which is defined with the four points used to calculate the homographic matrices. For this purpose, we have to extract four points of the walls, goal posts or any other vertical structure. Knowing both planes, we can calculate three vanishing points (2 horizontal ones and one vertical). These vanishing points permit us to project any point onto the coordinate axes, and elevate it vertically a number of pixels corresponding to the height in this point of the image. This number of pixels has been determined by a reference height in the image, that is, marking two points in the image, which are projected onto coordinate axes, and giving the real height in metres. Any vertical structure whose height is known can be used as reference. In the event that any real height in the training sequences is known, we will use a person of standard reference height, assuming everybody has the same height.

This methodology is able to return the head point (given the foot point), or return the height (given both points). We use the first application to determine the number of pixels which a person must have in this localisation.

The height estimator algorithm is shown in Algorithm 2 and Figure 3.11. Homogeneous coordinates are utilised to simplify the mathematical operations.

Algorithm 2: Height estimation algorithm

```
- Calculate the directional vector of the line $\mathbf{H2} - \mathbf{H1}$
      $v = \frac{y_{H2} - y_{H1}}{x_{H2} - x_{H1}}$
- Estimate point $\mathbf{H2}$ supposing the height of reference like 1,8 metres, and using the proportion between the number of pixels and the reference height in metres.
- Calculate the line $\mathbf{L_{H2-PFII}} = \mathbf{H2} \times \mathbf{PFII}$ in homogeneous coordinates, that is:
      $\mathbf{H2} = [x_{H2} \quad y_{H2} \quad 1]$
- Calculate the line $\mathbf{L_{A-PFI}} = \mathbf{A} \times \mathbf{PFI}$
- Calculate the axis $\mathbf{Y}$: $\mathbf{L_{H1-PFII}} = \mathbf{H1} \times \mathbf{PFII}$ where $\mathbf{H1}$ is the coordinate origin.
- Calculate the point $\mathbf{A'}$: $\mathbf{A'} = \mathbf{L_{A-PFI}} \times \mathbf{L_{H1-PFII}}$
- Calculate the line $\mathbf{L_{A'-PFIII}} = \mathbf{A'} \times \mathbf{PFIII}$
- Calculate the point $\mathbf{B'}$: $\mathbf{B'} = \mathbf{L_{A'-PFIII}} \times \mathbf{L_{H2-PFII}}$
- Calculate the line $\mathbf{L_{A-PFIII}} = \mathbf{A} \times \mathbf{PFIII}$
- Calculate the line $\mathbf{L_{B'-PFI}} = \mathbf{B'} \times \mathbf{PFI}$
- Calculate the point $\mathbf{B}$: $\mathbf{B} = \mathbf{L_{B'-PFI}} \times \mathbf{L_{A-PFIII}}$
- Calculate the height:
      $Height(x_A, y_A) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$
```

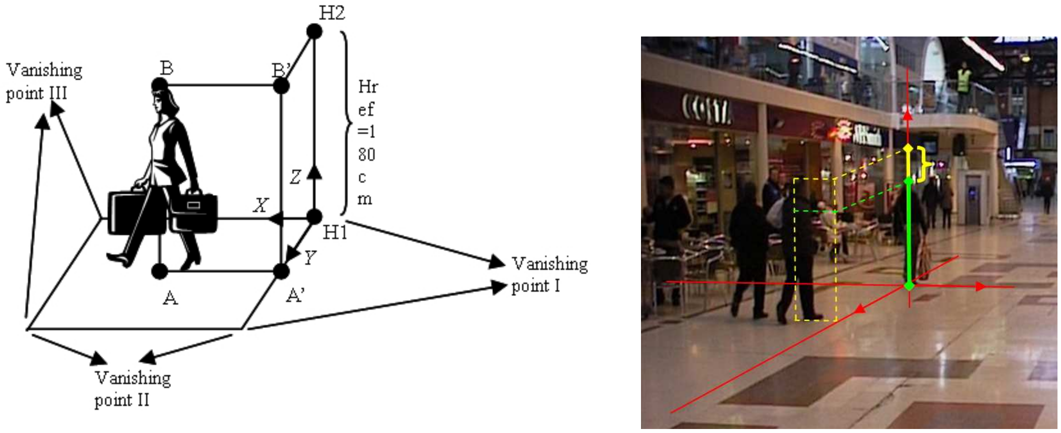


Figure 3.11: Height estimator.Coordinate system and comparison between a real measurement (yellow) and the reference height (green)

3.5 A Specific Application: Activity Recognition

In order to show the benefits of multi-camera tracking in general and MCUKF in particular, an activity recognition application, the detection of abandoned objects, is introduced.

The proposed system consists of two major parts: a multi-camera tracking algorithm based on UKF and a blob object detection system which identifies abandoned static objects or people. If any potential dangerous situation is detected, an alarm event is triggered.

3.5.1 Abandoned Luggage

Few works can be found about object-human interaction: In [Haritaoglu et al., 1999] the proposed system is able to detect if a person carries an object, and then the system tries to recognise them. M. Spengler and B. Schiele propose an approach [Spengler and Schiele, 2003] for detecting abandoned objects and tracking people using the Condensation algorithm in monocular sequences. A distributed surveillance system for the detection of abandoned objects in public environments is presented in [Foresti, 1999; Foresti et al., 2002; Sacchi and Regazzoni, 2000; Stringa and Regazzoni, 2000].

Smart surveillance systems have found an important application into public access facilities like airports and train stations, due to their strategic importance. However, ensuring high levels of security at public facilities is an extremely complex challenge. Several techniques must be applied to face all aspects of the security challenge, including crowd management, multi-camera people tracking and alarm generation. In this sense, one of the most complete systems is the European project ADVISOR. The ADVISOR [Siebel and Maybank, 2004] system aims at making public transport safer by automatically detecting criminal or dangerous situations. Using this system, people are tracked across the station and their behaviour analysed. Other specific work about public is shown in [Thirde et al., 2006], where a multi-camera surveillance and tracking system for monitoring airport activities is discussed.

In this section we present a method for detecting a particular kind of anomalous situation: abandoned objects. For this purpose, we have integrated three key areas of computer vision: video-based detection, tracking and object classification. By integrating them, being a novel multi-camera tracking algorithm based on UKF the cornerstone of both systems, we are able to detect and distinguish strange objects and people with dangerous behaviour, monitory them and trigger an alarm flag in the event of a potential dangerous situation.

The system identifies left luggage at the scene (suitcases, rucksacks, bags, etc.), identifies the person who has abandoned the baggage and sends an alarm if the object is abandoned for a period of time. The present approach can be divided into two parts: a detection stage consisting of locating left luggage and the person who abandoned it; and a tracking stage which integrates the detection results of several cameras in order to track the person and the object. A piece of left luggage is a static object which fulfils certain requirements in the same way that a person is a dynamic object which fulfils other requirements. Thanks to a couple of trackers we can know their positions at each time step and monitor them accordingly. A brief scheme is shown in Figure 3.12.

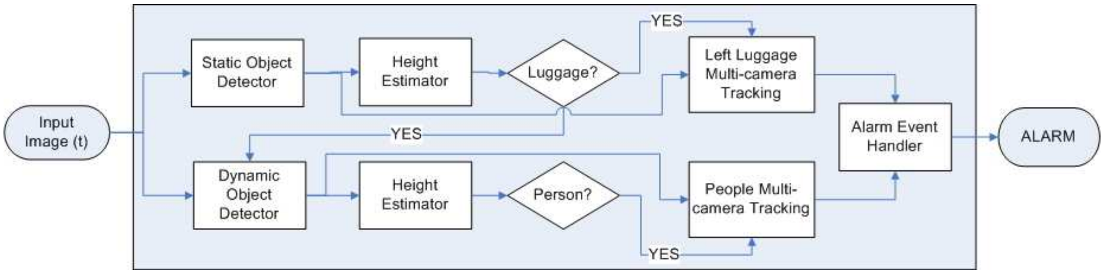


Figure 3.12: Abandoned Object General Scheme.

The algorithm has been tested with the PETS 2006 database (http://pets2006.net/) composed of sequences of video, in which people abandon luggage at a train station. This scenario, due to the presence of artificial constructions and fixtures, facilitates the extraction of reference points (even automatically), such as corners, and the calculation of the homographies (see Figure 3.8).

3.5.2 Observation

The observation stage is composed by a motion detector. No more complex visual clues can be applied due to the quality of the images, and our ignorance about the target to be tracked. Since we do not have a previous knowledge about the suspect, it is not possible to build a target model based on colour, for example.

Instead a conventional background subtraction algorithm, we propose a more advance methodology which allows us to distinguish between real moving objects and objects which has been moving recently. This technique is based on a double background subtraction between the current image and a short term and a long term backgrounds. This double background subtraction strategy was presented in [Herrero-Jaraba, 2005]. Originally it was developed to deal with slow and fast illumination changes as well as shadows and reflections. Here, it has been adapted to differentiate between moving objects and objects which have moved recently.

3.5.2.1 Static object detection

We can define a static object as an object which has been abandoned at the scene and which does not move, but was not there at the beginning. We have developed a method based on a double background subtraction which is capable of detecting this kind of objects [Herrero et al., 2003]. Long-term background represents a "clean" scene. All objects located in this background are not considered static objects: they are part of the scene. The long-term background is initialised with a temporal median filter of the initial frames and it is updated using a temporal median filter with a set of short-term backgrounds.

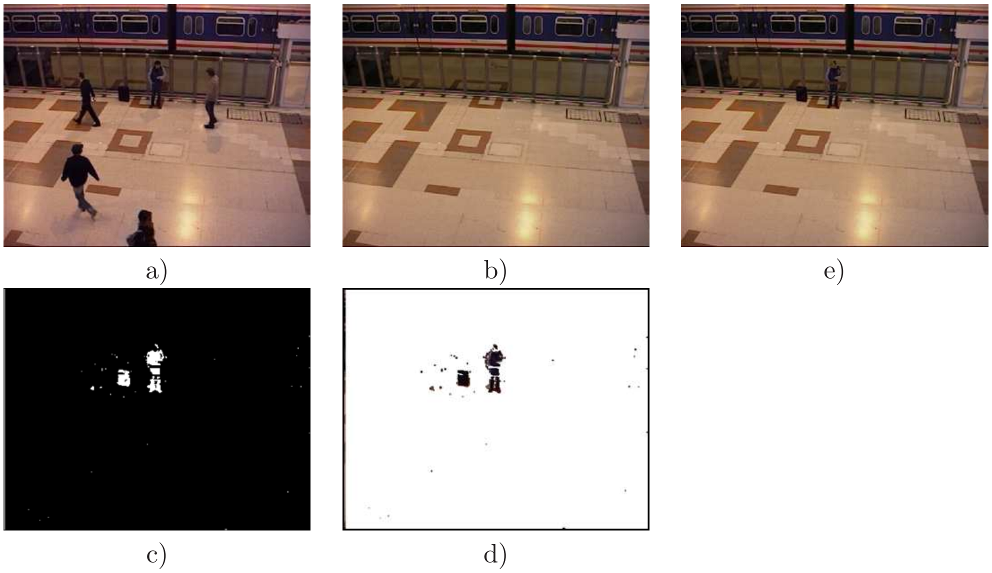


Figure 3.13: Short-term background extraction. a) Current Image, b) Last short-term background, c) Mask, d) Pieces, e) New short-term background.

Short-term background (Figure 3.13.e) shows static objects abandoned at the scene. This image is made up of the last background image (Figure 3.13.b) updated with pieces of the current image (Figure 3.13.a). These pieces (Figure 3.13.d) are then selected by a static object binary mask (Figure 3.13.c). This mask contains new blobs detected at the scene (given by a subtraction between the current frame and last stored background) and they are not currently moving (given by the opposite of a subtraction between current frame and the previous frame). The intersection between both subtractions is made in the blob level, not in the pixel level, that is, if blobs from both subtractions touch each other, they will be included in the static mask, even though they do not completely fit in.

Once both backgrounds have been calculated, their subtraction is accumulated. When the accumulation image rises above a value corresponding to a fixed time, the blob is classified as static object.

Left luggage requirements.

A static object will be considered as luggage if it fulfils several requirements. These requirements are a set of norms whose parameters can be changed in order to adapt the detection to different kinds of objects.

In our case we define a luggage item as a static object which has an area greater than a minimum area A_{min} (to eliminate noise), a maximum size equivalent to half the size of a person in the same point (the number of pixels of a person at this point will be obtained with the method explained in section 3.4.3) and a height-width ratio near one (we set a parameter $r_{H/W}$ which fixes the maximum error affordable).

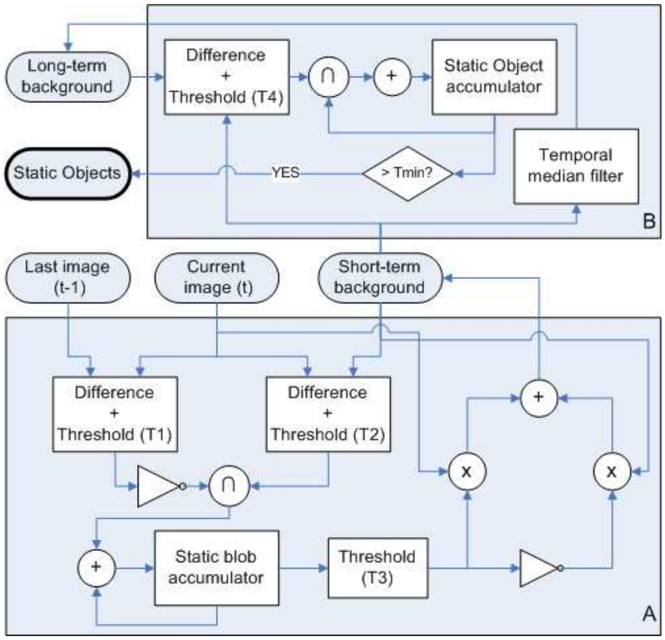


Figure 3.14: Static object detector. Block B represents the static object detection algorithm. Block A represents the short-term background creation algorithm.

3.5.2.2 Dynamic object detection

Dynamic object detection algorithm is a very simple process consisting of a subtraction and a binarisation between the current image and the long-term background. Short-term background is not used in the subtraction due to the fact that a person who waits a few seconds without moving will not appear in the resulting image.

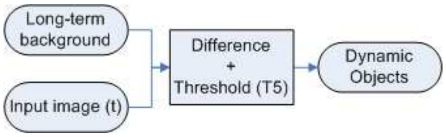


Figure 3.15: Dynamic object detector.

Person requirements.

We define a person as a dynamic blob with a height roughly equal to the number of pixels given by the height estimator (section 3.4.3). The other dynamic blobs will not be taken into account. A parameter r_{std_H} fixes the maximum admisible error to consider a blob as a person. However, blobs can be fragmented. In such a case, a blob corresponding to a person does not fulfil the height condition. In order to avoid this damaging effect, we apply an algorithm which groups blobs if they are in the bounding box corresponding to a person located in this position.

3.5.3 Modelling

A simple human model has been employed. By introducing x and y velocities in the state vector we can solve simple occlusions between tracked objects. We use a constant acceleration model and a motion dynamic which permits objects with variable trajectories.

$$\hat{\mathbf{x}}_{\mathbf{k}} = [\mathbf{x}\ \mathbf{v_x}\ \mathbf{y}\ \mathbf{v_y}] \quad \hat{\mathbf{x}}_{\mathbf{k|k-1}} = F \cdot \hat{\mathbf{x}}_{\mathbf{k-1}} \tag{3.26}$$

and the dynamic matrix will be given by

$$\begin{bmatrix} \mathbf{x_k} \\ \mathbf{v_k^x} \\ \mathbf{y_k} \\ \mathbf{v_k^y} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x_{k-1}} \\ \mathbf{v_{k-1}^x} \\ \mathbf{y_{k-1}} \\ \mathbf{v_{k-1}^y} \end{bmatrix} \qquad R^v = \begin{bmatrix} t^3/3 & t^2/2 & 0 & 0 \\ t^2/2 & t & 0 & 0 \\ 0 & 0 & t^3/3 & t^2/2 \\ 0 & 0 & t^2/2 & t \end{bmatrix} \tag{3.27}$$

With this configuration parameters, the measurement matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.28}$$

3.5.4 Multi-Camera Tracking

Once static and dynamic objects have been detected in each camera, we project all the measurements onto the plan, see Figure 3.16, in order to have a shared reference space. Each blob is projected converting it in a single point, the point which is in contact with the floor (the lowest point). Note that those people that occlude each other due to the camera perspective are not classified as human and, therefore, they are not projected as valid dynamic object for this camera.

The covariance matrix, which characterises the reliability of the location, is assumed circular and equal for all the blobs in the image. In this manner, only the homographic error differentiates the quality of the measurement. As this error grows with the distance to the camera, it is given priority to those cameras which perceive the target more clearly. Due to the fact that the perspective introduces more uncertainty in the vertical axis, and therefore a larger error than in the perpendicular axis, the transformed covariance can be represented as a ellipse enlarged in the camera axis (see Figure 3.16).

When these transformations have been made, the multi-camera tracking is applied: one tracker for the luggage item, and other for the owner. However, each tracker receives the measurement from a different source. While the static object tracker uses static object blobs like measurement, the dynamic object tracker uses dynamic object blobs.

A simple matching algorithm to establish the correspondences between cameras in the shared reference has been used: the filter selects the nearest measurement of each camera when there are several measurements for each camera. However, if the distance between the nearest measurement and the tracker is higher than 5.99 (assuming a confidence level of 95% in accordance with the chi square test) or two metres, noone is selected.

3.5.5 Results

Our system has been tested using several sequences of PETS 2006 database representing different situations, a diverse number of people and different types of luggage. This algorithm obtained the best result [Ferryman et al., 2006] among all the participants in the Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance.

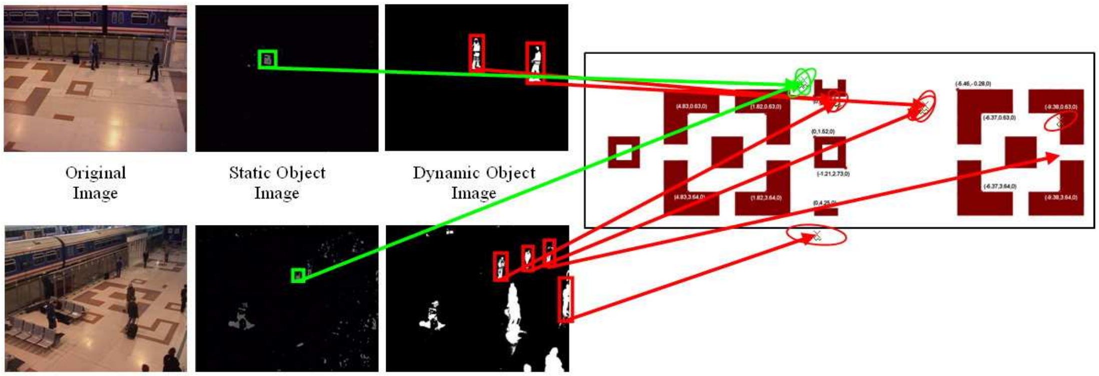


Figure 3.16: Shared reference plan for all cameras. Once objects have been detected in each camera, all the measurements are sent to the plan. In green: detected luggage items. In red: detected people.

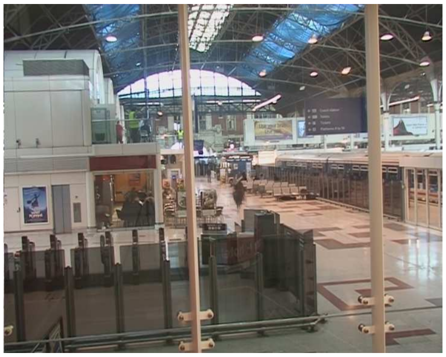
Once we know the positions of the person and the abandoned baggage in the plan coordinates, we can measure the distance between both objects easily and act accordingly.

This application establishes two circular zones around the static object with a radius a and b. The luggage is attended to by the owner when they are within a distance a metres from the luggage. A luggage item is unattended when the owner is further than b metres from the luggage. In this moment, an alarm event is set up, and a time counter is activated. When an item of luggage has been left unattended by the owner for a period of t consecutive seconds (in which time the owner has not re-attended to the luggage and nor has the luggage been attended to by a second party), the alarm event is triggered. The distance between a and b is determined to be a warning zone where the luggage is neither attended to, nor left unattended. This zone is defined to separate the detection points of the two states, reducing uncertainties introduced due to calibration and detection errors in the sensor system. When the owner crosses the line at a metres, a warning event is set up to trigger the event after a time t. Configurations parameters are $a = 2m, b = 3m, t = 30sg$. Cameras 1, 3 and 4 have been utilised. Camera 2 has been rejected due to its poor resolution. In order to justify this decision, we have calculated the homographic error for each camera using the covariance of the estimated homography [Criminisi et al., 1999]. In Figure 3.17, we can see how the poor resolution and the far-off view produce bad measurement.

Object detection thresholds (Figures 3.14 and 3.15): $T1 = 10, T2 = T4 = T5 = 30, T3 = 230, Tmin = 100, A_min = 150, r_{H/W} = \pm 5\%, r_{std_H} = \pm 25\%$.

Results for sequences 1, 2, 3 and 7 can be observed in Table 3.2 and 3.3, and in Figures 3.19, 3.20, 3.21. The green dot represents the piece of luggage, the red one the owner and the red line the trace of the subject once he has abandoned the object. We can see how the static object and the subject and correctly tracked in spite of occlusions and multiple people thanks to the robustness of the system.

The benefits obtained by a multi-camera system are shown in the Figure 3.18. When the target is only viewed by one camera, due to occlusions among other causes, the accuracy of the tracking system falls out drastically.



Camera	$Error^2X$	$Error^2Y$
1	0.72	3.08
2	2.75	**27.57**
3	2.12	2.06
4	5.12	10.02

Figure 3.17: Camera 2 image and homographic mean square error for each camera (in pixels).

Table 3.2: Numerical results test sequences

Sequence	Luggage item (x,y) [metres]	Warning trigger [sec (frame)]	Alarm trigger [sec (frame)]
S1-T1-C	(0.161,-0.338)	113.0 (2825)	113.6 (2840)
S2-T3-C	(0.458,-0.431)	91.32 (2283)	91.84 (2296)
S3-T7-A	(1.041,-0.458)	-	-
S7-T6-B	(0.336,-0.391)	92.32 (2308)	93.96 (2349)

3.6 Conclusions

In this chapter, we have introduced the concepts about punctual tracking. Their drawbacks have been remarked as well as its utility for specific application, specially in video-surveillance, and the different solutions to improve the performance. Thus, multi-camera systems, calibrated scenarios and robust motion detection algorithms have been postulated as successful strategies to be employed in this tracking domain.

Moreover, in section 3.5.1 we have proposed a system capable of detecting abandoned or left objects. When this happens, the owner is found and tracked until the static object moves or the owner goes out of the observed region. If the owner abandons the object for a time greater than a specified time, an alarm event is triggered. The presented approach is not based on tracking all people at the scene, which does not constitute the main goal of this research and would demand unnecessary computational resources. Instead, we identify when an object has been unattended and proceed to track the person closest to this object.

A static object detector based on a double-background subtraction has been developed on the basis of the algorithm presented in [Herrero et al., 2003], which can detect left luggage or other static objects in any scene. Furthermore, the height estimator reduces the number of false positives in the observation. Finally, we have applied a tracking method which enables us to manage several sensors to track the same object.

Results applied to a real scenario monitored by 4 distributed cameras show an accurate detection method once threshold parameters have been tuned for the scenario under consideration. In addition, the tracking algorithm is robust to distracters, and allows for dealing with occlusions provided that the tracked object is isolated in at least one of the views.

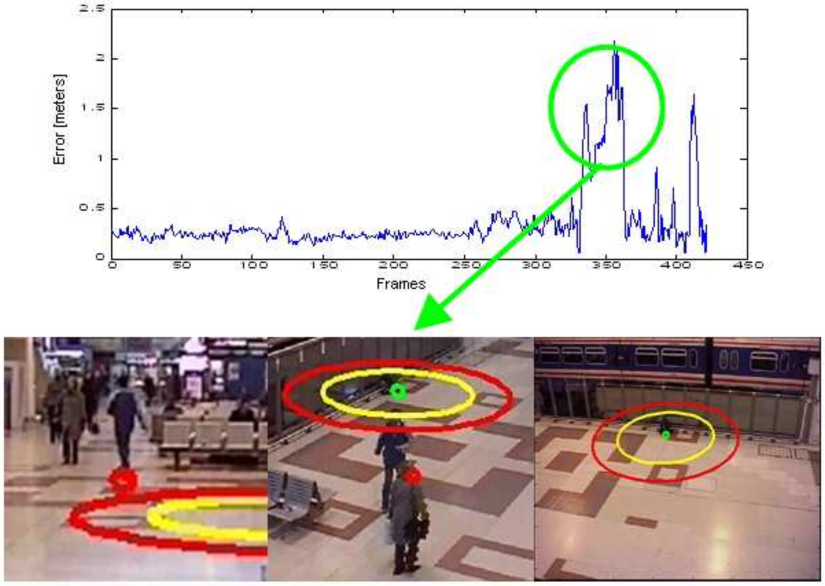


Figure 3.18: Dynamic Object error for sequence 1, and frames which produce a high error rate.

Table 3.3: Test sequence errors. (*FAR=False acceptance rate)

Seq.	Static Object Error [metres]	Dynamic Object Error [metres]	Time Error [sec (frs)]	FAR[%]
S1-T1-C	0.12202	0.359262	0.06 (1.5)	3.94
S2-T3-C	0.14815	0.548645	0.06 (1.5)	3.15
S3-T7-A	0.20297	0.279636	-	0.0
S7-T6-B	0.18444	0.289107	0.18 (4.5)	0.23
Mean Error	0.1644	0.369162	0.1 (2.5)	1.83

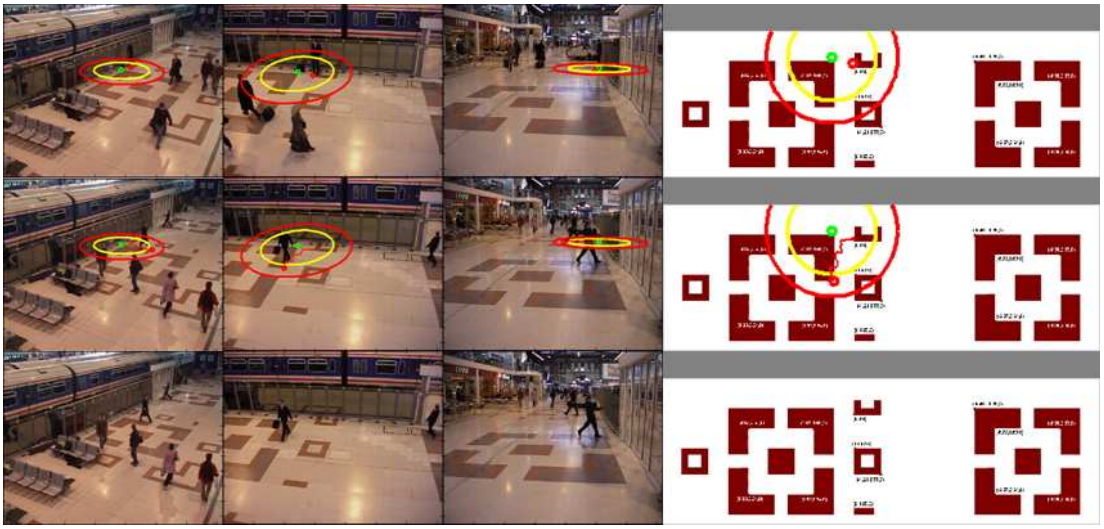


Figure 3.19: Results for sequence 3 (S3-T7-A).

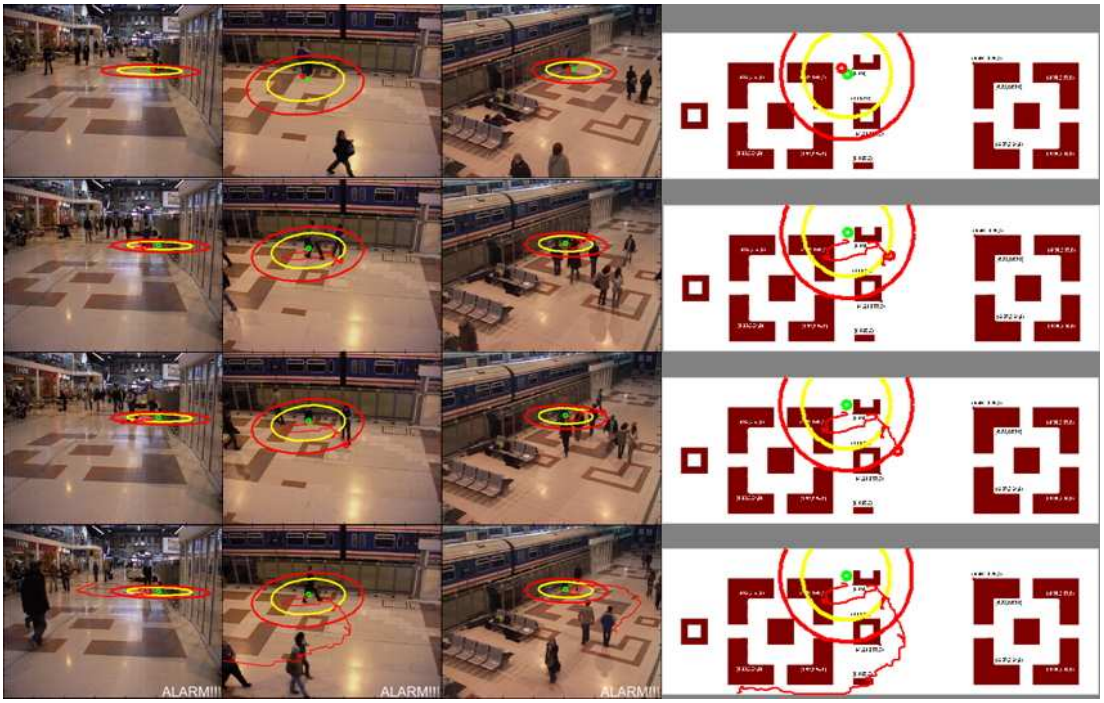


Figure 3.20: Results for sequence 7 (S7-T6-B).

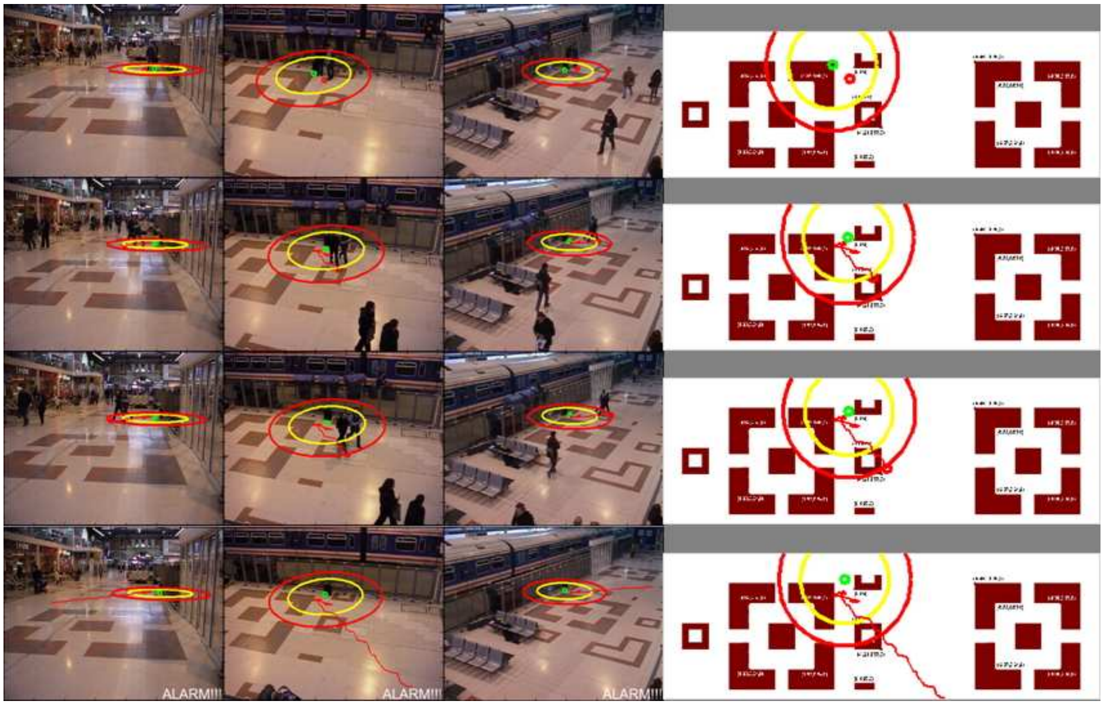


Figure 3.21: Results for sequence 2 (S2-T3-C).

4

Region Domain Tracking

"There are always flowers for those who want to see them."

-Henri Matisse-

Region tracking is the next level domain after punctual tracking. It is characterises by modelling the target as a region (a rectangle or an ellipse) or a set of connected regions. Therefore, it covers all the spectrum between the punctual and the articulated tracking, from one region to a complex layout which models deformable targets.

The main difference with regard to punctual tracking is the existence of a model which gives us information about the size, proportions, orientation and layout of the target. More parameters must be estimated simultaneously and, therefore, problems associated to larger dimensional spaces appear, such as local minima. More complex modelling requires more complex features in order to solve the ambiguities that this kind of modelling implies. Colour, gradients and motion are common features used in tracking.

The usage of complex features, based on prior models, and a larger probability to fall in local minima make the tracking domain more suitable for multiple hypothesis algorithms such as Particle filters or Monte Carlo algorithms. Moreover, multiple hypothesis tracking simplifies and makes more efficient the combination of measurements from different regions by converting them to probabilities.

Region tracking application goes from video surveillance to biometrics. Cars, faces, sport men and pedestrians are target which can be modelled easily like a region or a set of regions.

In this chapter we will explore the different kinds of features which can be used for region tracking. A particularisation for colour tracking is presented due to its flexibility and generality. A complete framework for this feature tracking is proposed in order to increase the performance and efficiency at the same time that the computational cost is decreased. In addition, its application to visual surveillance and the advantages that it implies is shown. Finally, the problem of updating the feature model is addressed. This chapter has been structured in the same way that the Figure 2.1: Observation, Filtering and Modelling.

4.1 State of the Art

This level of abstraction tries to identify and track over time one or multiple connected regions associated to every moving object in the sequence. A cross-correlation measurement is used. It is traditionally the most widely approach and results are reasonably good even in the presence of partial occlusions. Tracking a target composed of several regions will give as a result a more robust algorithm against noise and occlusions.

The region based tracking strategy allows us to perform a hierarchical sub-division: regions, people and groups. A region is defined like a part of the body whose pixels have a feature in common. A person is composed of one or more regions which fulfil the geometric structure constraint of the body model. A group consists of people grouped together by a common characteristic.

This tracking domain involves a better integration with the feature extraction being the tracker the responsible of making the feature matching. The quality of the results depends to a great extent on the selected features. Given that high level features (colour, omega models, contours) are more difficult to extract than low level features (points, lines, motion), there is a trade-off between tracking efficiency and feature complexity which is defined by the particular necessities.

The simplest example of region tracking consists in dividing the target in several regions which are modelled as a punctual target or using really simple features. Polana et al. [Polana and Nelson, 1994] model a person by a bounding box divided into a region grid. Thus, even when an occlusion happened, tracking of the centroid is still successful. Segen et al. [Segen and Pingali, 1996] divide the silhouette into four corner points which are used as features to be tracked by matching between successive frames.

When the complexity of the scenario demands more sophisticated methods, more advanced features must be used for each region. Jang and Choi [Jang and CHoi, 2000] characterise each region combining shape, texture, colour and edges. Tracking is done by a Kalman filter and the final estimation is performed by minimising a feature energy function during the matching. Intille et al. [Intille et al., 1997] use a zenithal camera to track several people in a close environment, tracking each one by matching motion blobs using colour and location features. Mckenna et al. [McKenna et al., 2000] propose an algorithm in which the bounding box of the target can merge or split into several regions. In this way, combining a region tracker with individual colour appearance model for each region, a good track of multiple people, composed of multiple parts, is achieved, even with self-occlusions. The measurement is obtained with a background subtraction method that combines colour and gradient information.

Modelling region with more complex features allows us to characterise univocally each one. The success of the global tracking does not depend on the relative location of each region regarding the others, as in the case of simple features, being the global performance more robust against partial occlusions. Colour has been one of the most widespread features, and it is considered a vital cue for segmentation and tracking of human beings, as long as extra computational cost is not a major concern. Heisele et all [Heisele et al., 1997] clusterise automatically (using k-means) the object in a set of clusters whose centroids are tracked in the next frames. The clustering process categorises the subject basing on colour in RGB space. Fieguth and Terzopoulos [Fieguth and Terzopoulos, 1997] divide the face and torso of a person in a set of 6 colour region. An explicit occlusion model permits to deal with severe partial occlusions. Wren et al. [Wren et al., 1997] consider the human body as a set of region automatically extracted and usually representing the different body parts: head, torso and limbs. The motion blobs associated to each region are obtaining using a foreground extraction algorithm in which background and subject have been modelled like a mixture of Gaussian distributions. By tracking the blobs of each region, the global tracking of the person is successfully achieved.

Comaniciu and Ramesh [Comaniciu et al., 2000; Comaniciu et al., 2003] proposed a new paradigm for the efficient colour-based tracking of objects seen from a moving camera. The proposed technique employs the mean shift analysis to derive the most similar target candidates for a given target model, while the prediction of the next target location is computed with a Kalman filter. The dissimilarity between target model and target candidates is expressed by a metric based on the Bhattacharyya coefficient. The resulting tracking is robust to simple partial occlusion, significant clutter, target scale variations, rotations in depth, and changes in camera position. However it is not able to deal with multi-modal distributions.

Nummiaro, Koller-Meier and Van Gool [Nummiaro et al., 2002; Nummiaro et al., 2003] proposed one of the most influential paper in regional tracking. The paper presents the integration of colour distributions into particle filtering which had typically used edge-based image features until that moment. The advantages of colour distributions are their robustness to partial occlusion, and the rotation and scale invariance. These characteristics make colour a perfect complement for the particle filter which can model multi-modal distribution to deal with partial occlusions. A target colour model, which must be obtained in the initialisation, is used for comparing each hypothesis of the particle filter using Bhattacharyya distance. Perez et al. [Pérez et al., 2002] extend this idea within a probabilistic framework. This probabilistic approach is very flexible and can be extended in a number of useful ways. In particular, in three different ways: multi-part colour modelling to capture a rough spatial layout ignored by global histograms i.e. multi-region, incorporation of a background colour model when relevant, and extension to multiple objects.

From this last approach as basic pillar for regional tracking, we will show the problems that it implies and develop alternative proposals to cope with them.

4.2 Modelling

Three different target aspects should be modelled: the structure of the target, described in the state vector, the dynamic model and the appearance of the target which permits to evaluate the hypothesis. In this section, we have focused on colour because it is one of the most versatile and general feature to identify univocally a target.

4.2.1 Structural Model

Since we are in the region domain, the structural model that we have applied is quite simple. If the appearance of the target is uniform, like in face tracking, a single rectangular region is enough. Otherwise, two or more region have been combined. In the event that more than one region compose the model, an appearance model for each of them will be needed.

4.2.2 Dynamic Model

Given the unpredictable nature of the targets in video surveillance, complex dynamic models are not really useful to solve the tracking problem. Even they can make worse the final result. Instead, we will use a simple linear model with constant acceleration, being the multiple hypothesis tracking with the stochastic noise the responsible of a successful estimation.

4.2.3 Appearance Model: Colour

Essential information about the target can be obtained on the bases of colour modelling and extraction techniques. Colour is one of the most discriminative cue which is able to differentiate between object and background, but also between objects. Moreover, its combination with motion features, which can not distinguish between objects by themselves, is very useful to eliminate background areas with the same colours than the objects, for instance, lines which define a football field and white t-shirts in a football context.

Colour model techniques, which have the property of being invariant to rotation or scale variations, need a prior reference, which is extracted from the first frame of the sequence. This initialisation is usually obtained manually, although motion detection algorithms can help us to automatise this process. We have used several HSV and RGB colour spaces for modelling the target, depending on the particular sequence that we have processed and its environment conditions. Some of the techniques here presented can also be applied to gray level image, in which the colour model can be considered as a texture model.

To model colour density functions there are two possibilities: parametric and non-parametric. The major advantage of non-parametric approaches is the flexibility to represent complicated densities effectively. However, they suffer from high memory requirements and computational complexity. On the contrary, parametric approaches simplify the target feature modelling and reduce drastically the computational cost needed to process them, but the assumptions introduced limit their application to simple distributions.

Parametric models reduce the colour distribution of the target to easily parameterisable functions like a Gaussian function or a mixture of several Gaussians. There are many parametric density representations, for instance McKenna et al. [McKenna et al., 1997] uses Gaussian mixture models in hue-saturation space to model object's colour distributions. They propose an adaptive learning algorithm used to update these colour models over time. In [McKenna et al., 1997; Stauffer and Grimson, 2000], authors suggest Gaussian Mixture Models, but their method requires knowledge about the number of components. In [Bohyung Han, 2004], authors propose a density approximation methodology where the density is represented by a weighted sum of Gaussians, whose number, weights, means and covariances are automatically determined.

Unlike parametric models, non-parametric density estimator is a more general approach that does not assume any specific shape for the density function, so it is able to represent very complicated densities effectively. They estimate the density function directly from the data without any assumptions about the underlying distribution. As mentioned in [Bohyung Han, 2004], this avoids having to choose a model and estimating its distributions parameters.

Histogram is the simplest non-parametric density estimator. However, histograms exhibit some problems: not smooth, dependence on end points of bins and dependence on width of bins. One way to alleviate the first two problems and, at the same time estimating the underlying density without having to store the complete data, is the kernel density estimation technique (KDE) [Elgammal et al., 2002]. However, an incorrect usage of KDE algorithms can even makes the problem worse. On the one hand, KDE techniques are suitable when the number of target pixels is lower than the size of the feature space. On the other hand, if the number of target's colour pixels is high or the feature space dimension are low, then the histogram approach would give better results.

In our experiments we have used both parametric and non-parametric techniques such as GMM, histogram or kernel density estimation. In video-surveillance applications, a colour histogram is an adequate solution to characterise each target, due to its simple initialisation and the necessity of a model which identify each target univocally i.e. with as many details as possible to differentiate the targets. However, sport provides an environment with plain colours known previously that can be reused for more than one target. So, the usage of a parametric model such as a Gaussian or a Gaussian mixture model to generate the object distribution would be more advisable.

4.2.3.1 Parametric methods: GMM

As parametric method, we have developed a segmentation algorithm based on Gaussian Mixture Models in the colour space. In this manner, a new input frame is projected onto the target probability space to generate the Probability colour Density Image (PDI). We will generate a PDI for each kind of object that we want to track. In our particular case, we are going to explain the results for sport segmentation problems, specifically for football matches.

Therefore, we need two PDIs, one for each team, but more PDIs could be generated: two PDIs for each team if clothes have complex colours, or even one for each person in a video surveillance application. The values of the PDI pixels are taken from the Gaussian mixture models of the targets used as classifier for each pixel in the input picture. The probability assigned to each pixel is based on the distance (using as metric the Mahalanobis distance) to the nearest Gaussian. The exact metric is given by:

$$PDI^n(x) = \left(1 + \frac{min\{\delta(x,i)\}_{i\in G^n}}{min\{\delta(x,i)\}_{i\in G}}\right)^{-1} \tag{4.1}$$

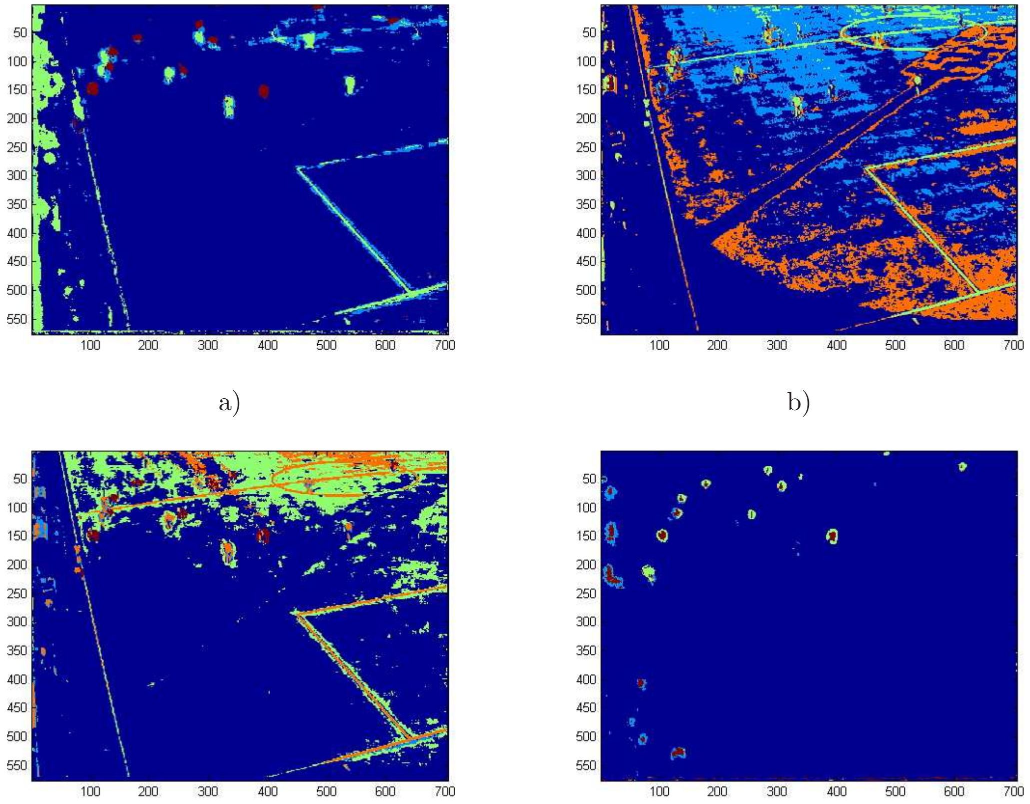
with $\delta(x,i) = (x-\mu^i)^T \cdot (\sigma^i)^{-1} \cdot (x-\mu^i)$ where x is the value of the pixel at coordinates (x,y), n is the target, G is the whole GMM and G^n are the gaussians assigned to target n.

GMM Building In order to generate the Gaussian mixture model, we extract pixels of both foreground and background. Each pixel is converted to the HSV space to reduce the influence of changing illumination and shadows, but other colour spaces can be used (see Figure 4.1). Using EM algorithm, we obtain several Gaussians which model the whole colour space of the environment. The number of Gaussians is chosen depending on the necessities. Many validation indexes can be applied [Yang and Wu, 2006; Kim and Ramakrishna, 2005] for this purpose.

Fuzzy C-means Cluster analysis is the process of clustering a data set into groups of similar elements. A lot of literature exits about this topic, which can be considered one of the major techniques in pattern recognition. Traditionally, conventional clustering algorithms (also called hard clustering) restrict the membership of an element to a unique cluster. However, the development of the fuzzy logic produced the appearance of fuzzy clustering, with an idea of partial membership. Like in conventional clustering, it is necessary to pre-assume the number c of clusters algorithms, which is in general un-known. Fuzzy c-means (FCM) is one of the most well-known and used method. We have decided to apply it after a brief comparison with other methods like hard c-means. A more detailed description of the algorithm is given in Appendix C.

In real data analysis, noise and outliers are unavoidable and make the clustering more difficult. In order to avoid these problems, one may process a Principal Component Analysis (PCA) which eliminate redundancy, noise and non-discriminative components. Furthermore, PCA normalises input data, simplifying the comparison between characteristic spaces. On the other hand, the existing validation indexes can loose their efficiency, as we can see in the following section.

Due to the high complexity of the colour feature space, and the noisy and changing training samples, we have applied a PCA previous to the FCM. Moreover, the best results are obtained by generating two different data spaces: one for background and another one for foreground. This is due to the fact that, if a unique colour space to background and players is generated, results are confused because Gaussians models tends to model shadows instead of foreground. Thus we have created a half-supervised clustering framework. Different colour spaces have been explored (Figure 4.1), being the best the HSV space due to its shadow invariance.



a)

b)

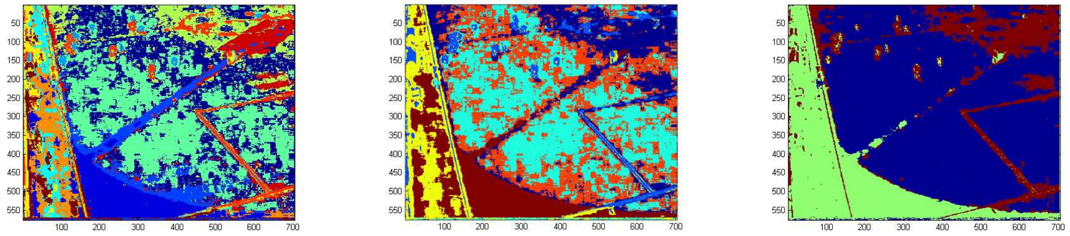
c)

d)

Figure 4.1: GMM for colour clustering. The number of Gaussians has been chosen manually, obtaining the best results with 4 foreground Gaussians and 2 background Gaussians. a) HSV, b)YcbCr, c)RGB, d)Normalised RG

But this methodology, as we have mentioned before, does not only allow us to obtain a hard decision. By computing the Mahalanobis distance to the corresponding Gaussian, we can obtain an image in which the value of each pixel is the probability of belonging to this cluster. An example of the clustering and the PDIs can be seen in the Figure 4.3.

Validation Indexes As it is shown in Figure 4.2 the goodness of the results depends on the number of Gaussians, among other parameters. Therefore, once the segmentation method has been chosen, we use the validation indexes for fuzzy clustering [Yang and Wu, 2006; Kim and Ramakrishna, 2005] to estimate automatically the most propitious number of clusters which accurately presents the structure of the data set. Several approaches have been tested and their results are shown in Appendix C. However, all the previous CVIs have not given robust and coherent results between them. This can be due to the fact that the colour space is complex and has arbitrary shapes, and this fact produces that no index guarantees the correct election. Figures C.2.b and C.3.b show how the number of gaussians chosen to maximise the index value are not able to modelise the space correctly, and more Gaussians would be necessary for a correct representation.



a) b) c)

Figure 4.2: GMM for colour clustering varying the number of gaussians a) 12, b)6, c)3

However, we can obtain a more coherent clustering introducing auxiliary information besides the colour. In our case, we introduce motion as a clue to check the coherence of the clustering. In addition, using the motion as a mask for filtering the training colour samples simplifies the clustering task thanks to a cleaner characteristic space.

We have developed a simple index, called Motion Validation Index (MVI) based on the usage of motion to validate the colour segmentation. This index is calculated for each Gaussian as the number of pixels which are classified into that cluster (I_{gauss}) and have been detected as movement (I_{mov}) divided by the total number of pixels which are classified into that cluster. Those Gaussians which obtain the higher values are selected like corresponding to the foreground,

$$MVI_{gauss}(i) = \frac{\sum_{x\in X}\sum_{y\in Y} I_{gauss}(x,y,i) \cap I_{mov}(x,y)}{\sum_{x\in X}\sum_{y\in Y} I_{gauss}(x,y,i)} \qquad \forall i \in G \tag{4.2}$$

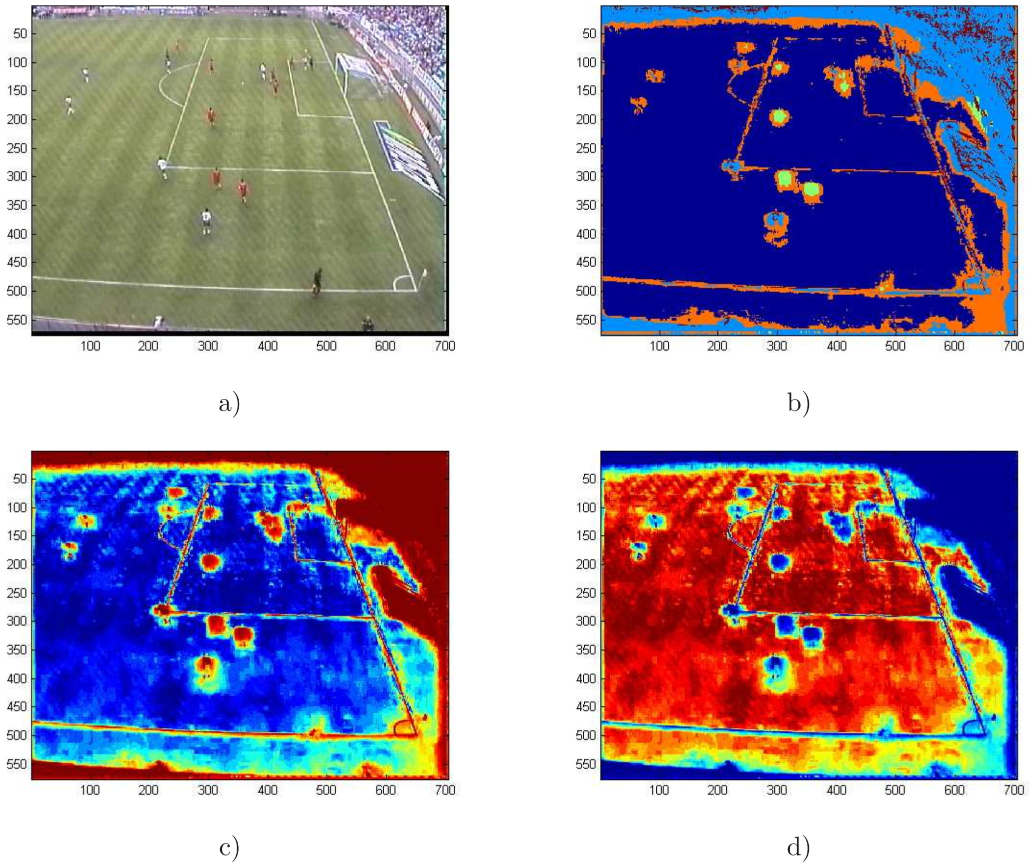
where G is the whole GMM and X and Y are the size of the images to be processed.

We can see robust and coherent results in the Table 4.1 and Figure 4.4. In order to facilitate the figure 4.4 understanding, we have to say that most pixels corresponding to both teams: the first team is modelled by gaussian 1 (4 gaussians) or 5 (6 gaussians) and the second team by gaussian 4 (4 gaussians) or 1+4 (6 gaussians). The rest of the pixels corresponds to noise, halos or pieces of background, which are more spread in the colour space and conform a sparse distribution. This is because few samples have been extracted to sample their distribution properly. Therefore, the gaussians do not fit exactly their distribution but a mean approximation is obtained. However we are not interested in a perfect modelling of this noise pixels, our goal is to discriminate them from the target appearance and this fact is achieved with this number of gaussians.

Table 4.1: MVI validation index with variable number of Gaussians

	$i \in [1,4]$				$i \in [1,6]$					
	i=1	i=2	i=3	i=4	i=1	i=2	i=3	i=4	i=5	i=6
MVI(i)	0.307	0.007	0.045	0.506	0.219	0.027	0.049	0.495	0.315	0.050

Several factors can affect to the correct estimation of the percentages. In the particular case of football segmentation:



a)

b)

c)

d)

Figure 4.3: Clustering and PDIs for a football sequence. a) Original Image, b)Clustering, c)Foreground PDI, d)Background PDI

- Shadows are classified as movement and therefore as player.

- Lines in the ground are static but are usually into player cluster.

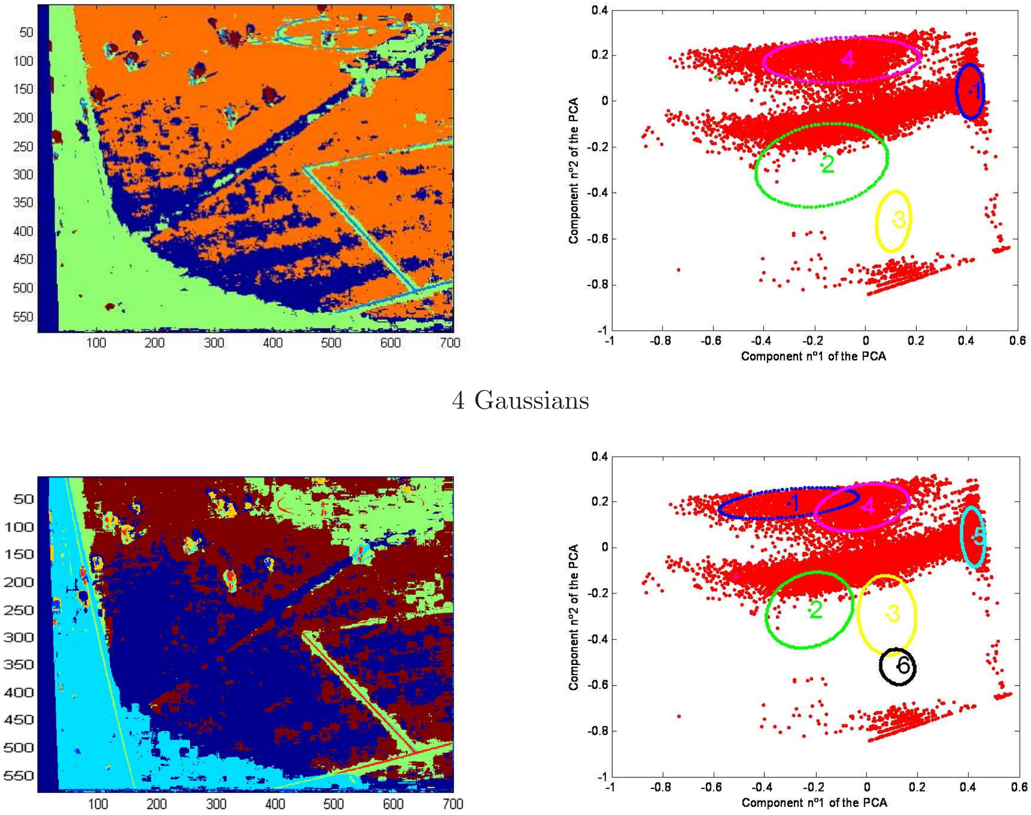
- Video compression blurs the players and introduces ground samples into the motion pixel group.

- Motion blobs are rough, then including halo.

However, these error are not significant, and the index does not require a critic or accurate threshold.

The MVI index also checks that we have selected an adequate number of Gaussians to model the whole colour space. If the sum of the MVI indexes corresponding to the Gaussians associated to each football team decreases when we split or merge the Gaussians, we will stop and select the optimum number of Gaussians (Table 4.2).

$$MVI_{Tot} = \frac{1}{c_{target}} \sum_{i \in target} MVI(i) \tag{4.3}$$



4 Gaussians

6 Gaussians

Figure 4.4: MVI clustering results for 4 and 6 Gaussians.

where c_{target} is the number of Gaussians belonging to the whole GMM G, which have been classified as part of the target model.

Table 4.2: MVI validation index to determine the optimum number of Gaussians

	4 Gaussians (2+2)	6 Gaussians (4+2)	10 Gaussians
MVI_Tot	0.2962	0.4338	0.3360

In a scenario with static cameras, the inclusion of motion information is a simple task which can be performed using a background substraction technique. If we wanted to extend this index to dynamic cameras, the motion information would require more complex motion algorithms, which measure the relative motion, like for instance, optical flow.

4.2.3.2 Non parametric methods

Non parametric modelling techniques allow us to parameterise colour or texture models more accurately, since these techniques do not assume Gaussian or mixed-gaussian models. On the other hand, their computational complexity grows exponentially.

Histogram comparison Histogram comparison is the best known colour evaluation technique. It was integrated with the particle filter in [Nummiaro et al., 2002; Nummiaro et al., 2003]. Authors extract the histogram of a certain region and compare it with a reference histogram using the Bhattacharyya distance as metric. Histogram can be applied to colour images or gray-level ones. In gray-level images, the information that histogram contains can be understood as texture or density. Colour histograms are more discriminative than black and white ones, but their calculation implies a huge computational cost. In order to reduce this effect, the measurement is made over the sub-sampled colour histograms.

A very low level of sub-sampling (still using many bits to represent each colour channel) will produce low values in the comparison, even between very similar and correct objects. On the other hand, a high level of sub-sampling will accept erroneous matching between considerably different objects.

The histogram is a colour distribution which defines the number of pixels in a region for each one of the possible colours. That is, it assigns one of the m-bins of the histogram to a given colour at location x_i, being $i \in Image$. The value of the histogram should be normalised by the total number of pixels of the region in order to obtain invariance to size and allowing the comparison between regions of different scales.

Note that all the pixels are equally important to describe the objects. However, pixels further away from the region center can get smaller weights by employing a weighting function. This is because pixels in the middle of the region have a more pure colour whereas pixels near the border are prone to get mixed up with background colours or get occluded. A usual example of weighting function is the Epanechnikov kernel [Nummiaro et al., 2002] (see Figure 4.5).

$$k(r) = \begin{cases} 1 - r^2 & if \quad |r| < 1 \\ 0 & otherwise \end{cases} \tag{4.4}$$

where r is the distance from the region center.

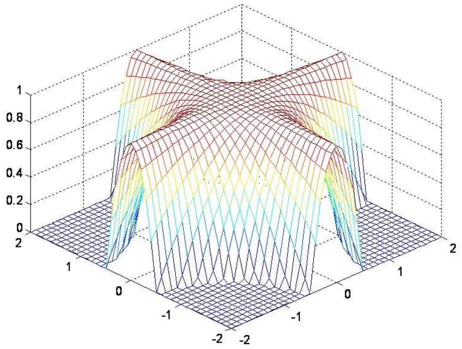


Figure 4.5: Epanechnikov kernel

Then, the new histogram is calculated as

$$q_u(y) = C \cdot \sum_{i \in I} k\left(\frac{\|y - x_i\|}{r}\right) \delta[h(x_i - u)] \tag{4.5}$$

where δ is the Kronecker delta function, h is the conventional histogram and the parameter r is the radius of the kernel i.e. $\sqrt{W^2 + H^2}$ with W and H the width and the height of the region respectively. C is a normalization factor which ensures that $\sum_{u=1}^{m} q_u(y) = 1$

$$C = \frac{1}{\sum_{i \in I} k\left(\frac{\|y - x_i\|}{r}\right)} \tag{4.6}$$

Measurement metric The measurement distance is the similarity function which defines the difference among target model and candidates. The Bhattacharyya distance measures the distance between two distributions. It is the most used metric because of its computational simplicity. The Bhattacharyya distance between two normalised histograms p, q can be calculated as follows, being ρ the Bhattacharyya coefficient.

$$d = \sqrt{1 - \rho} \qquad \rho = \sum_{u=1}^{m} \sqrt{p(u) \cdot q(u)} \tag{4.7}$$

Instead of this distance, we can apply a new metric that we have called minimum distance, which is more discriminative since Bhattacharyya distance takes account of shared colours in both histograms, whereas minimum distance increases the importance of the disparity of values corresponding to these shared colours (Figure 4.6). This distance also has the desirable properties for being a metric: exists for all discrete distributions p and q, is positive, symmetric, and is equal to one if $p = q$.

$$\rho = \sum_{u=1}^{m} \min p(u), q(u) \tag{4.8}$$

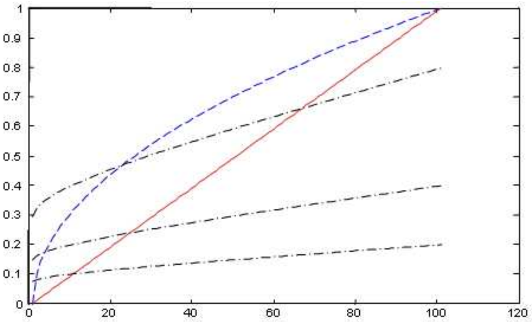


Figure 4.6: In blue, results applying Bhattacharyya coefficient between a histogram with only one colour and another with a variable percentage in this colour. In red, results with minimum coefficient. In black, results with exponential to the power of Bhattacharyya coefficient and σ equal to 0.5, 1, 2.

Kernel Density Estimation However, histograms exhibit some drawbacks: not smooth, dependence on end points of bins and dependence on width of bins. One way to alleviate the first two problems and, at the same time estimating the underlying density without having to store the complete data, is the kernel density estimation (KDE) technique [Duong, 2001]. Applying KDE, the distribution function is modelled as

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} w_i K_\sigma(x - x_i) \tag{4.9}$$

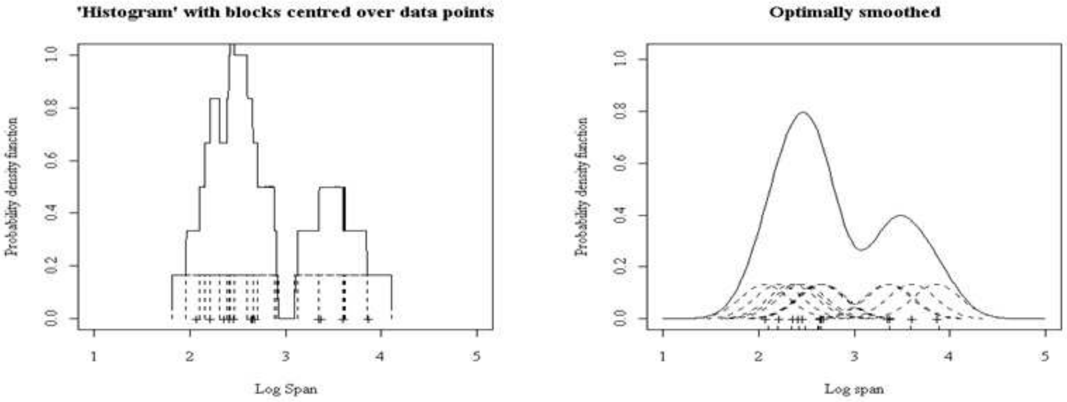


Figure 4.7: Kernel density estimation from a conventional histogram. (Image source: http://www.maths.uwa.edu.au/∼duongt/seminars/intro2kde/)

Typically, the Gaussian kernel is used (Figure 4.7). In this sense, it is worthy to note that choosing the Gaussian as a kernel function is different from the distribution of a Gaussian model. The method requires a parameter σ which regulates the kernel width (Figure 4.8).

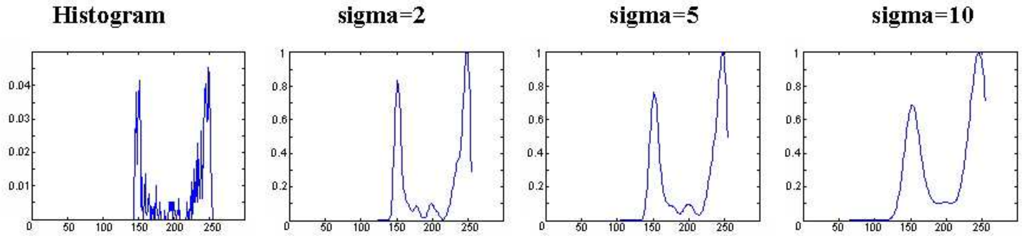


Figure 4.8: KDE results varying σ parameter

The algorithm makes possible better results with larger sub-sampling levels, reducing thus the computational cost. However, this computational charge moves from the number of histogram bins to the number of samples to be computed.

Both previous methods allow the comparison between two image patches whose histograms are extracted. However, in order to accelerate the computation and generate the prior information (Section 4.3.2), we also need obtain a rough approximation of the membership of a single pixel to the model. We have obtained the required images using different techniques (Section 4.4.1.2).

Fuzzy-Logic colour models This technique is an alternative method to the both previous ones. Considering each channel histogram as a probability density function fdp and calculating the right-left accumulated fdps $f_I Dp$ and $f_D Dp$, we can obtain the membership functions μ for each channel (Figure 4.9).

$$\mu_1 = \frac{fdp}{\max{(fdp)}}, \qquad \mu_2 = \min\left(1, \frac{f_I Dp}{0,25}, \frac{f_D Dp}{0,25}\right), \qquad \mu = \frac{c_1 \cdot \mu_1 + c_2 \cdot \mu_2}{c_1 + c_2} \tag{4.10}$$

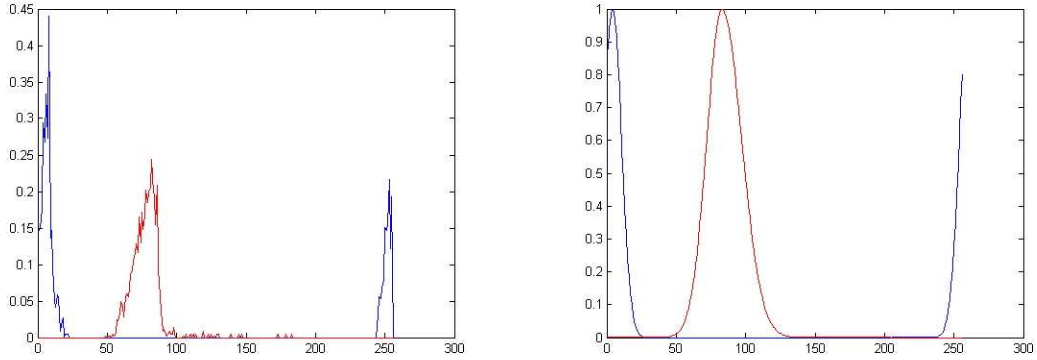


Figure 4.9: Input histogram with two colour channels, and resulting model obtained with fuzzy logic.

Using the degree of membership as probability and applying a threshold, we can obtain the required PDI (see Figure 4.10)

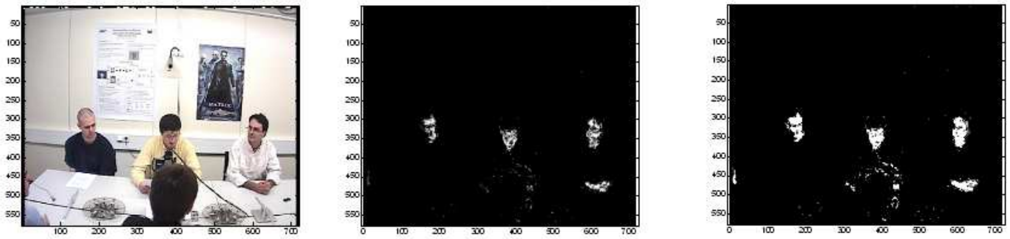


Figure 4.10: Result of the Fuzzy colour modelling for skin segmentation.

In order to model more complex colour distributions like multi-modal distribution, we can apply T-conorms, a concept frequently employed in fuzzy logic (see Figure 4.11).

4.2.3.3 Conclusions

Colour is, without discussion, one of the most important features to identify univocally a target. The election between parametric or non-parametric techniques will depend on our necessities and the particular application of our system, being both of them useful and valid due to the advantages that both methodologies entails.

Due to its flexibility, we will use colour as main feature for our tracking algorithm. However, the same framework can be extended for other features or for the combination of all of them (see Chapter 6).

4.3 Filtering

Contrary to punctual tracking, in which the simplicity of the model and the observation force us to assume Gaussian and uni-modal distributions, region tracking, much closer to reality, must cope with multi modal distributions if we want to exploit all the potential that this domain contains.

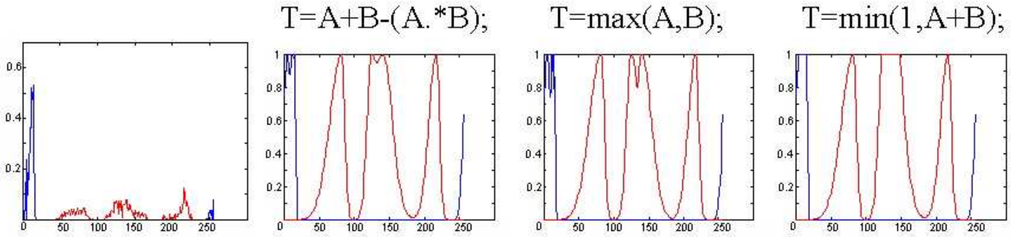


Figure 4.11: Fuzzy model for multi-gaussian distribution using T-conorms.

Particle Filter is the most representative and broadly extended multiple hypothesis algorithm. It exhibits good performance even in situations in which the target is partially occluded or several distracters appear in the scene [Isard and Blake, 1998a; Nummiaro et al., 2003; Pérez et al., 2002; MacCormick and Blake, 2000]. Particle filter can be regarded as a hypothesis tracker that approximates the filtered *a posteriori* distribution to a set of weighted hypotheses called particles. It weights the particles on the basis of a likelihood function and distributes them in accordance with a motion model.

This tracking algorithm is especially useful for region tracking due to two main factors:

- The capacity for dealing with non-gaussian and multi-modal distributions, which usually appear as the complexity of the model grows.
- The probabilistic treatment of the observation, which provides a suitable framework to employ more complex features and to integrate the measurement of several regions simultaneously.

In this manner, even though several regions could be occluded, the measurement of the other regions in combination with the multi-hypothesis methodology allow us, theoretically, to solve the occlusion.

However when the scene/situation is complicated due to occlusions, clutter or bad dynamic models, the performance decreases significantly unless a large number of particles are used. This increases the computational cost and time performance and affects further surveillance functionalities.

This section sets out a framework for a fast and efficient colour-based tracking using Particle Filter algorithm based on different and complementary sampling techniques [MacCormick and Blake, 2000; Isard and Blake, 1998b; Rui and Chen, 2001; Pitt and Shephard, 1999; Torma and Szepesvari, 2004; Okuma et al., 2004]. This proposal is composed of three main modules, depicted in Figure 4.12:

- the *Observation Process* to generate *a priori* probability and evaluate in a fast way the likelihood,
- the *Importance Sampling* technique which uses this *a priori* probability to improve the proposal distribution,
- a *Partitioned Sampling* to split the target state and reduce the number of particles involved.

The main purpose of Importance sampling is to improve the proposal distribution [Isard and Blake, 1998b; Rui and Chen, 2001; Okuma et al., 2004] and so, reduce the cost of

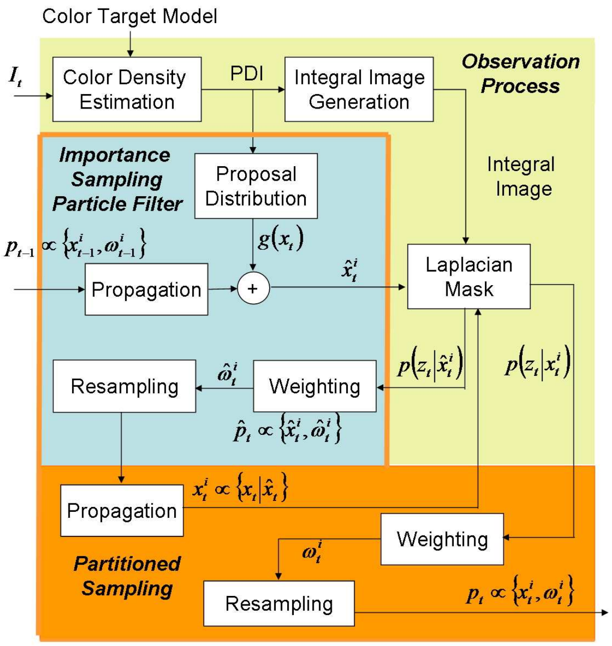


Figure 4.12: Diagram overview.

evaluating extremely low likely particles. These particles occurare due to poor prior density estimation, particularly when the motion of the object is badly modelled by the dynamic model. By including the current observation in the proposal distribution this undesirable effect is avoided.

The present approach employs *a priori* probability, i.e., a proposal distribution that introduces information about the current observation. Consequently, the probability density function is obtained to describe an observed target. The tracked object appearance is modelled by acquiring sample pixels from the object to represent the feature distribution. The probability of any observation is then given by using feature density estimation methods [Silverman, 1986]. In this way, the Probability Density Image (*PDI*) is calculated, where each of these pixels points its membership to the target model. On the basis of the conclusions exposed in the previous section, a particularisation of this methodology for colour tracking is presented due to its flexibility and generality.

The use of the *PDI* yields to an additional advantage when particle weights are evaluated. The calculation of the likelihood $p(z|x)$ of a certain observation z given a hypothesis x of the state of the system involves the integration over a set of observation points. To accomplish this task efficiently, the Integral Image [Viola and Jones, 2001; B.Han et al., 2005] of the *PDI* is calculated in advance. Therefore the integration is reduced to a simple combination of sums, increasing the time efficiency of evaluating the likelihood of all particles.

The second sampling technique, partitioned sampling [MacCormick and Blake, 2000], was originally developed to track various objects utilising a unique filter. This approach reduces the number of particles which grows with the state dimensionality. In this paper the Partitioned sampling is used to decompose the required parameters of the target (location, scale, shape, etc.). The estimation of the first parameters facilitates to estimate the remaining ones in a second round, requiring less particles than a joint estimation.

4.3.1 Particle Filter

Target tracking can be expressed as a Bayes filter [Ripley, 1987], in which the posterior distribution $p(x_t|z_{1:t})$ is updated recursively over the target state x_t given all previous observations $z_{1:t}$ up to time t, according to:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|z_{1:t-1}) dx_{t-1} \tag{4.11}$$

The integral is generally not solvable analytically, so numerical methods have been applied in the majority of cases. Monte Carlo methods approximate the target density, that is the probability of being target in each point of the state space, by number of samples $\{x_t^i, \omega_t^i\}_{i=1}^N$ that are distributed on a target density basis, where ω_t^i is the weight for particle x_t^i and are approximations to the relative posterior probabilities (or densities) of the hypothesis. The filtering distribution is given by:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \sum_{i=1}^{N} \omega_{t-1}^i \cdot p(x_t|x_{t-1}^i) \tag{4.12}$$

Isard and Blake [Isard and Blake, 1998a] developed the Condensation algorithm (also called Particle filter) which is one of the most extended methodology applied to track multi-modality and

non-linear and non-Gaussian models [Isard and Blake, 1998a; Nait-Charif and McKenna, 2003; Nummiaro et al., 2002; Vermaak et al., 2003a; Nummiaro et al., 2003].

The particle filter can be understood as an importance sampler for this distribution. In this way, N samples x_t^i are drawn from a proposal distribution q (also called importance density). Proposal distribution controls the layout of the hypothesis in the dimensional space. Therefore, the proposal distribution is used to perform the sampling and to evaluate the likelihood and the transition probability. A prior set of samples is generated and the importance weights are computed iteratively as:

$$\omega_t^i \propto \omega_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)} \tag{4.13}$$

The choice of an adequate proposal distribution $q(x_t^i|x_{t-1}^i, z_t)$ is one of the most critical design issues. The optimal proposal distribution is given by the true conditional state density, i.e. $q(x_t^i|x_{t-1}^i, z_t) = p(x_t^i|x_{t-1}^i, z_t)$. In this regard, the proposal distribution includes support of true posterior distribution as well as most recent observations. However, sometimes this approach may be unfeasible or computationally too expensive.

Bootstrap particle filters use the state transition prior $q(x_t^i|x_{t-1}^i, z_t) = p(x_t^i|x_{t-1}^i)$ as the proposal distribution to place the particles, since it is intuitive and can be easily implemented [Isard and Blake, 1998a; Nummiaro et al., 2003; Pérez et al., 2002]. Since this probability does not take into account the most recent observation z_t, all the particles may have low likelihood, contributing to an erroneous posterior estimation. Therefore, the exclusive use of the transition probability, as proposal distribution, makes the algorithm prone to be distracted by background clutters.

The transition probability $p(x_t^i|x_{t-1}^i)$ is computed as $x_t^i = F \cdot \tilde{x}_{t-1}^i + w_{t-1}$, where F is a transition matrix which contains the relationships between the state variables to evolve the state over time, and w_t is a stochastic noise which introduces variance and uncertainty in the hypothesis to explore a higher range of possibilities.

Let's describe in some detail a particle filter iteration for single object tracking. At time t, the filter receives $p(x_{t-1}|z_{t-1})$, the *pdf* of the state vector x_{t-1}, at time $t-1$. This distribution is approximated by a set of samples x^i_{t-1} , with associated weights ω^i_{t-1}. Given the set $\{x^i_{t-1}, \omega^i_{t-1}\}^N_{i=1}$, the new location is estimated using the following standard procedure:

> **Resampling:** The set $\{x^i_{t-1}, \omega^i_{t-1}\}^N_{i=1}$, is resampled (sampling with replacement) according to the weights ω^i_{t-1}. We obtain the new set $\{\tilde{x}^i_{t-1}, \tilde{\omega}^i_{t-1} = 1/N\}^N_{i=1}$.
>
> **Propagation:** Particles are propagated to the new set through the application of a random dynamic model $x^i_t = F \cdot \tilde{x}^i_t - 1 + w_{t-1}$, where F defines the deterministic and w_{t-1} the stochastic component.
>
> **Weighting:** Finally, using some external measurement on the feature z^i_t, samples x^i_t are weighted in order to obtain the output of the iteration t, that is, $\{x^i_t, \omega^i_t\}^N_{i=1}$,approximating $p(x_t|z_t)$.

The final estimation can be worked out as $E[x_t] = \sum^N_{i=1} \omega^i_t \cdot x^i_t$.

Resampling step is used to avoid the problem of degeneracy of the algorithm, that is, avoiding the situation that all but one of the importance weights are close to zero. The performance of the algorithm can be also affected by proper choice of resampling method. The stratified resampling proposed by Kitagawa [Kitagawa, 1996] is optimal in terms of variance.

4.3.2 Importance Sampling

An alternative approach consists in sampling the observation to improve the efficiency of particle filter. A function $g(x^i_t)$, which introduces information about the current observation, is then applied as proposal distribution. The particles are consequently distributed using this proposal instead of the transition probability (Fig. 4.13). The idea is to avoid the generation of low-weight particles which provide a poor contribution to the posterior probability.

This implies that every particle in this stage is not generated from a previous one, and therefore, the particles cannot be paired with the previous ones to compute the probability $p(x^i_t|x^i_{t-1})$. So, an additional factor $f_t(x^i_t)/g(x^i_t)$ is applied to guarantee that the importance sampling has no effect on the consistency of the posterior approximation. Since particles are generated from $g(x^i_t)$ instead of the prediction distribution, the correction ratio f/g maintain the temporal coherence and the particle set still approximates $p(x_t|z_t)$.

$$\omega^i_t \propto \frac{p(z_t|x^i_t) f_t(x^i_t)}{g_t(x^i_t)}, \qquad \text{where} \qquad f_t(x^i_t) \equiv p(x^i_t|z_{t-1}) \tag{4.14}$$

The prediction density $f_t(x^i_t)$ represents the probability of appearance, which is obtained using the weighted mean over all possible transitions due to the fact that the new samples are not generated from the prior.

$$f_t(x^i_t) = \sum^N_{j=1} \omega^j_{t-1} \cdot p(x^i_t|x^j_{t-1}) \tag{4.15}$$

Note that this modification implies that the dynamical model is not only used but it is also evaluated. Although the sum in equation (4.15) increases the complexity of the algorithm from $O(N)$ to $O(N^2)$, the real effect is negligible in practice because the computational cost of this stage (for practical values of N) is dwarfed by the time expended on the observation process for instance. Moreover, a more efficient particle set implies a lower number of particles and therefore, a smaller complexity growth in this stage.

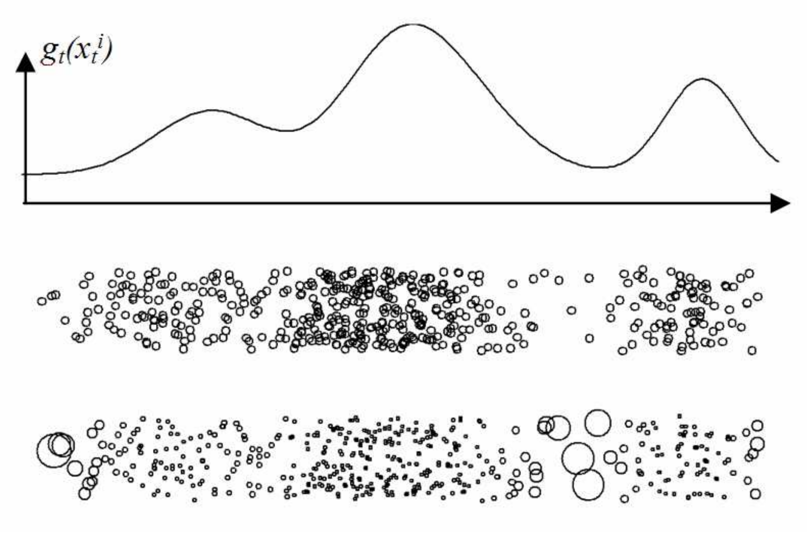


Figure 4.13: Improving the proposal distribution with a function $g_t(x_t^i)$. Top: Importance function. Middle: Particles drawn proportionally to the proposal distribution function. Bottom: Particles with the compensation factor included.

This approach balances the particle distribution and ensures that the importance function does not distort the calculation of the posterior probability $p(x_t|z_t)$. In this way, any proposal distribution can be chosen if the number of particles N is large enough.

In practice, the proposal distribution derives from a rough observation process and might produce errors and imperfect estimations. In this regard, it is recommended to add a percentage of particles by conventional sampling, as shown in Algorithm 3.

In order to generate better proposal distributions, there are two main approaches: direct and indirect ones. While indirect approach [Isard and Blake, 1998b] applies an auxiliary tracker based on secondary observations for generating an updated and more accurate proposal distribution, direct approach [Rui and Chen, 2001] also improves the proposal distribution but using the same observation employed to estimate the posterior probability. Therefore, indirect approaches seem more advisable due to the fact that they introduce new and uncorrelated information in the tracker algorithm. However, indirect approaches have two non-negligible limitations. First, finding auxiliary trackers and additional sensors is not possible most of the times. Second, the auxiliary tracker itself needs a good proposal distribution if we want that it helps the main tracker. Otherwise, it will produce worse estimations. For both reasons, we have chosen an option based on direct approaches, which consists in an imperfect observation process.

Our approach is similar to [Isard and Blake, 1998b], where an auxiliary tracker generates a more accurate proposal distribution using secondary observations. On the other hand, our proposal uses the same features than the likelihood function to improve the sampling process due to the two important reasons exposed before: the availability of finding auxiliary observations and the reliability of their proposal distributions.

By making proposals which have high conditional likelihood, we reduce the costs of sampling many times from very low likelihood particles, improving the statistical efficiency of sampling procedures. This means that the number of the particles can be reduced substantially.

As proposal distribution function $g(x_t)$ we will use a rough and fast approximation of the

Algorithm 3: Particle filter based on an efficient proposal distribution

```
Given a particle set $\{x^i_{t-1}, \omega^i_{t-1}\}^N_{i=1}$ which represents the posterior probability
$p(x_{t-1}|z_{t-1})$ at time $t-1$

  1. Generate $N$ new samples from the importance function $x^i_t \sim g(x_t)$.

  2. Compensating factor: weight the particles
     $\hat{\omega}^i_t = f_t(x^i_t)/g_t(x^i_t)$

  3. Multiply by likelihood: weight the particles
     $\omega^i_t = \hat{\omega}^i_t \cdot p(z_t|x^i_t) = p(z_t|x^i_t) \cdot f_t(x^i_t)/g_t(x^i_t)$
     and normalize them $\sum^N_{i=1} \omega^i_t = 1$.

  4. Estimate the new position of the state
     $E[X_t] = \sum^N_{i=1} \omega^i_t \cdot x^i_t$
```

likelihood function that is given directly by the PDI. We choose the most probable location in the image, which are the PDI pixels with the highest values. These locations have been evaluated only using one pixel instead of a region, like in the likelihood function, and therefore they are not reliable as final estimation but provide a good *a priori* initial approximation. The computation of PDI is explained in Section 4.4.

4.3.3 Orientation and Scale Tracking

Particle filter can track simultaneously all the state vector variables. However, some state parameters are often correlated with others. In those cases, the previous estimation of some parameters could facilitate to estimate the other ones. The approach presented in this paper uses this idea to divide the state vector into several steps. Thus, estimating the target location firstly, the algorithm is able to sharpen the rotation and scale changes of the object accurately and with a lower computational load.

This approach is feasible if we assume that changes on scale and orientation are independent of changes on location. Although this is not exactly true because in many occasions they are correlated, it is reasonable if we think how the camera perspective affects to this perception. Thus, a person walking towards the camera does not change its location but its size. However, if the person traces out a diagonal both parameters change simultaneously. Synthetic sequences proved this assertion where a rectangle is rotated, scaled and moved simultaneously, given similar results for partitioned and traditional tracking, as it is shown in Table 4.3.a. This assumption is also confirmed in the test sequences (see Section 4.6).

4.3.3.1 Partitioned sampling

Partitioned sampling is a sampling method developed by Maccormick and Blake [MacCormick and Blake, 2000]. This technique deals with high-dimensional states avoiding an excessive computational cost. Partitioned sampling decomposes the dynamic into two stages, between which an estimation of the first dimensions is calculated. Thus, the sampling methodology reduces the computational load from $O(M^2)$ to nearly to $O(2M)$ mitigating the curse of dimensionality. This methodology has been used to track two independent objects using a unique filter in a more efficient way, but we can generalise it to partition a space of parameter into several stages or subspaces of arbitrary dimensions, if the required conditions are fulfilled.

Assuming the partitioning does not modify the probability $p(x_t|x_{t-1})$, i.e the particle distribution, the dynamic can be decomposed into two steps, as follows

$$p(x_t|x_{t-1}) = \int_{\hat{x}_t} p(x_t|\hat{x}_t)p(\hat{x}_t|x_{t-1})d\hat{x}_t \tag{4.16}$$

where $p(\hat{x}_t)$ are the dynamics in the subspace $\hat{x}_t$ composed of M_1 parameters and similarly for the second subspace of M_2, so that $M = M_1 + M_2$. This assumption would hold if, as it is often the case, the dynamics of the targets were independent of each other. Following the same reasoning, we can generalise the partition for three or more stages.

In this way, a target which deforms and moves in a space of M dimensions can be tracked by performing two inferences, one of M_1 dimensions and another one ofM_2, at each time step. Then we can estimate the two sub-spaces efficiently by first inferring the configuration of parameter subspace $\hat{x}$. This technique is particularly beneficial when the previous knowledge of some dimensions simplifies the search of other dimensions. Examples can be observed tracking several targets, in which it is more efficient to search first for the target that occludes another; or locating a target in the image before obtaining its shape.

To keep the statistical coherence of the particle filter, it is crucial that the partitioning stage does not alter the distribution of the particle set. For this purpose an additional operation on particle sets, called weighted resampling, is required. Although weighted resampling does not modify the distribution, it allows repositioning the locations of the particles for a more efficient representation.

Weighted resampling produces a new particle set by resampling proportionally to the importance function $h_t(x_t^i)$. This importance function should be strictly positive and continuous so that the resampling has no effect the posterior distribution (asymptotically when the number of particles tends to infinity). The underlying idea is to produce a new particle set by resampling, with replacement, from the x_t^i, using probabilities proportional to $h_t(x_t^i)$. That is, many particles are selected in regions where h_t is peaked. Since the resampling can not alter the distribution, the weights of the new resampled particles are calculated in such a way that the overall distribution represented is the same as the old one. Therefore, the weights of the resulting particle set are inversely proportional to the values of the importance function, which have been used to perform the resampling (Fig. 4.14).

$$\hat{\omega}_t^i = \omega_{t-1}^i / h_t(x_t^i) \tag{4.17}$$

Weighted resampling is based on the same principles as, the previously described, importance sampling. Both approaches share similar goals, that is keeping the original distribution, but their methodologies are entirely dissimilar. Importance sampling generates new particles from the proposal distribution function, adding later a correction weight. Nevertheless, partition sampling evolves the new particles from the old particles through the importance function. We can see the similarities and differences between both techniques in equations 4.14 and 4.18.

$$\omega_t^i \propto \hat{\omega}_t^i \cdot p(z_t|x_t^i) = \omega_{t-1}^i \cdot p(z_t|x_t^i)/h_t(\hat{x}_t^i) \tag{4.18}$$

Partitioned sampling algorithm is shown in Algorithm 4.

As importance function we will use the likelihood of the first subspace $p(z_t|\hat{x}_t^i)$, as indicated in next section. Since the goal of partitioned sampling is to obtain an accurate representation of the posterior with a moderate number of particles, we would like the weighted resampling step to position as many particles as possible near peaks in the posterior. Hence we are interested

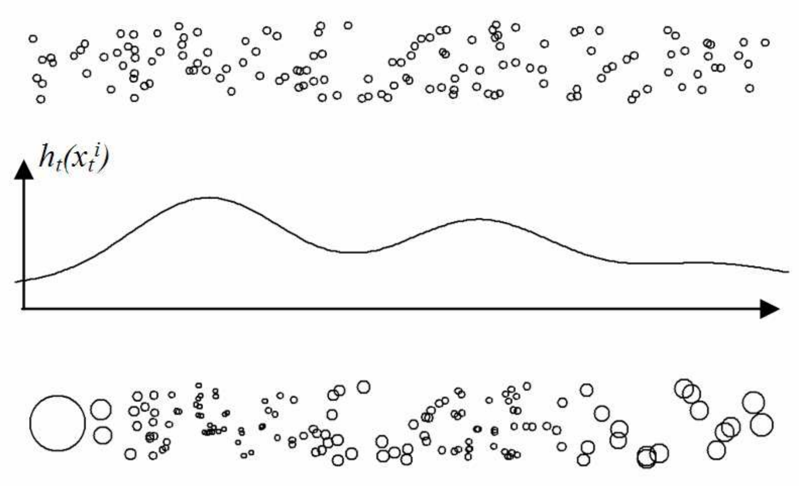


Figure 4.14: Weighted resampling. Top: Particles propagated from the previous time-step importance function. Middle: Importance function $h_t(x_t^i)$. Particles drawn proportionally to the proposal distribution function. Bottom: Particle set resulting of the weighted sampling with the compensation factor included.

Algorithm 4: Partitioned Sampling

```
Given a particle set $\{x_{t-1}^i, \omega_{t-1}^i\}_{i=1}^N$ which represents the posterior probability
$p(x_{t-1}|z_{t-1})$ at time $t-1$

    1. Propagate $N$ samples from the resampled particles in the previous time step applying
       dynamic in the main partitioned dimensions $\hat{x}_t^i \sim p(\hat{x}_t|x_{t-1})$.

    2. Weighted resampling: Particles are resampled using the importance function $h_t(x_t)$, and
       a compensation factor is applied to the weights $\hat{\omega}_t^i = \omega_{t-1}^i / h_t(\hat{x}_t^i)$.

    3. Propagate $N$ samples from the generated samples applying dynamic in the remaining
       dimensions $x_t^i \sim p(x_t|\hat{x}_t)$.

    4. Multiply by likelihood: weight the particles
       $\omega_t^i = \hat{\omega}_t^i \cdot p(z_t|x_t^i) = \omega_{t-1}^i \cdot p(z_t|x_t^i) / h_t(\hat{x}_t^i)$
       and normalize them $\sum_{i=1}^N \omega_t^i = 1$.

    5. Estimate the new position of the state
       $E[X_t] = \sum_{i=1}^N \omega_t^i \cdot x_t^i$
```

on choosing the most similar importance function to the likelihood. Specifically, we use the same likelihood function for both estimations but evaluating different parameters on the same measurement, in particular, only the first M_1 parameters. Particles surviving the weighted resampling step lie in peaks of subspace $\hat{x_t}$, which will be refined by the second estimation.

4.3.3.2 Efficient partitioned sampling PF

The resampling techniques presented previously are complementary. Importance sampling enables using *a priori* estimation of some dimensions, which can be extracted easily, for estimating the remaining ones using partitioned sampling. The position coordinates, which can be easily obtained with regard to the shape parameters, can be mentioned as an example. Once the location of the target is known, extracting its shape is a simpler task. As these dimensions are more evident, estimating them with a basic measurement process as first approximation is a feasible task. Therefore, an importance function $g_t(x_t^i)$ can be obtained to be used as proposal distribution only for these main dimensions. Since partitioned sampling can be repeated several times, the estimation procedure can be decomposed either to be used in a high dimensional space or to further improve the efficiency. The compensation factor, due to the better proposal distribution, is only included in the first stage.

For instance, given a state vector $x_t = [x, y, \dot{x}, \dot{y}, h, w, \dot{h}, \dot{w}, \theta, \dot{\theta}]$, where h and w are the height and the width of the region respectively and θ the orientation, we can define a intermediate estimation $\hat{x_t}$ by partitioning where only the location parameters are estimated, that is, $x_t = [x, y, \dot{x}, \dot{y}]$.

Algorithm 5 describes the complete algorithm combining both a better proposal distribution and partitioned sampling. Note that a percentage of conventional particles N_2 have been added to compensate imperfect proposal distribution, as mentioned in Section 4.3.2. Although the number of particles can be fixed, as in the conventional PF, we have modified it in that way that the number of particles is variable. That is, it can change from one frame to another in order to adapt oneself to the proposal distribution and sampling it until a pre-defined percentage. For that the sampling algorithm has been modified to make possible the increase or decrease of the number of particles

Figure 4.15 shows sequentially the algorithm and the evolution of the particle throughout the sampling stages.

4.4 Observation

All tracking methods require a measurement more or less accurate. The measurement is the way by means of tracking algorithms interpret the reality in order to track the targets.

Although the better measurement, the better tracking results, under no conditions is it possible to obtain a perfect measurement. For that reason, a good agreement between detection an tracking algorithms is always sought. By combining adequately the complexity of measurement and tracking, taking into account the requirements of our particular application, a robust, accurate and effective tracking system will be achieved.

Although we have focused on colour, other techniques to extract features of an object have been tested. These techniques allow us to define, differentiate and identify a target. Since the object can be non-rigid and it can move in 3D being observed by multiple cameras, this characteristics should be invariant to parameters like scale, rotation, viewpoint and so on. Results and conclusions for gradients and corners are presented in Appendixes A and B respectively.

Algorithm 5: Efficient Particle Filter Proposal

```
Given a particle set $\{x_{t-1}^i, \omega_{t-1}^i\}_{i=1}^{N=N_1+N_2}$ which represents the posterior probability $p(x_{t-1}|z_{t-1})$ at time $t-1$
    1. Generate $N_1$ new samples from the importance function in the main partitioned dimensions. $\hat{x}_t^i \sim g(x_t)$.
    2. Propagate $N_2$ samples from the old samples applying dynamic $\hat{x}_t^i \sim p(\hat{x}_t|x_{t-1})$.
    3. Weighted resampling: Particles $N_1$ are resampled using the importance function $h_t^1(x_t) = p(z_t|\hat{x}_t^i) \cdot f_t(\hat{x}_t^i)/g_t(\hat{x}_t^i)$, and a compensation factor is applied to the weights $\hat{\omega}_t^i = \omega_{t-1}^i/h_t^1(\hat{x}_t^i)$.
    4. Weighted resampling: Particles $N_2$ are resampled using the importance function $h_t^2(x_t) = p(z_t|\hat{x}_t^i)$, and a compensation factor is applied to the weights $\hat{\omega}_t^i = \omega_{t-1}^i/h_t^2(\hat{x}_t^i)$.
    5. Propagate $N = N_1 + N_2$ samples from the generated samples applying dynamic in the remaining dimensions $x_t^i \sim p(x_t|\hat{x}_t)$.
    6. Multiply by likelihood: weigh the particles
       $\omega_t^i = \hat{\omega}_t^i \cdot p(z_t|x_t^i) = \omega_{t-1}^i \cdot p(z_t|x_t^i)/h_t(\hat{x}_t^i)$
       $= \begin{cases} \omega_{t-1}^i \cdot \frac{p(z_t|x_t^i) \cdot g_t(\hat{x}_t^i)}{p(z_t|\hat{x}_t^i) \cdot f_t(\hat{x}_t^i)} & i = 1, ..., N_1 \\ \omega_{t-1}^i \cdot \frac{p(z_t|x_t^i)}{p(z_t|\hat{x}_t^i)} & i = N_1 + 1, ..., N \end{cases}$
       and normalize them $\sum_{i=1}^N \omega_t^i = 1$.
    7. Estimate the new position of the state
       $E[X_t] = \sum_{i=1}^N \omega_t^i \cdot x_t^i$
```

4.4.1 Colour Density Estimation

The observation process is the stage in which the likelihood of each particle is evaluated and its corresponding weight is assigned. This task is usually achieved by comparison to an observation model. In this work it is considered how to use the efficient particle filter described before with colour-based image features. Thus, the tracked object appearance is modelled by sampled pixels which represent its colour distribution.

Following the diagram shows in Fig. 4.12, the input image I_t, at instant t, is transformed into a Probability colour Density Image (*PDI*). Each pixel of this image points its membership to the target colour model.

Ideally, the value of each pixel in the PDI should measure the membership of all the possible hypotheses for the whole range of parameters (scale, orientation, etc.) in that location. This is the procedure that the traditional colour-based particle filter follows, but of course without evaluating all the pixels but only the hypotheses that the filter launches. However this methodology is unapproachable and collides with our objective to speed up the estimation of the PF and the weighting of the particles. Instead, we propose that each pixel measures only its own membership to the model. The membership of the whole target in a certain location will be estimated by adding the nearby pixels that the parameters of the hypothesis delimit. The validity of this proposal will be shown in the result section.

The PDI is used in two ways: to generate the proposal distribution which introduces information about the current observation (prior probability), and to compute the likelihood of

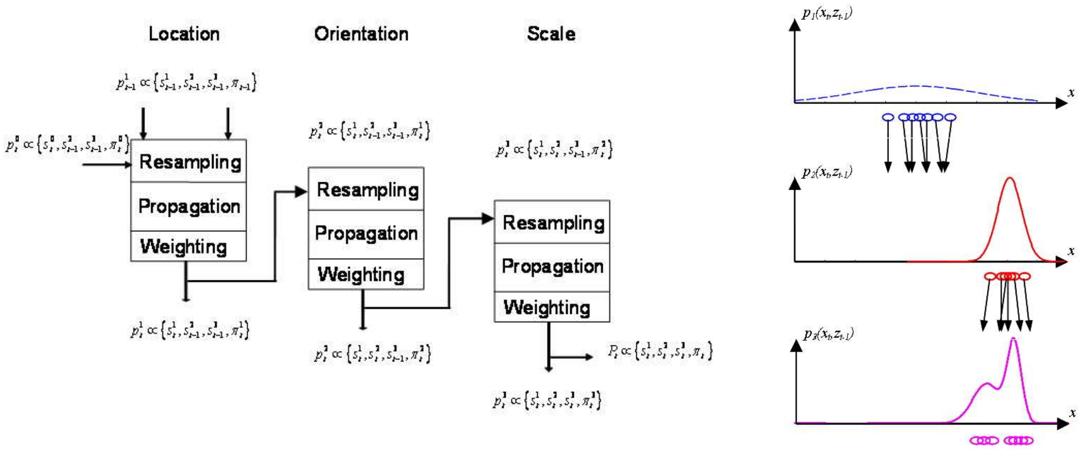

Location
Orientation
Scale
p^1_{t-1} ∝ {s^1_{t-1}, s^2_{t-1}, s^3_{t-1}, π_{t-1}}
p^0_t ∝ {s^0_t, s^2_{t-1}, s^3_{t-1}, π^0_t}
Resampling
Propagation
Weighting
p^1_t ∝ {s^1_t, s^2_{t-1}, s^3_{t-1}, π^1_t}
p^2_t ∝ {s^1_t, s^2_{t-1}, s^3_{t-1}, π^1_t}
p^2_t ∝ {s^1_t, s^2_t, s^3_{t-1}, π^2_t}
p^3_t ∝ {s^1_t, s^2_t, s^3_{t-1}, π^2_t}
P_t ∝ {s^1_t, s^2_t, s^3_t, π_t}
p^3_t ∝ {s^1_t, s^2_t, s^3_t, π^3_t}
p1(xt,zt-1)
p2(xt,zt-1)
p3(xt,zt-1)
x

Figure 4.15: An example of partitioned sampling algorithm in which location, scale and orientation are estimated consecutively. a) Algorithm, b) Sampling of particles.

each particle (posterior probability).

A priori: Once the PDI is calculated, the particles are distributed in those parts of the image with high PDI value. This is achieved following the proposal distribution function $g_t(x_t^i)$ described earlier.

A posteriori: The likelihood $p(z|x)$ of a certain observation z given a hypothesis x of the state (position, size, etc.) involves the integration over a set of observation points (PDI pixels). By calculating in advance the integral image of the likelihood image, it becomes in a simple combination of sums, reducing significatively the time to evaluate all particles.

Colour density functions can be modelled by parametric and non-parametric methods.

4.4.1.1 Parametric PDI

The simplest technique for colour density estimation consists in assuming the target as a one-colour region, [McKenna et al., 1999], and modelling it as a Gaussian distribution, $p_{target} \sim N(\mu, \sigma)$. Although it is a limited assumption, it can be easily extended dividing the target in a set of fixed one-colour regions. Following this parametric approach, the colour density estimation over an image I can be calculated as

$$PDI(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(I(\mathbf{x})-\mu)^2/2\sigma^2} \tag{4.19}$$

where we denoted as PDI to the resulting Probability colour Density Image and $I(x)$ is the colour of the pixel at location x in the input image.

A more sophisticated technique is to generate a model of the target using gaussian mixture models. Depending the number of Gaussians M chosen, we must estimate $3 \cdot M$ parameters

$$PDI(x) = \sum_{i=1}^{M} \alpha_i \frac{1}{\sqrt{2\pi\sigma_i}} e^{-(I(x)-\mu_i)^2/2\sigma_i^2} \tag{4.20}$$

being α_i the coefficient which multiply each Gaussian to generate the specific mixture.

Both previous techniques are based on parametric methodologies. However these techniques exhibit some problems: they require a detailed prior knowledge, like the selection of the number

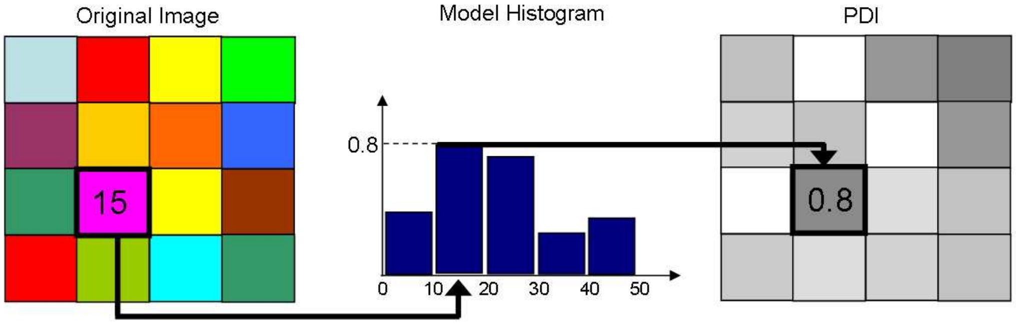


Figure 4.16: Histogram used as look up table to generate the PDI

of Gaussians to be fitted, and are not adequate for all kind of targets, specially in video-surveillance systems. Therefore, non-parametric models have been chosen in this chapter to generate the PDI, although parametric models can be applied for specific application, as we will show in the next chapter.

4.4.1.2 Non-parametric PDI

The major advantage of non-parametric approaches is the flexibility to represent complicated densities effectively since they do not assume any specific shape for the density function. Histogram is the simplest non-parametric density estimator. A classical and widespread approach uses colour histogram as model and Bhattacharyya distance as similarity measurement [Nummiaro et al., 2003; Pérez et al., 2002; Bohyung Han, 2004; Vermaak et al., 2003a; Comaniciu et al., 2003]. However, we should realise that this approach exhibits some drawbacks: spatial layout is lost, cluttered backgrounds can confuse the tracker and the computational cost may become excessive for large regions or high colour resolutions.

This segmentation is made detecting groups of colours over the sub-sampled colour histograms. It is worthy to note that the sub-sampling of the colour resolution is a important process to obtain general results. A very low level of sub-sampling (using many bits to represent each colour channel) will produce low values in the comparison, even between very similar and correct regions. On the other hand, a high level of subsampling will accept erroneous matching with regions considerably different.

We propose two methodologies to obtain valid PDI from an histogram or any other distribution function obtained by non-parametric approaches.

Look-up Table The new input frame is projected onto the target probability space to generate the a priori Probability Colour Density Image (PDI). The values of the PDI pixels are taken from the probability density histogram of the target, using it as a look-up table for the corresponding pixels in the input picture. A visual scheme of this procedure is depicted in Figure 4.16.

Punctual Histogram Distance This method employs a metric to determinate the conditional probability for a pixel of belonging to the target object. In the case of grey level images, we have defined two metrics, which give similar results:

$$PDI(x,y) = max_u \left\{ \frac{1}{1+|I(x,y)-u|} \right\} \tag{4.21}$$

$$PDI(x,y) = e^{-min_u\{|I(x,y)-u|\}} \tag{4.22}$$

where u is the grey level of the target histogram with any positive value. This probability gives a value of 1 when the grey level of the pixel corresponds to someone grey level in the target histogram and gives some positive value. The more different the pixel is to any value of the histogram, the lower the probability is.

For colour tracking, metrics are also valid for colour spaces like RGB. However, in circular colour spaced like HSV, it has to be taken into account that the largest and smallest values are almost equal. For instance, a model based on "hue" histogram we have to pay attention because the hue channel is an angular representation. So, the distance between two hue values h_1 and h_2 is given by

$$d_{hue}(h_1, h_2) = \sin\frac{h_1 - h_2}{2} \tag{4.23}$$

In this sense, the probability density image is given by:

$$PDI(x,y) = e^{-min_u\{|d_{hue}(h(x,y),u)|\}} = e^{-min_u\left\{\left|\sin\frac{h(x,y)-u}{2}\right|\right\}} \tag{4.24}$$

4.4.2 Prior Probability

In this section, we propose different ways to include a priori information in the condensation algorithm. This a priori measurement in the colour space is based on the PDI and reduces the costs of evaluating hypotheses with a very low likelihood.

As it is shown in [Comaniciu et al., 2003], these hypotheses (or particles) appear due to poor prior density estimation, particularly when the object motion between frames is badly modelled by the dynamic model. The improvement of the statistical efficiency of the sampling allows a reduction in the number of particles substantially.

Colour-based image features are used as proposal distribution. This importance function $g(x_t^i)$ will be introduced in the algorithm like a mask. The mask represents the areas of the image where it is recommendable to launch particles because we have some sings of the target. It can be binary, in whose case the values of $g(x_t^i)$ will be 0 and/or 1, or be weighted by the degree of membership to the model, in whose case the values of $g(x_t^i)$ will be given by the PDI function.

The mask is obtained by extracting the main colours of the object and detecting them in the full image or in the zone surrounding the estimate position (extracted with the last mean state) whose dimensions depend on the position and speed variances. Only hypotheses located into this mask are evaluated and used in the estimation. Moreover, we also apply this information to obtain a fast estimation of the posterior density.

4.4.2.1 Parametric mask

Assuming a Gaussian model, we can apply the chi squared test to create a mask only using those pixels corresponding to the target's Gaussian. If the model is composed of a mixture of Gaussians, a test for each Gaussian should be applied.

Chi squared value is calculated as

$$\chi^2 = \frac{(x-\mu)^2}{\sigma^2} \tag{4.25}$$

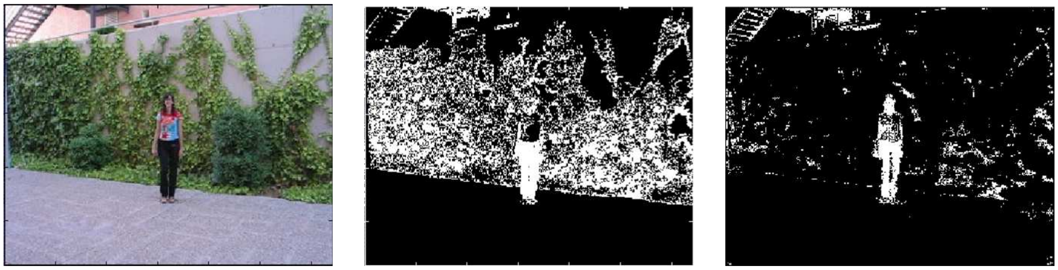


Figure 4.17: Mask obtained with sub-sampling method: a) Original image, b) Measure mask, c) Mask without background colours

where x is the observation, μ is the mean or expectation and σ is the standard deviation. By comparing the chi-squared value with the value of a Normal function, we can adjust the mask to contain a fix percentage of the possibilities. The value changes in accordance with the degree of freedom of our variable. For instance, to take into account the 95% of the possibilities, the chi squared value is 3.84 for one dimension or 5.99 for a bidimensional gaussian.

Finally, if the model also contains the background, like in Section 4.2.3.1, the mask can be created by comparison with the most probable clusters. Most segmentation techniques generates a hard clustering besides a soft one, which can be used as mask. In these cases, the image is clusterised by taking the nearest Gaussian corresponding and checking its correspondence to the target.

4.4.2.2 Histogram sub-sampling mask

The mask can be obtained from the PDI by choosing the maximum peak of the model, all the peaks whose value is larger than a percentage of the maximum peak, or simply all the values larger than a minimum threshold. Results of these three alternatives do not differ too much.

However, the colours of the object can be similar to those existing in the background. If this occurs, the a priori probability image may be saturated of false points, and therefore the measure attractor will be useless in the best case, and will damage the tracking in the worst one. We avoid this effect with a logical methodology: if some colour group of the object is equal to one of the background colours, we do not use it to build the mask, although it will be the colour with the greatest value in the histogram. Instead of that, we select colours in which the object and the background differ. The improvement in the extraction of the mask is shown in the figure 4.17.

4.4.2.3 Chi squared mask

Another possibility consists in considering the approximation of the histogram by Gaussian functions, applying techniques like KDE. So, we can use the chi squared test for accepting or rejecting the hypothesis whether or not an image pixel belongs to the target histogram.

4.4.2.4 Density mask

The mask is obtained filtering the PDI with a Laplacian kernel with the same size as the target object, and thresholding by the half value of the estimated density. An example is shown in Figure 4.18

The three different techniques were calculated and tested as a priori likelihood. Some conclusions could be extracted:

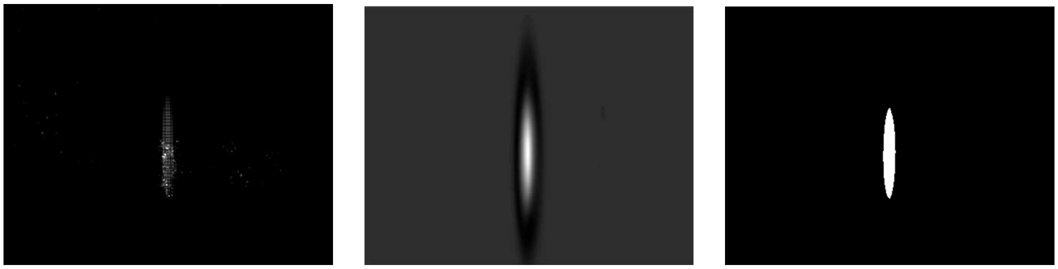


Figure 4.18: PDI and mask obtained using density method on the figure 4.17 image: a) Prior density, b) Filtered PDI, c) Mask.

- The histogram sub-sampling mask increases the efficiency of the system minimising the number of useless particles. It gives the best results in colour sequences.
- The density mask provides a more reliable mask because it includes size information. Therefore, it needs less particles than the previous method. Nevertheless, the Laplacian filter requires a very high computational cost, which only will be make up for a lower number of particles in very complex sequences. In cases with many simultaneous targets, it can be used to determine the number of targets and distracters which appear in the scene.
- The chi squared mask gives good results with grey level images or targets composed of flat regions, but in colour sequences with multi-colour targets it introduces an error when we suppose one gaussian histograms. To improve it a Gaussian Mixture Model should be applied.

4.4.3 Colour likelihood by Integral Image

The colour likelihood function $p(z|x)$ is based on target colour distribution p_{target} and a background colour distribution p_{bkdg} [Stenger et al., 2006; Sudderth et al., 2004]. Given a state vector x, corresponding to a particular target position, the region encompass by the target is defined as $S(x)$. All the pixels in the image can be partitioned into two sets corresponding to the inside $\{v : v \epsilon S(x)\}$ or outside $\{u : u \epsilon \bar{S}(x)\}$ of the target. Assuming pixel-wise independence, the likelihood function of an image is

$$p(z|x) = \prod_{v \epsilon S(x)} p_{target}(I(v)) \prod_{u \epsilon \bar{S}(x)} p_{bkgd}(I(u)) \propto \prod_{v \epsilon S(x)} \frac{p_{target}(I(v))}{p_{bkgd}(I(v))} \tag{4.26}$$

where $I(v)$ is the colour vector at location v in the image.

When taking the logarithm,

$$log(p(z|x)) \propto \sum_{v \epsilon S(x)} \big(log(p_{target}(I(v))) - log(p_{bkgd}(I(v)))\big) \tag{4.27}$$

this term is converted into a sum which can be computed efficiently as a sum table or integral image.

In general, the background colour distribution is not given in advance or it is constantly changing. In lack of that model, the absence of foreground distribution can be rewarded. For instance, to deal with objects in clutter environment, some authors [Collins, 2003] proposed to

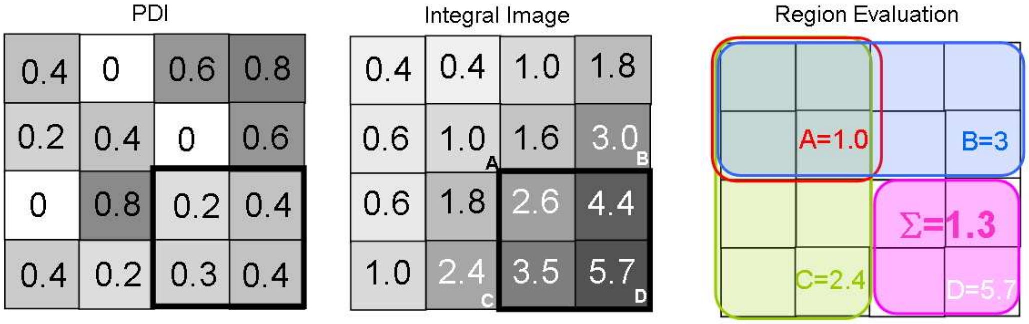


Figure 4.19: Integral image example: a)PDI and rectangle to be evaluated (black). b)Integral image obtained from the PDI. c) Physical interpretation of the integral image and resulting value of the rectangle using eq. (4.30).

use a Laplacian filter. The negative ring surrounding the kernel measures the non-membership grade around the target. This modification increases the accuracy of the estimation as well as facilitating the tracking of size changes in the target.

$$log(p(z|x)) \propto \sum_{v \epsilon S(\mathbf{x})} log(p_{target}(I(v))) - \sum_{v \epsilon R(x)} log(p_{target}(I(v))) \tag{4.28}$$

where $R(\mathbf{x})$ represents the ring surrounding the target.

Equation (4.28) is expressed in a very convenient way. When an exhaustive search must be done or many hypotheses should be evaluated, it is quite common that those hypotheses overlap themselves and therefore the same region is evaluated several times. However, it is possible to calculate the sum of the values within rectangular regions in linear time without repeating the summation operator for each possible region. Instead of evaluating each hypothesis until finding the optimum, we can pre-calculate a cumulative image that gives us the value of every hypothesis in every possible location.

This cumulative image function, called integral image, is defined such that each element of this function holds the sum of all values to the left and above of the pixel including the value of the pixel itself. The cumulative image can be computed for all pixels with four arithmetic operations per pixel. Starting from the top left corner and traversing first to the right and then to the down, the value of the cumulative image at the current pixel is obtained by the addition of the left and the up pixel and subtraction of the upper left pixel's cumulative values.

$$I\_Intg(y,x) = I\_Intg(y,x-1) + I\_Intg(y-1,x) - I\_Intg(y-1,x-1) + I(y,x) \tag{4.29}$$

where I is the original image and I_{Intg} is the integral image.

After the cumulative image is obtained, the sum of image function in a rectangle can be computed with another four arithmetic operations with appropriate modifications at the border. Thus the value of the rectangular region is given by the addition of the upper left vertex A and the lower-right vertex D and subtraction of the lower left and upper right pixel's cumulative values C and D respectively. An example is shown in Figure 4.19.

$$\sum(S(x)) = A + D - B - C \tag{4.30}$$

The integral image has been used to evaluate a large number of rectangular masks on a gray scale image. By converting the probability density image to the integral image, where each

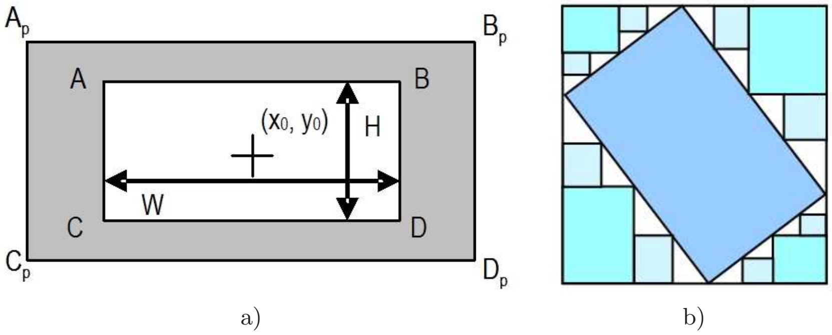


Figure 4.20: (a)Laplacian rectangular kernel. W and H stand for particle Width and Height. The gray area represents the negative values of the matrix. (b)Rotated mask modelled by a set of 13 rectangles

pixel is the sum of all pixels above and to left of its current position, the estimation of each rectangular mask (i.e. the probability of this region) can be computed in four array reference. The Laplacian filter is approximated by a rectangular mask centered at (x_0, y_0), with width W and height H, surrounding by a negative ring mask, as depicted in Fig. 4.20.a. The evaluation can now be performed efficiently in linear time by computing the integral image.

In [B.Han et al., 2005], authors describe a particle filter whose observation process combines a measurement of the similarity between color regions with an edge orientation histogram, in order to assign weights to the particles. Each color channel and feasible gradient orientation produce an integral image. Then, likelihood functions (in this case, Euclidean distance) measure the similarity between target model and the value computed from the integral images. The proposed observation process is based on a similar methodology. However, while [B.Han et al., 2005] generates one integral image for each observed feature, the proposed technique summarises all the integral images into a unique representation. Integral image, traditionally a feature counter, becomes a membership counter by being generated from a PDI. Summarising all features in an image emphasises the advantages of the integral image, decreasing the fixed cost of calculating the integral images and the number of lookup access for each particle.

Thus, the estimation of a rectangular mask is computed as the sum of the vertice values $(A+D-B-C)$. Taking into account that we should subtract the laplacian ring, the likelihood is formulated as follows

$$\begin{aligned} p(z|x) &\propto (A+D-B-C) - ((A_p+D_p-B_p-C_p) - (A+D-B-C)) \\ &= 2(A+D-B-C) - (A_p+D_p-B_p-C_p) \end{aligned} \tag{4.31}$$

where A,B,C,D,A_p,B_p,C_p and D_p are the vertices of the mask and its laplacian ring indicated in Figure 4.20.a. The laplacian ring is calculated as the external rectangular mask minus the inner rectangle.

4.4.3.1 Rotated rectangular mask

In Section 4.4.3 a method to evaluate the likelihood of each particles is described. Nevertheless, this fast technique is only valid for targets which can be modelled by rectangles. One of the major drawbacks of the integral image consists in that only rectangular masks can be evaluated. This requirement limits the possibilities of feature based tracking. For instance, a property so important as orientation changes can not be tracked. Therefore, the partitioned sampling can only be applied to track location and shape parameters, never for angular parameters. In order to track angular movements and simultaneously enjoy the advantages that the integral image advantages entails, a novel solution is proposed.

In the event that rotated rectangles or even more complex shapes need to be evaluated, a simple solution involves decomposing the shape into the smaller rectangles which it contains. The content of the rotated mask is computed as the difference between the elements of the bounding box and the contents of those rectangles confined by the triangles (Fig. 4.20.b). The larger is the number of rectangles, the lower is the error of the covered area. The importance of this intrinsic error obtained depends on the number of rectangles used to rebuild the original shape. Although an infinite number of rectangles are needed, Table 4.3.b shows that the use of a limited number of rectangles can approximate an adequate result and the rotation angle is correctly estimated even with few rectangles (although few rectangles produce a lower resolution in the angle estimation).

Table 4.3: (a)Comparison between partitioned and traditional rotated rectangle estimation. (b) Error vs number of rectangles.

a)

	Trad	Part
MSE x	0.48	0.58
MSE y	0.73	0.57
MSE ang.	2.47	3.52
Time[s]	459.1	4.92

b)

Rectan.	Area Er.	Ang. Er.
5	25%	11.97^o
13	12.5%	8.96^o
33	6.25%	5.54^o
61	3.125%	3.79^o

Table 4.3.a shows the differences in time (seconds) and in error (Mean Square Error) between estimating location and angle simultaneously (traditional), or extracting a first location estimation to help the final one (partitioned). A similar error is obtained with a much lower time cost because of the lower number of particles.

4.4.3.2 Multiple region conjugation

If our model is composed of several regions, the conjugation of the probabilities of each region is done assuming independence between all of them. Thus, if the state is $x_t = \{x_t^{r_1}, x_t^{r_2}, ..., x_t^{r_N}\}$ where N is the number of region r_n, the probability is

$$p(z_t|x_t) = p(z_t^{r_1}|x_t^{r_1}) \cdot p(z_t^{r_2}|x_t^{r_2}) \cdot \quad \cdots \quad \cdot p(z_t^{r_N}|x_t^{r_N}) \tag{4.32}$$

The individual probability of each region $p(z_t^{rn}|x_t^{rn})$ is calculated as described in Section 4.4.3.

4.4.4 Integral Histograms

Another approach, called integral histograms, presented in [Porikli, 2005] enables an exhaustive search using histogram comparison. Integral histogram is a recursive propagation method to store histograms which, once constructed, allows for the computation of histogram within a rectangular area. Specifically, it is a superset of cumulative images, as many as histogram bins considered, which are the accumulation of pixel in the original image that are classified in that bin. To perform histogram comparison using this methodology, an integral histogram is generated first by propagation, and then the histograms of target regions is computed by intersection.

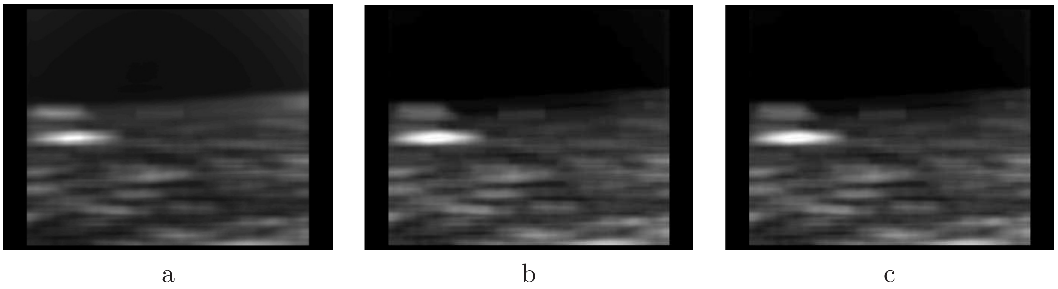


Figure 4.21: Results obtained comparing our method with Integral Histograms: a) Our method, b) Integral histograms using minimum distance, c) Integral histograms using Bhattacharyya distance.

The propagation step allows generating the cumulative images sequentially so that

$$\begin{array}{rcl} H(x,y,b) = H(x-1,y,b) + H(x,y-1,b) - H(x-1,y-1,b) & \forall b \in Bins \\ H(x,y,b) = H(x,y,b) + 1 & if I(x,y) == b \end{array} \tag{4.33}$$

where $Bins$ is the number of bins considered in the histogram representation and I is the original image.

The histogram of a target region $S(x)$ can be computed using the integral histogram values at the boundary points $[(x^-,y^-),(x^+,y^-),(x^-,y^+),(x^+,y^+)]$ of the region. This intersection is based on simple arithmetic operations, in a similar way that equation (4.31).

$$h(S(x),b) = H(x^+,y^+,b) - H(x^-,y^+,b) - H(x^+,y^-,b) + H(x^-,y^-,b) \quad \forall b \in Bins \tag{4.34}$$

However, the computational advantages are attenuated by this methodology, since it involves generating as many integral image as histogram bins considered. Furthermore, the larger the number of integral images is, the greater the computational cost of evaluating each particle is. Unlike integral histograms, our proposal needs to compute a unique integral image. On the other hand, integral histogram method enables more accurate hypothesis evaluation in the event of complex regions and highly detailed histograms. This is due to the fact that integral histograms perform a real histogram comparison for each hypothesis (using the Bhattacharyya distance, for instance). However, a detailed histogram requires many bins to be modelled and therefore a large initial computational load. The accuracy of our proposal can be improved in these situations splitting the target in several regions, and combining the evaluation of each region in the final estimation.

For these reasons, integral histogram methodology is only suitable for extremely exhaustive hypothesis evaluation, like object detection and maximum search, in which a high initial computational cost is made up for evaluating a large set of hypothesis with a lower cost.

Figure 4.21 and Table 4.4 shows the comparison of our method, in time and in probability, against the integral histograms. Whereas the probability results are practically identical, as shown in 4.21, our methods means a substantial decrease in time. Results has been calculated assuming an hypothesis for every pixel of the probability image. An extra comparison between both methods is shown in Section 4.6.

4.5 Colour Appearance Model Update

Although colour-based particle filters have been used exhaustively in the literature, given rise to multiple applications, tracking coloured objects over time has an important drawback, since

Table 4.4: Results obtained comparing our method with Integral Histograms.

	Our	Integral histograms using minimum distance	Integral histograms using Bhattacharyya distance
Time [sec]	3.040398	122.058649	251.561755

the way in which the camera perceives the object can change. This is specially required in surveillance applications, in which the environmental conditions are not controlled and susceptible to change due to external factors like clouds, weather conditions and so on.

Both parametric and non-parametric methods require an observation model to make the comparison. Although the multiple hypotheses provided by particle filters permits us to deal with occlusions and multiple targets, the validity of each hypothesis is based on the comparison with a predefined model. Due to the fact that the appearance of an object changes with the time, a fixed model could not be very useful. The observation model could be completely recalculated adding the new samples to the previous ones. However this option is unfeasible due to the computational cost of generating a new model. For this reason, online updates are employed

Simple updates are often used to address this problem, which imply a risk of distorting the model and losing the target. Several solutions have been proposed [Vermaak et al., 2002; Nummiaro et al., 2003; McKenna et al., 1999; Sigal et al., 2000]. In [Pérez et al., 2002; Okuma et al., 2003; Comaniciu et al., 2003; Sigal et al., 2000], authors propose a popular approach for non parametric methods. At the end of each filter iteration, the estimation is used to update the model, using a parameter to control the update velocity.

$$hist[t+1] = (1-\alpha)\cdot hist[t] + \alpha \cdot hist_{E[x_t]}[t] \tag{4.35}$$

where parameter α controls the updating velocity.

A more sophisticated version of the previous non-parametric updating scheme is the one presented by Li and Chua [Li and Chua, 2007]. They propose an adaptive particle filter based on transductive inference, which tries to induce the decision of minimum error on the whole particle distribution.

$$hist[t+1] = (1-\alpha-\beta)\cdot hist[t] + \alpha \cdot hist_{E[x_t]}[t] + \beta \sum_{j=1\ldots N_s} \omega_t^j hist_{x_t^j}[t] \tag{4.36}$$

where α and β are scalars between 0 and 1 that allow us to adjust the speed of adaptation. Weights ω_t^j express the classification confidence and are calculated on the basis of the particle weights assigned by this weak classifier $hist_{x_t^j}[t]$

$$\omega_t^j = \omega_{t-1}^j \frac{p(z_k|x_k^j)\cdot p(x_k^j|x_{k-1}^j)}{q(x_k^j|x_{k-1}^j, z_k)} \tag{4.37}$$

However, this approach has a similar drawback to the previous technique. If the object model is adapted too readily, the object will encounter partial occlusions, outlier clutter or some transient tracking failures. Otherwise, the updating process will not be fast enough to respond to the illumination changes. Therefore, it is easy to adapt erroneously to some part background in the above situations. Ad-hoc techniques have been used to mitigate this undesirable effects, such as an adaptation threshold or updating the object colour model when the observation probability is above a pre-defined threshold.

The same updating philosophy can be applied to parametric approaches, where model parameters means μ, variances σ and weights w (this last one only for GMM) can be updated in order to compensate illumination changes, using an on-line updating algorithm. Considering GMM as the general case, the parameters of each Gaussian (mean μ, variance σ and weight$w$) are updated as

$$\hat{w}^i = w^i + s \cdot (1 - w^i) \tag{4.38}$$

$$\hat{\mu}^i = \mu^i + s\frac{1}{M_i}\sum_{m=1}^{M_i}\left(\frac{x_m^i}{\hat{w}^i} - \mu^i\right) \tag{4.39}$$

$$\hat{\sigma}^i = \sigma^i + s\frac{1}{M_i}\sum_{m=1}^{M_i}\left(\left(\frac{x_m^i}{\hat{w}^i} - \mu^i\right)\cdot\left(\frac{x_m^i}{\hat{w}^i} - \mu^i\right) - \sigma^i\right) \tag{4.40}$$

where s is the updating constant which adjusts the updating speed and x_m^i are the pixels used to the updating process.

However this kind of techniques has an important drawback: if the estimation is not accurate enough, the model could degenerate in few frames, because of the feedback between the tracking and the model errors. Moreover, the updating speed is usually chosen empirically. In [Li and Zheng, 2004], an interesting technique to adjust automatically this speed during the tracking is shown. Other algorithms try to compensate the changes in the image before applying the tracking algorithm, such as gray level [Cho et al., 2002] or perfect reflector algorithm, which introduce a gamma correction. Although they are able to correct soft changes in the illumination conditions, they fail against strong ones (Fig. 4.26) and are not useful to assimilate changes in the colour or appearance of the object.

In this section we propose a new updating strategy, which improves substantially the efficiency of the updating process. A joint image-characteristic space tracking updates the model simultaneously to the object location. By including the model parameters in the state vector, the probability of losing the target corrupting the model is reduced. Furthermore, a joint tracking helps to track the characteristic-space parameters using the image-space parameters: inconsistent movements in the image are a clue to detect bad estimations.

Nevertheless, the inclusion of the colour model in the state can increase substantially the complexity of the estimation just as the computational cost. To avoid this undesirable effect, we split the state into two parts: the image-space parameters which are estimated stochastically and the characteristic-space parameters which are estimated analytically. Both parts are jointly tracked using a Rao-Blackwellised particle filter. In [Khan et al., 2004], authors model target's appearance using components in a PCA space. However, the usage of PCA limits the possible appearance changes. In this sense, this proposal is more general and allows tracking whatever appearance model is. Thus the colour model can change drastically and the algorithm still works. Using this technique, the hypotheses are evaluated depending on the difference between the model and the current target appearance during the updating stage. Convincing results have been obtained in sequences under both sudden and gradual illumination condition changes.

4.5.1 Colour-based Rao-Blackwellised Particle Filter

In typical tracking applications, the state of the target is modelled by position $\{x, y\}$, scale s and orientation θ, as well as the temporal change of these variables. When modelling the target colour appearance, the state space is augmented with appearance coefficients such that the state $X_t = (l_t, a_t)$, where l_t models the position of the target and a_t the colour appearance. Target tracking can be expressed as a Bayes filter, in which the posterior distribution is recursively updated $P(X_t|Z^t)$ over the target X_t given all observations $Z^t = \{z_1 \ldots z_t\}$ according to:

$$P(l_t, a_t|Z^t) \propto P(Z^t|l_t, a_t)\int_{l_{t-1}}\int_{a_{t-1}} P(l_t, a_t|l_{t-1}, a_{t-1})P(l_{t-1}, a_{t-1}|Z^{t-1}) \tag{4.41}$$

Rao-Blackwellisation permits to integrate out part of the state to be calculated analytically (in our case, the colour appearance part a_t), obtaining a marginal filter for the location l:

$$P(l_t|Z^t) \propto \int_{a_t} P(Z^t|l_t,a_t) \int_{l_{t-1}} \int_{a_{t-1}} P(l_t,a_t|l_{t-1},a_{t-1}) P(l_{t-1},a_{t-1}|Z^{t-1}) \tag{4.42}$$

Therefore, fewer samples will be needed for the same level of performance, since part of the posterior estimation over the state is calculated exactly instead of approximately by a more expensive sample particle set.

Using Rao-Blackwellised particle filter, we can approximate the non-linear, non-Gaussian target location using particle filter, while we apply Kalman filter to estimate the colour appearance conditioned on that estimated location state, and modelled by a Normal distribution $N(a_t;\hat{a}_t,P_t)$. So, we estimate the posterior $P(l_{t-1},a_{t-1}|Z^{t-1})$ over the previous joint state by a set of particles $\{l_{t-1}^{(i)},\omega_{t-1}^{(i)},\hat{a}_{t-1}^{(i)},P_{t-1}^{(i)}\}_{i=1}^{N}$. We can now follow the Khan's approximation [Khan et al., 2004], obtaining:

$$P(l_t|Z^t) \propto \sum_i \omega_{t-1}^{(i)} P(l_t|l_{t-1}^{(i)}) \cdot \int_{a_t} P(Z^t|l_t,a_t) \int_{a_{t-1}} P(a_t|l_t,l_{t-1}^{(i)},a_{t-1}) P(a_{t-1}|l_{t-1}^{(i)},Z^{t-1}) \tag{4.43}$$

where the second integral gives $P(a_t|l_{t-1}^{(i)},Z^{t-1})$ defined as the density on the current colour appearance coefficients a_t conditioned on particle i and the measurements Z^{t-1} and calculated by a Kalman filter considering a Gaussian approximation.

4.5.1.1 Calculating the importance weights

To calculate the importance weight $\omega_t^{(j)}$ we have:

$$\omega_t^{(j)} = \int_{a_t} P(z_t|\hat{l}_t^{(j)},a_t) \int_{a_{t-1}} P(a_t|\hat{l}_t^{(j)},l_t^{(i)},a_t) P(a_{t-1}|l_{t-1}^{(i)},Z^{t-1}) \tag{4.44}$$

As stated before, we have considered that the density $P(a_{t-1}|l_{t-1}^{(i)},Z^{t-1})$ on the previous colour appearance coefficients a_{t-1} is a normal density $N(a_{t-1};\hat{a}_{t-1},P_{t-1})$. On the other hand, we assume that the colour coefficients a_t change smoothly over time according to a Gaussian process:

$$P(a_t|\hat{l}_t^{(j)},l_{t-1}^{(i)},a_{t-1}) \sim N(\hat{a}_{t-1},Q_{ij}) \tag{4.45}$$

The diagonal variance Qij can be adapted to the distance between the location $\hat{l}_t^{(j)}$ and $l_{t-1}^{(i)}$. In this manner, it is assumed that the target colour appearance tends to change more when the target moves quickly. Using the following equation, the uncertainty is supposed 0 when the target is completely static and it grows in accordance with the velocity of the target.

$$Q_{ij} = q \cdot \Big(1 - \exp \|\hat{l}_t^{(j)} - l_{t-1}^{(i)}\|\Big) \tag{4.46}$$

where q is a tune parameter.

Using Gaussian assumptions and under the inductive assumption that the density on the previous appearance coefficients a_{t-1} is a normal density $N(a_{t-1}|\hat{a}_{t-1},P_{t-1})$, the result of the integral over the previous appearance coefficients in (4.44) is a Gaussian $N(a_{t-1}|\hat{a}_{t-1},P_{t-1}+Qij)$.

Therefore, the importance weight $\omega_t^{(j)}$ takes the expression:

$$\omega_t^{(j)} = \int_{a_t} P(z_t|\hat{l}_t^{(j)}, a_t) N(a_{t-1}|\hat{a}_{t-1}, P_{t-1} + Qij) \tag{4.47}$$

In Section 4.5.3, different solutions for this equation are shown, depending on the chosen colour model.

4.5.1.2 PDA Kalman filter

As we have mentioned previously, appearance parameters are updated using Kalman filter. Thus, a soft transition between the learned model and the new measurements is obtained. Traditional Kalman filter can be applied for this purpose, but it is not the most adequate for colour update tasks. Conventional KF associates one measurement to one hypothesis at each time step. However, in the updating process, it is necessary to update appearance models using a region of interest composed of many pixels, that is, many measurements should be assigned to the same hypothesis. Furthermore, some of these measurements can be more relevant than another like, for example, when compression distorts the pixel colour near the region border. In contrast, Probabilistic Data Association filter (PDAF) [Bar-Shalom and Li, 1993] enables us to combine several valid measurements with a unique tracker at each time step. The state estimation scheme is almost as simple as KF, but much more effective in clutter. Therefore, we propose the inclusion of PDA to allow an effective updating of the appearance when the size of the target changes over time and the number of pixels corresponding to each appearance model parameter varies.

In our approach we assume a smooth change of the colour coefficients a_t over time. So, a dynamic model of zero-order is applied and the state transition function F and the measurement function H are diagonal identity matrices I_d. With these premises, the prediction of the colour coefficients is as follows:

$$\hat{a}_{t|t-1} = F\hat{a}_{t-1} + \varpi_t \tag{4.48}$$

$$P_{t|t-1} = FP_{t-1}F' + Q_t \tag{4.49}$$

$$z_{t|t-1} = H\hat{a}_{t|t-1} + \upsilon_t \tag{4.50}$$

where ϖ_t is the noise input function with covariance Q_t and υ_t is the sensor noise with covariance R_t. It is worthy to note that, even though we are assuming a smooth change and applying zero-order dynamic model, the method would be able to track abrupt changes with an adequate tuning of the matrixes Q_t and/or R_t. We will check this assertion in the result section (see Table 4.5), where a high value of the noise factor allows tracking successfully abrupt illumination changes.

The state update equation of the PDAF is

$$\hat{a}_t = \hat{a}_{t|t-1} + W_t\nu_t \tag{4.51}$$

where the combined innovation ν_t of the corresponding innovation ν_t^r of each possible measurement z_t^r is given by

$$\nu_t = \sum_{r=1}^{m_t} \beta_t^r \nu_t^r \tag{4.52}$$

$$\nu_t^r = z_t^r - z_{t|t-1} \tag{4.53}$$

and β_k^r is the conditional probability of association obtained from the PDA procedure. m_t is the number of measurements in the validation region at time t. In our case,

$$\beta_k^r \triangleq P\left(z_k^r|Z^k\right) = exp^{-0.5*\nu_t^{r'} S_t \nu_t^r} \tag{4.54}$$

The gain W_t and the covariance of the innovation are the same as in the standard filter

$$W_t = P_{t|t-1} H_t' S_t^{-1} \tag{4.55}$$

$$S_t = H_t P_{t|t-1} H_t' + R_t \tag{4.56}$$

The covariance associated with the updated state is

$$P_t = \beta_k^0 P_{t|t-1} - \left[1 - \beta_k^0\right] P_{t|t-1}^c + \tilde{P}_t \tag{4.57}$$

where the covariance of the stated updated with the correct measurement is

$$P_{t|t-1}^c = P_{t|t-1} - W_t S_t W_t' \tag{4.58}$$

and the spread of the innovation term (similar to the spread of the means term in a mixture) is

$$\tilde{P}_t \triangleq W_t \left[\sum_{r=1}^{m_t} \beta_t^r \nu_t^r \nu_t^{r'} - \nu_t \nu_t'\right] W_t' \tag{4.59}$$

Noise parameters tuning The presented methodology requires a set of parameters, such as noise or update speed. These parameters must be configured according to the environment, just like conventional approaches. Nevertheless, as noted, Kalman filter estimation can fail due to the inappropriate parameters.

Some automatisation mechanisms to tune some variables like state and measurement noise have been proposed in the literature. The robust adaptive filtering [Peng et al., 2005] is one of the most popular approaches. It provides an automatic tune of the corresponding tracking filter. In this way, the noise parameter can change to adapt its value depending on the goodness of the new measurement. Two factors are taken into account:

- The Kalman filter with fixed Q_t and R_t is sensitive to outliers
- Q_t and R_t vary over time in dynamic scenes.

Firstly, as it is known, KF provides the optimal estimates for Gaussian distributions. In practice, due to occlusions or imperfections of the motion model used, the measurement error may deviate too much from the gaussian distribution. These outliers should be removed from the filter state estimation to keep the assumptions of the KF satisfied. By using a robust estimation scheme, when the innovation exceeds a certain times its standard deviation, the measurement is rejected and the state is not updated. This methodology makes the template robust against short-time occlusions.

$$\hat{a}_t = \left\{ \begin{array}{cc} \hat{a}_{t|t-1} + W_t \nu_t^i & if \quad \nu_t < \alpha \cdot \bar{\nu} \\ \hat{a}_{t|t-1} & otherwise \end{array} \right. \tag{4.60}$$

where α is a predefined coefficient.

Secondly, to mitigate the variation of noise parameters over time, a special kind of KF, called adaptive or self-tuning [Peng et al., 2005; Maybeck, 1982], allows us to estimate these

Algorithm 6: RB Particle Filter Summary

Given a particle set $\{l_{t-1}^{(i)}, \omega_{t-1}^{(i)}, \hat{a}_{t-1}^{(i)}, P_{t-1}^{(i)}\}_{i=1}^{N}$ which represents the posterior probability $p(l_{t-1}|z_{t-1})$ and z_t.

1. Propagate samples.

   Resampling: Randomly select a particle $l_{t-1}^{(i)}$ from the previous time step according to the weights $\omega_{t-1}^{(i)}$.

   Propagation: Sample from the motion model $P(l_t|l_{t-1}^{(i)})$ for the chosen particle to obtain a prediction location $\hat{l}_t^{(j)}$.

   Kalman prediction: Obtain expected state $\hat{a}_{t|t-1}^{(j)}$ and the uncertainty estimated $P_{t|t-1}^{(j)}$ from $a_{t-1}^{(i)}$, $P_{t-1}^{(i)}$

2. For every new particle j at location $\hat{l}_t^{(j)}$

   Update $\hat{a}_t^{(j)}$ and error covariance $P_t^{(j)}$ $(j = 1, \dots, N)$ from $\hat{a}_{t|t-1}^{(j)}$ and $P_{t|t-1}^{(j)}$ with measurement $z_t = \phi(\hat{l}_t^{(j)})$ according to the PDAF equations (4.51) and (4.57). $\phi(l)$ is a function which extracts the measurements (pixels) in the validation region around l.

   Evaluate weight $\omega_t^{(j)}$ according to (4.71), (4.73) or (4.78) and normalise them $\sum_{j=1}^{N} \omega_t^{(j)} = 1$.

3. Estimate the new position of the state $E[X_t] = \sum_{j=1}^{N} \omega_t^{(j)} \cdot l_t^{(j)}$

parameters simultaneously with the states. The input data for the estimation of the noise parameters are the innovation sequence and its variance, which are estimated by averaging over the last K frames:

$$\bar{\nu}^2 = \frac{1}{K} \sum_{i=t-K+1}^{t} \nu_t^2 \tag{4.61}$$

where ν_t is the innovation seen in Eq.4.53.

One of the two noise parameters can be readjusted if the other one is known beforehand. Tuning one parameter is usually sufficient for the filter to adapt to changes of orientation or illumination, particularly, the re-estimation of Q_t is especially useful. This is because object intensities change faster, leading to the increase of $\bar{\nu}^2$, and hence, the increase of Q_t as well. Let us assume the measurement noise R_t is known, then the state noise is estimated as:

$$Q_t = \bar{\nu}^2 - R_t - P_{t-1} \tag{4.62}$$

The higher value of Q_t weighs the measurement in Eq. 4.60 more heavily, and therefore, keeps the template up-to-date with the object appearance. It remains to specify Q_t and the initial values for Q_t and P_t. They are set up such that initially the states and measurements have equal weights: $Q_0 = 0$ $R_0 = 0.5 \cdot \nu_1^2$ $P_0 = 0.5 \cdot \nu_1^2$

Although this method looks coherent, its practical application has not provided convincing results, even makes results worse than conventional filter with fixed parameters. We think it is due to the double tracking strategy and the feedback between the stages. While in a robust adaptive filter a high value of the innovation can be caused for a bad prediction or a bad measurement, in our case an extra factor should be taken into account: a bad prediction of the previous state vector parameters. This external factor destabilises the robust adaptive strategy because the KF can increase the noise parameters and discard measurements when the high value of the innovation is due to a bad estimation of the other state parameter. In our case, the algorithm can try to update the colour model assuming a illumination change when the change is due to a bad estimation of the location. A similar reflection is carry out in [Maybeck, 1982], in which it is postulated that simultaneous estimation of more than one noise parameters in general is not reliable for the same reason.

4.5.2 Colour Appearance Modelling

The simplest technique to model the appearance coefficients consists in assuming the target as a one-colour region and modelling it as a Gaussian using two parameters: mean μ and covariance σ^2. Although this assumption limits the generality of the methodology, it can be easily extended dividing the target into a predefined set of one-colour regions [Pérez et al., 2002].

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \tag{4.63}$$

A more sophisticated technique is to generate a model of the target based on a Mixture of Gaussian (MoG). Depending on the number of Gaussians chosen, up to $2 * N^o Gauss$ parameters should be estimated.

$$p(x) = \sum_{i=1}^{N} w_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(x-\mu_i)^2/2\sigma_i^2} \tag{4.64}$$

Both previous techniques are based on parametric methodologies. With regard to non-parametric techniques, histograms have been broadly applied as a functional method to model regions. Nevertheless, histograms exhibit some problems, as pointed by [Bohyung Han, 2004; Elgammal et al., 2003]. As it has been shown through this chapter, one way to alleviate the first two problems and, at the same time, estimating the underlying density without having to store the complete set of data is the kernel density estimation technique (KDE). In this technique the underlying probability density function is estimated as

$$p(x) = \sum_{i=1}^{N} w_i K(x - x_i) p(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i) \tag{4.65}$$

where K is a "kernel function" (typically a Gaussian) centered at the data points, $x_i, i = 1, \ldots, N$ and w_i are weighting coefficients (typically uniform weights are used, i.e., $w_i = 1/N$).

$$K_\sigma(u) = \frac{1}{\sqrt{2\pi}\sigma} e^{-u^2/2\sigma^2} \tag{4.66}$$

Note that choosing the Gaussian as a kernel function is different from fitting the distribution to a mixture of Gaussian model. Here, the Gaussian is only used as a function that weighs the data points.

Since two or three dimensional colour spaces are usually used in colour modelling, the previous equation must be adapted. For multivariate data we use the following expression:

$$K(u) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{(-\frac{1}{2}u^T \Sigma^{-1} u)} \tag{4.67}$$

Note that we are using the covariance matrix Σ instead of standard deviation of the Gaussian σ in Eq. (4.66) and d denotes the dimension of the colour space.

KDE does not assume any specific underlying distribution and, theoretically, the estimate can converge to any density shape with enough samples [Bohyung Han, 2004; Elgammal et al., 2003]. Therefore, this approach is suitable to model the colour distribution of regions with patterns and mixture of colours. If the underlying distribution is a mixture of Gaussians, KDE converges to the right density with a small number of samples. Unlike parametric fitting of a mixture of Gaussians, KDE is a more general approach that does not require the selection of the number of Gaussians to be fitted. However, it exhibits a significant disadvantage in that it is computationally intensive. To overcome this limitation, several authors propose the use of Fast Gauss Transform, to estimate efficiently Kernel Density. Elgammal et al. [Elgammal et al., 2003] use this approach for an efficient Kernel density estimation with applications to colour modelling and tracking.

4.5.3 Colour-Based Hypothesis Evaluation

In this section we will compare several approaches based on RB particle filter combined with different kinds of colour models. We have applied our tracking algorithm with a simple parametric models and with a model that can be considered either as a limit of a KDE model or a multi-part model.

4.5.3.1 Parametric model

Assuming one-colour region, we can model the target as a single Gaussian. For this assumption, the likelihood function is given by:

$$P(z_t|\hat{l}_t^{(j)}, a_t) \propto \sum_{x,y \in \hat{l}_t^{(j)}}^{area} p(I(x,y)|a_t) \tag{4.68}$$

where I is the colour image.

Taking into account the Gaussian approach for colour estimation, we have:

$$P(z_t|\hat{l}_t^{(j)}, a_t) \propto \frac{1}{(2\pi)^{d/2}\sqrt{|R_t|}} \sum_{x,y \in \hat{l}_t^{(j)}}^{area} e^{-\frac{1}{2}\|I(x,y)-a_t\|^2_{R_t}} \tag{4.69}$$

where $\| A - B \|_{C_B}$ is the Mahalanobis distance between A and B.

Substituting in (4.47) we obtain:

$$\omega_t^{(j)} \propto \int_{a_t} \sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{(2\pi)^d \sqrt{|P_{t|t-1}^{(j)}||R_t|}} e^{-\frac{1}{2}\|I(x,y)-a_t\|^2_{R_t}} \cdot e^{-\frac{1}{2}\|a_t - \hat{a}_{t-1}^{(i)}\|^2_{P_{t|t-1}^{(j)}}} \tag{4.70}$$

Since the product of two Gaussians is, up to a constant, another Gaussian [Roweiss, 1999], the above integral can be computed giving

$$\omega_t^{(j)} = \sum_{x,y \in \hat{l}_t^{(j)}}^{area} k^{(j)} \cdot e^{-\frac{1}{2}\left(\|I(x,y)-\hat{a}_t^{(j)}\|^2_{R_t} + \|\hat{a}_t^{(j)} - \hat{a}_{t-1}^{(i)}\|^2_{P_{t|t-1}^{(j)}} \right)} \tag{4.71}$$

where $k^{(j)}$ is the normalization factor:

$$k^{(j)} = \frac{1}{area} \cdot \frac{1}{(2\pi)^{d/2}} \cdot \frac{|P_t^{(j)}|^{1/2}}{|R_t|^{1/2} \cdot |P_{t|t-1}^{(j)}|^{1/2}} \tag{4.72}$$

In the event of MoG modelling, we can easily extend this solution for a Gaussian Mixture of C components:

$$\omega_t^{(j)} = \sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{C} \sum_{c=1}^{C} k^{(c,j)} \cdot e^{-\frac{1}{2}\left(\|I(x,y) - \hat{a}_t^{(c,j)}\|^2_{R_t^{(c)}} + \|\hat{a}_t^{(c,j)} - \hat{a}_{t-1}^{(c,i)}\|^2_{P_{t|t-1}^{(c,j)}} \right)} \tag{4.73}$$

4.5.3.2 Non-Parametric model

Although parametric methods can model a target if an adequate number of parameters has been chosen, they have problems to characterise a complex target with a non-uniform appearance model. Moreover, this complexity together with a high number of parameters increase the difficulty of the initialisation process. If the model should be updated, the problem is even harder since the required number of parameter could change over time. In these situations, it is possible to simplify the description of the model by means of non-parametric methods, which are a more general approach that do not assume any specific shape for the density function and are able to represent very complicated densities effectively. Thus, these methodologies could extend the potential of the algorithm to model more complex targets, as well as simplifying the initialisation process to the extraction of a good training sample.

If we wish model the colour appearance using non-parametric methods such as histograms, we can apply the same previous methodology by applying kernel density estimation (KDE) techniques. In this approach, the likelihood model $P(z_t|\hat{l}_t^{(j)}, a_t)$ is given by:

$$\begin{aligned} P(z_t|\hat{l}_t^{(j)}, a_t) \quad & \propto \textstyle\sum_{x,y \in \hat{l}_t^{(j)}}^{area} p(I(x,y)|a_t) \\ & \propto \textstyle\sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{M} \sum_{m=1}^{M} K_m(I(x,y) - a_t^m) \\ & \propto \textstyle\sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{(2\pi)^{d/2}|R_t^m|^{1/2}} e^{-\frac{1}{2}\|I(x,y) - a_t^{(}m)\|^2_{R_t^m}} \end{aligned} \tag{4.74}$$

KDE converts the histogram into a Gaussian mixture, with the same number of Gaussians as samples used for the reference model. Therefore, the posterior probability is similar to (4.73) being now the parameter C equal to the number of training samples M.

It is clear the advantages that non-parametric techniques exhibit in relation to parametric to model complex distributions. However, this approach exhibits a great limitation: if the number of samples used to model the target is too high, (4.74) becomes unaffordable to compute since it involves a double summation. This is due to the fact that each pixel influences all the Gaussians that compose the KDE model of the region.

In order to solve efficiently the posterior probability, we introduce spatial information into the equations. This can de done by considering a multi-part object. The target is made of a set of patches (or pixels, in the limit), each one modelled by a Gaussian which is located in the patch coordinates. We have the same number of Gaussians than patches composing the target region and we know the relationship between patches and Gaussians. Now, each patch is considered only for those Gaussians that were originally in the same relative location inside the region, on the contrary to the conventional KDE approach. A piece of the region can be composed of a unique pixel (according to the classical KDE methodology) or a group of pixels (for computational reasons), and the number of pixels that composes it changes if the size of the target change over time.

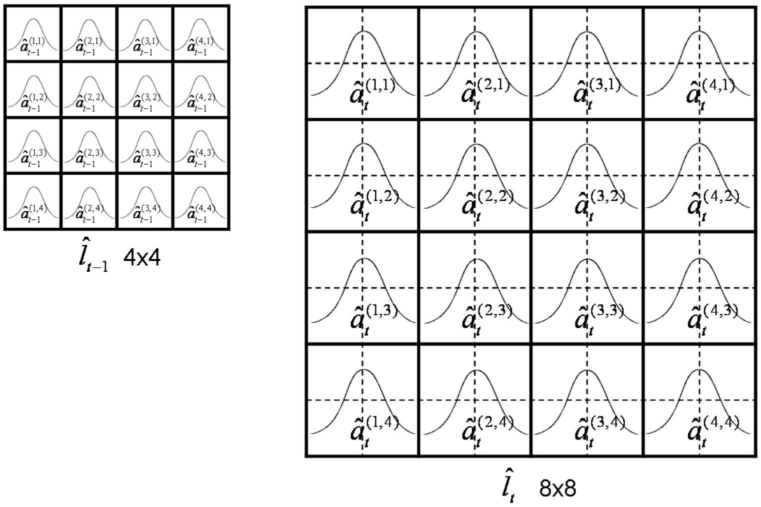

$\hat{a}_{t-1}^{(1,1)}$ $\hat{a}_{t-1}^{(2,1)}$ $\hat{a}_{t-1}^{(3,1)}$ $\hat{a}_{t-1}^{(4,1)}$
$\hat{a}_{t-1}^{(1,2)}$ $\hat{a}_{t-1}^{(2,2)}$ $\hat{a}_{t-1}^{(3,2)}$ $\hat{a}_{t-1}^{(4,2)}$
$\hat{a}_{t-1}^{(1,3)}$ $\hat{a}_{t-1}^{(2,3)}$ $\hat{a}_{t-1}^{(3,3)}$ $\hat{a}_{t-1}^{(4,3)}$
$\hat{a}_{t-1}^{(1,4)}$ $\hat{a}_{t-1}^{(2,4)}$ $\hat{a}_{t-1}^{(3,4)}$ $\hat{a}_{t-1}^{(4,4)}$
$\hat{l}_{t-1}$ 4x4
$\tilde{a}_{t}^{(1,1)}$ $\tilde{a}_{t}^{(2,1)}$ $\tilde{a}_{t}^{(3,1)}$ $\tilde{a}_{t}^{(4,1)}$
$\tilde{a}_{t}^{(1,2)}$ $\tilde{a}_{t}^{(2,2)}$ $\tilde{a}_{t}^{(3,2)}$ $\tilde{a}_{t}^{(4,2)}$
$\tilde{a}_{t}^{(1,3)}$ $\tilde{a}_{t}^{(2,3)}$ $\tilde{a}_{t}^{(3,3)}$ $\tilde{a}_{t}^{(4,3)}$
$\tilde{a}_{t}^{(1,4)}$ $\tilde{a}_{t}^{(2,4)}$ $\tilde{a}_{t}^{(3,4)}$ $\tilde{a}_{t}^{(4,4)}$
$\hat{l}_{t}$ 8x8

Figure 4.22: Non-parametric likelihood example. A rectangular 4x4-pixel target at time $t-1$ evolves to a 8x8-pixel region at time t. Initially, the target has been divided into 16 patches and there is one colour appearance coefficient (modelled as one Gaussian) per pixel. At the next time step, since the size has increased, 4 pixels are used for updating each coefficient by means of PDAF.

A graphical example can be seen in Figure 4.22. If we considered M_1 pixels for the model and M_2 pixels in the area under evaluation, (4.74) involves M_1xM_2 kernel evaluations. However, following the proposed methodology, and considering the example depicted in Figure 4.22, where $P = 16$ patches are used, now every patch for the model has M_1/P pixels and every patch in the evaluation image has M_2/P pixels. Taking into account we have to evaluate 16 different patches the computation involves M_1x$M_2/16$ kernel evaluations, just lower than an order of magnitude in relation to the simple KDE. The same figure shows the problem related to object size change. Initially, the target has been divided into 16 patches and there is one colour variable (modelled in this example as a single Gaussian) per region. At the next time step, since the size has increased, a bigger region is used for updating the Gaussian parameters by means of PDAF.

The advantages of PDAF are here stood out. When the size of the object is increased, new pixels with new details appear, which must be assigned to existing Gaussians by proximity. In these situations, a conventional KF would require, as measurement, the mean of all the corresponding pixels for that Gaussian. While this mean could be distorted by a few pixels, the PDAF could obtain a better estimation by reducing their contribution according to their reliability. We should remember that the region can be composed of a small number of pixels, and therefore just one pixel can damage the KF estimation.

In the limit, when every patch is composed of a single pixel, the double summation in (4.74) becomes in a single one, since $M_1/P = 1$. Thus, the likelihood model $P(z_t|\hat{l}_t^{(j)}, a_t)$ is given by:

$$P(z_t|\hat{l}_t^{(j)}, a_t) \propto \sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{(2\pi)^{d/2}|R_t^{(x,y)}|^{1/2}} e^{-\frac{1}{2}\|I(x,y)-a_t^{(x,y)}\|^2_{R_t^{(x,y)}}} \tag{4.75}$$

where $a_t^{(x,y)}$ is the estimated color model for the pixel in the reference model corresponding to that pixel $I(x,y)$ in the image.

And substituting (4.75) in (4.47) we obtain:

$$\omega_t^{(j)} \propto \int_{a_t} \{ \sum_{x,y \in \hat{l}_t^{(j)}}^{area} \frac{1}{(2\pi)^{d/2} |R_t^{(x,y)}|^{1/2}} e^{-\frac{1}{2} \| I(x,y) - a_t^{(x,y)} \|^2_{R_t^{(x,y)}}} \cdot \frac{1}{(2\pi)^{d/2} |P_{t|t-1}^{(j)(x,y)}|^{1/2}} e^{-\frac{1}{2} \| a_t^{(x,y)} - \hat{a}_{t-1}^{(i)(x,y)} \|^2_{P_{t|t-1}^{(j)(x,y)}}} \} \tag{4.76}$$

Weights are calculated as:

$$\omega_t^{(j)} \propto \sum_{x,y \in \hat{l}_t^{(j)}}^{area} \int_{a_t} \frac{1}{k} \cdot e^{-\frac{1}{2} \left\{ \| I(x,y) - a_t^{(x,y)} \|^2_{R_t^{(x,y)}} + \| a_t^{(x,y)} - \hat{a}_{t-1}^{(i)(x,y)} \|^2_{P_{t|t-1}^{(j)(x,y)}} \right\}} \tag{4.77}$$

where $k = (2\pi)^{d/2} |R_t^{(x,y)}|^{1/2} \cdot (2\pi)^{d/2} |P_{t|t-1}^{(j)(x,y)}|^{1/2}$

Finally, as in Equation (4.71), the above integral can be computed giving:

$$\omega_t^{(j)} = \sum_{x,y \in \hat{l}_t^{(j)}}^{area} k'^{(j)(x,y)} \cdot e^{-\frac{1}{2} \left\{ \| I(x,y) - \hat{a}_t^{(j)(x,y)} \|^2_{R_t^{(x,y)}} + \| \hat{a}_t^{(j)(x,y)} - \hat{a}_{t-1}^{(i)(x,y)} \|^2_{P_{t|t-1}^{(j)(x,y)}} \right\}} \tag{4.78}$$

with

$$k'^{(j)(x,y)} = \frac{1}{area} \cdot \frac{1}{(2\pi)^{d/2}} \cdot \frac{|P_t^{(j)(x,y)}|^{1/2}}{|R_t^{(x,y)}|^{1/2} \cdot |P_{t|t-1}^{(j)(x,y)}|^{1/2}} \tag{4.79}$$

We use the term KDE to refer to the proposed methodology for the sake of clarity, but this is not an accurate definition. The methodology is rather a limiting case of KDE with two differences. First, in a typical KDE a sample pixel has influence on the probability of all other pixels. Second, the covariance is fixed, while we allow a different covariance for each Gaussian, $R_t^{(x,y)}$. The introduction of spatial information means that rotation invariance is somehow lost, but this can be compensated by the appearance tracking. On the other hand, spatial information gives three advantages. First, a more accurate estimation can be obtained if the target has some characteristic details. Second, it makes the system robust against partial occlusions. Third, a significative reduction of the computational load is achieved.

Computational issues: The double updating technique By incorporating the appearance model into the state vector, each particle evolves its own appearance model. This implies a high number of particles to avoid that the model degeneration could produce a tracking failure in spite of a good target location, or viceversa.

We propose a second updating step, once the mean estimation has been computed, in order to refine the appearance model. We call this strategy double updating. First each particle is updated with its corresponding region. After estimating the mean state, an extra updating step with the region of the mean state is applied over all the samples. In this way, less particles are needed due to the smaller and slower degeneration of the colour model.

4.5.4 Comparison with Gray Level Algorithm

As explained in the introduction of this section, there are other valid approaches to compensate the colour changes, based on gamma correction. To check the advantages and drawbacks that these methodologies introduce, a comparison has been done. Gray Level algorithm is based on

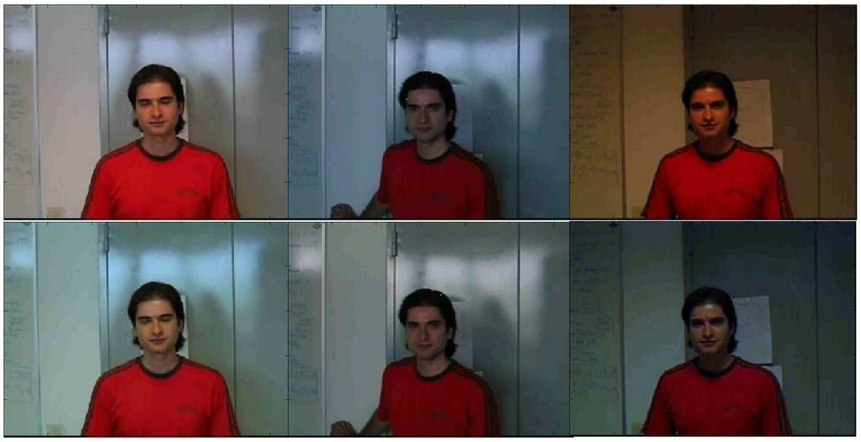


Figure 4.23: Gray Level algorithm results

the assumption that given an image with sufficient amount of colour variations, the average value of the R, G and B components of the image should average to a common gray value. This assumption can be considered true in many real situations because it is usually true that there are a lot of different colour variations in a unique image and those variations in colour are random and independent. In that case, the average would converge to the mean value, gray, by given an enough amount of samples. Colour balancing algorithms can apply this assumption by forcing the images to have a common average gray value for the R, G and B components. In the case an image is taken by a digital camera under a particular lighting environment, the effect of the special lighting cast can be removed by enforcing the gray world assumption on the image. As a result of approximation, the colour of the image is much closer to the original scene.

The algorithm applied a gamma correction by weighting the pixel value for a factor calculated as the mean of the pixels:

$$\bar{R} = \frac{1}{N} \sum_{(x,y)\in I} R_{in}(x,y) \quad \bar{G} = \frac{1}{N} \sum_{(x,y)\in I} G_{in}(x,y) \quad \bar{B} = \frac{1}{N} \sum_{(x,y)\in I} B_{in}(x,y) \tag{4.80}$$

where N is the number of pixel in the image and $I = \{R_{in}, G_{in}, B_{in}\}$ is the image composed of the three RGB channels.

We also calculate the global mean

$$G\bar{ray} = \frac{\bar{R} + \bar{G} + \bar{B}}{3} \tag{4.81}$$

The value of the pixels after applying the correction is:

$$R_{out} = \frac{G\bar{ray}}{\bar{R} \cdot f} \cdot R_{in} \quad G_{out} = \frac{G\bar{ray}}{\bar{G} \cdot f} \cdot G_{in} \quad B_{out} = \frac{G\bar{ray}}{\bar{B} \cdot f} \cdot B_{in} \tag{4.82}$$

where f is the normalisation factor $f = \max\left[R_{out}, G_{out}, B_{out}\right]_{x,y}$.

Using this approach, the sequence shown in Fig. 4.24 can be solved without update algorithm (see Fig. 4.23). However, it fails for more complex sequences like 4.26. In addition, this algorithm could be applied for illumination changes but is not useful to assimilate changes in the colour or appearance of the object.

	KDE		Gaussian		Double-Update KDE	
	q=10	q=0.5	q=10	q=0.5	q=10	q=0.5
Error [pix]	41.84	59.78	22.77	51.11	17.85	20.13
Lost	Y	Y	Y	Y	N	Y

	Conventional				Transductive Inference			
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = \beta = 0.1$	$\alpha = \beta = 0.25$	$\alpha = \beta = 0.7$	$\alpha = \beta = 0.95$
Error [pix]	65.8	67.6	79.83	83.38	24.97	27.12	114.1	116.8
Lost	Y	Y	Y	Y	N	N	Y	Y

Table 4.5: Comparison between our updating strategy and conventional updating techniques.

4.5.5 Experimental Results

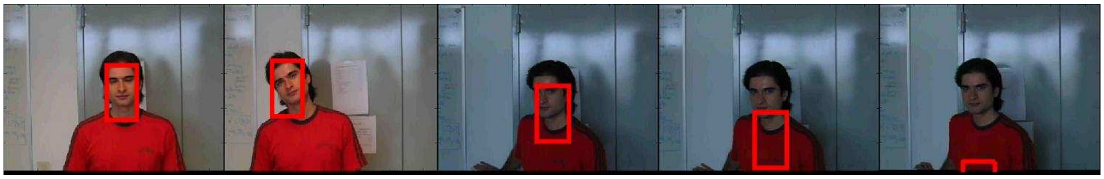
In this section we analyse the results of the proposed algorithm using parametric and non parametric colour models. For that, we have applied the algorithm to face sequences since faces present a characteristic colour appearance which can be easily modelled under normal conditions using both methodologies. However, it does not mean that the present approach is limited to this kind of images. Thanks to the KDE modification, any target can be modelled, tracked and updated.

Tracking results can be seen in Figures 4.24-4.25, where both strong and sudden light changes happen as well as gradual ones. We have tested the system using our own sequence, with two strong illumination changes (frame 85 and 140) and a slow change (frame 203). Table 4.5 shows the comparison between our updating strategy and two existing updating techniques: the conventional approach (see eq. 4.35) and a transductive inference-based approach (see eq. 4.36). Parameter q is a multiplication factor of Q_t which regularises the reliability of the measurement. Parameters α and/or β controls the updating velocity of the classical approaches (see Equations (4.35) and (4.36)). Both parameters are fixed manually at the beginning of the sequence and they control the updating speed.

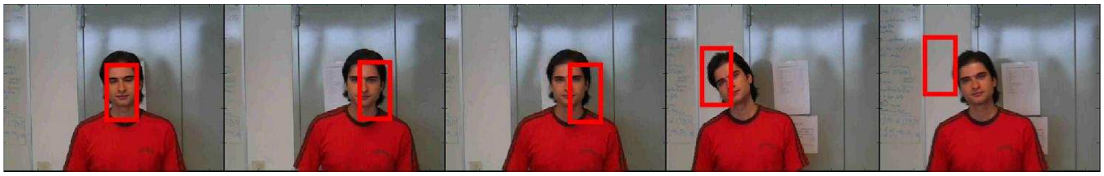
Although one of the classical approaches (4.36) gives reasonably good results for this sequence, we can see its weakness when stronger changes appear. For that, we have applied those algorithms that worked in the previous tests to a standard database in order to facilitate the comparison, the AVSS face tracking sequences [Maggio and Cavallaro, 2005]. In these sequences four different illumination situations appear, with slow or sudden transition, and strong or slight illumination conditions. Results are summarised in Table 4.6, where the divergence of the classical approach is clear, whereas our approach works correctly.

Finally, given the difficulty of finding standard sequences with illumination changes, diverse sequences from movies have been tested. Thus, we present the results for sequences extracted from *Lola Rentt*(Tom Tykwer, 1998) and *Blade Runner*(Ridley Scott, 1982), obtaining good results in spite of strong changes in the illumination conditions. Results are depicted in Figures 4.27,4.28,4.29,4.30. The double updating methodology has been applied since it gives the best results and the value of q has been set to 10.

Our method has been proved more robust against sudden illumination changes than methods in the literature. Both tested classical approaches (Eq. (4.35) and (4.36)) are not capable of providing a compromise between robustness against clutter and speed to respond against fast changes and they can easily adapt themselves erroneously to some part background. An example of this can be seen in Table 4.5 and 4.6. Although transductive inference is able to deal with the first sequence by setting a slow threshold, it fails when a strong or sudden change appear as in AVSS sequence.



(a) $\alpha = 0.1$. Frames:2,36,85,88,89.



(b) $\alpha = 0.7$. Frames:2,19,23,36,40.

Figure 4.24: Face tracking under illumination changes using conventional PF and the classical update.

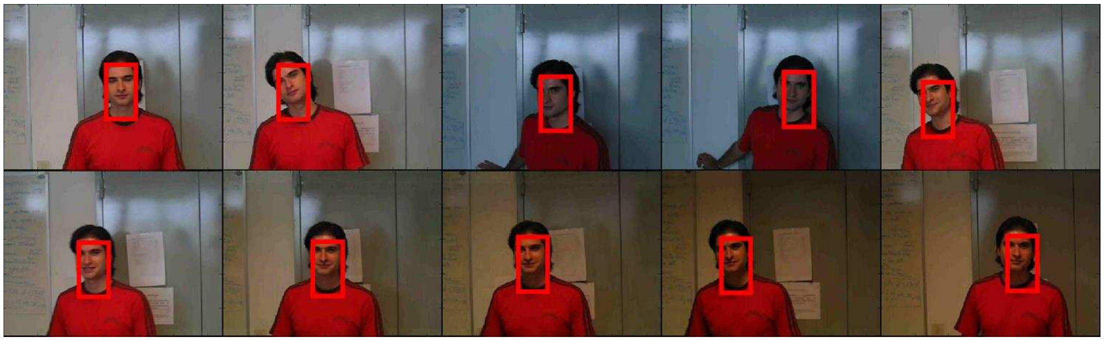
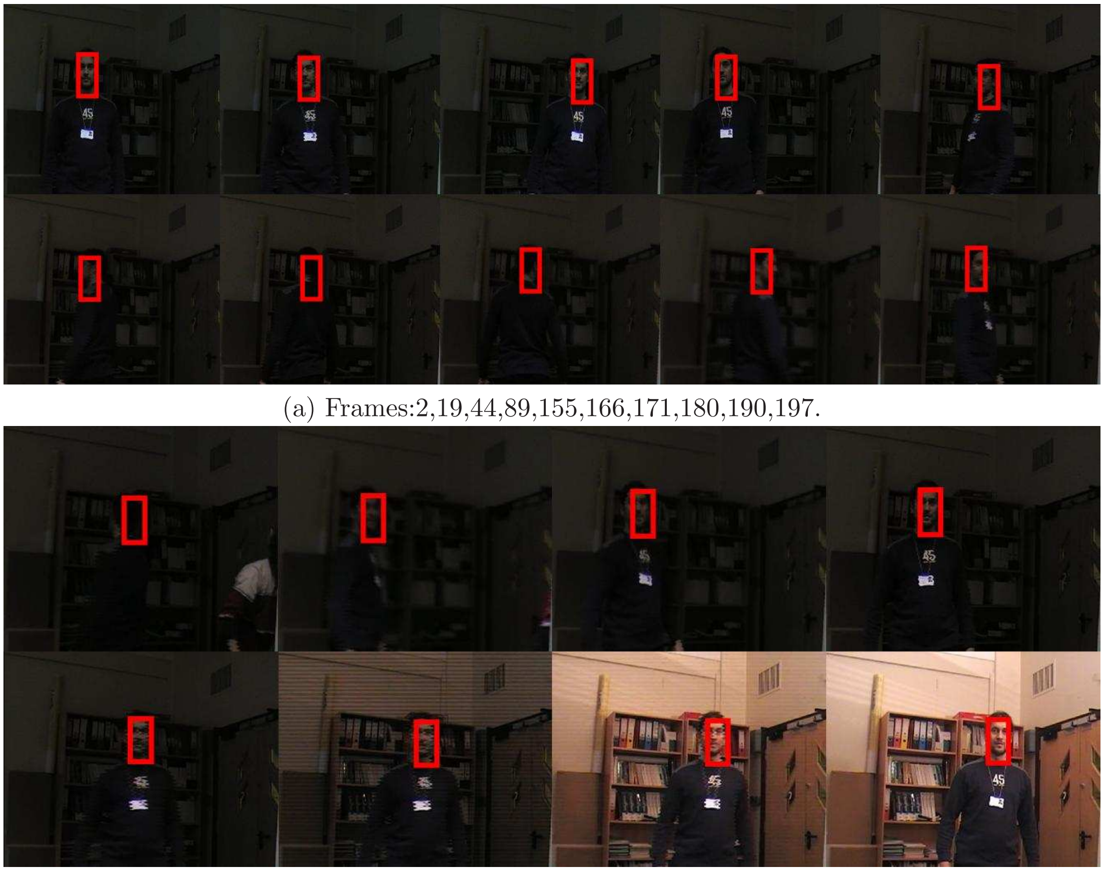


Figure 4.25: Tracking face with illumination changes using RBPF and non-parametric model. Frames:2, 36, 85, 119, 185, 196, 214, 217, 223, 279, 319.

	Double-Update KDE		Transductive Inference	
	Mean	Stand. Dev.	Mean	Stand. Dev.
Error [pix]	10.34	11.15	141.36	51.7

Table 4.6: Comparison for AVSS sequence



(a) Frames:2,19,44,89,155,166,171,180,190,197.

(b) Frames:419,439,445,477,487,489,493,500.

Figure 4.26: Face tracking under strong illumination changes. AVSS sequence *′motinas_toni_change_ill.avi′*.

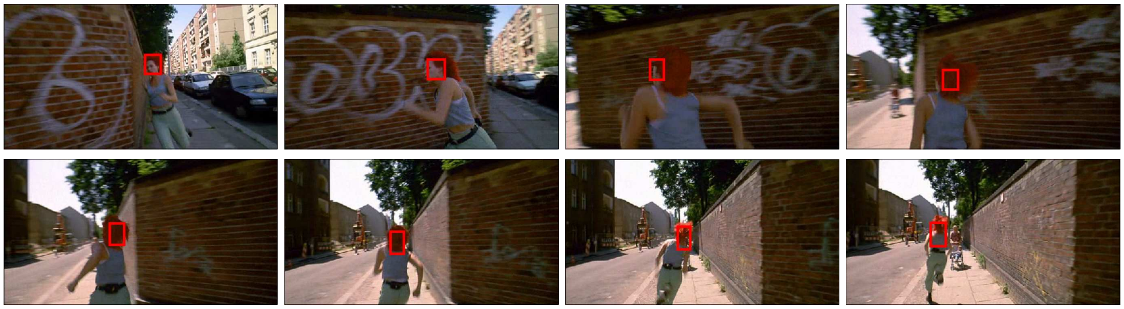


Figure 4.27: *Lola Rentt sequence 1*

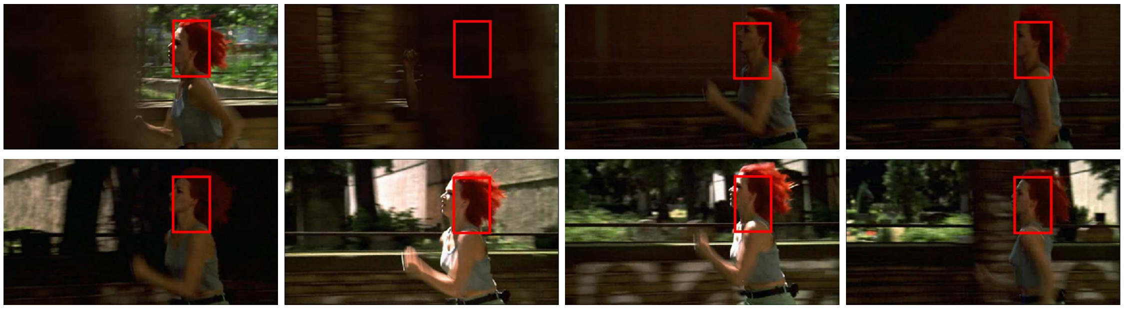


Figure 4.28: *Lola Rentt sequence 2*

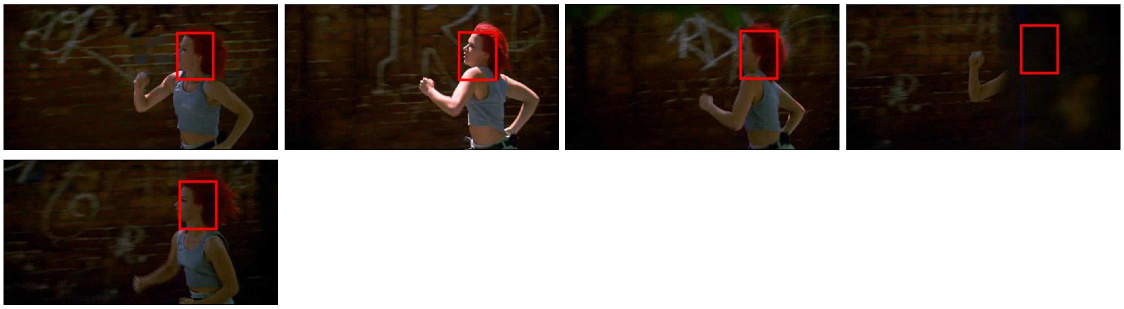


Figure 4.29: *Lola Rentt sequence 3*

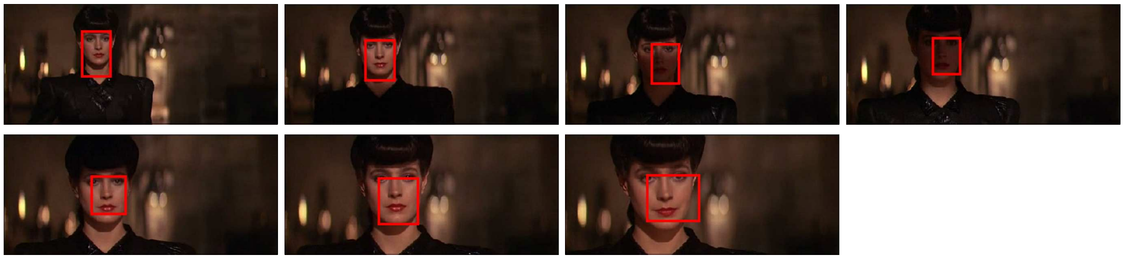


Figure 4.30: *Balde Runner sequence 1*

4.6 A Specific Application: Surveillance and Facial Tracking

Due to the characteristics of region tracking, as intermediate domain between punctual and articulated tracking, it enables more complex application than the first one but without achieving the level of specialisation that the second one makes reality. In this manner, region tracking is usually applied as first stage of higher level application, such as poser recovery, activity recognition or biometrics. Additionally, it makes possible to extend surveillance and non-ambitious applications to more complex scenarios, where occlusions and unexpected situations happen, or hardware limitations (due to monetary or technical reasons) make more difficult the success of punctual tracking algorithms.

We have decided to show the potential of this tracking domain applying the proposed algorithm to two particular fields: pedestrian and vehicle tracking for surveillance application, and facial tracking as cornerstone for biometric identification and gesture recognition.

To check the suitability of our proposal for surveillance and facial tracking, we have tested our algorithms with sequences from PETS and AVSS, that is, two well-known meeting specialised in video-surveillance which provide test database. The purpose of standard datasets is to establish a common test framework where different algorithms from different researches can be objectively evaluated and compared. In particular, our proposal has been tested with different test sequences that illustrate different situations: location/angle/size tracking, static or dynamic camera, colour or gray level model, single target or multi target (as we will see in the next chapter).

As assumptions, the initial colour or gray-level model corresponding to the target must be given in advance, or in lack of that, the initial pose of the target in the sequence. The target appearance model, in this chapter based on colour histogram, is updated in each frame to compensate lighting changes, following the approach shown in [Nummiaro et al., 2003].

Particle propagation is made by a first order model: an object moving with constant velocity plus a stochastic component consisting in acceleration noise.

4.6.1 Maritime Surveillance

The present approach has been applied to the zodiac sequences taken from PETS 2005. In this domain of application, the zodiac tracking presents a high complexity due to multiple aspects like outdoor functioning with a high number of distracting moving targets (waves), dynamic non stationary occlusions, complex backgrounds and sudden changes in scale. Furthermore, the sequences has been recorded using moving camera.

Given the first frame, the initial histogram of the moving object to track (the zodiac) is obtained. For this sequence, the PDI is obtained using equation (4.22).

Particle distribution evolution throughout the different stages (Fig. 4.12) is depicted in Fig. 4.31. Once PDI (4.31.b) and its integral image are calculated, particles are distributed in accordance with the proposal distribution (4.31.c). By evaluating these particles in the first dimensions (location), the weights (4.31.c) needed to make the weighted resampling are obtained. Resulting particles (4.31.d) are finally evaluated in the remaining dimensions (size/angle) (4.31.e) using the likelihood function to estimate the final state (4.31.f). Colours like yellow and magenta represent low weights while cyan and dark blue represent high values.

Fig. 4.32 and 4.33 show the state target estimation (position and size) highlighted by a rectangle. Results of our system are depicted in Fig. 4.32, in which sudden camera shakes are carried out. Note the error in frame 100 due to a wave which changes the appearance of the boat for several frames. Frames of 4.33 have been chosen because the zodiac changes suddenly

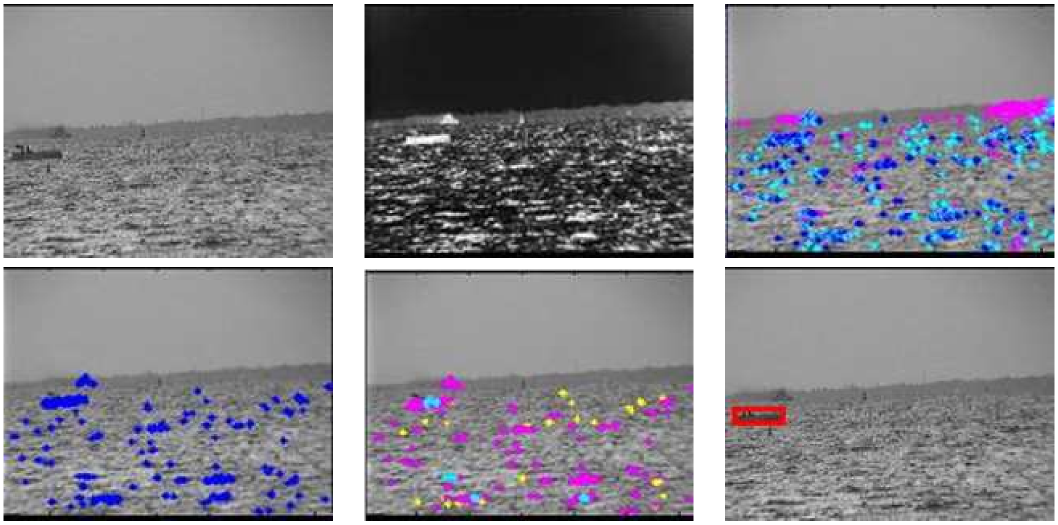


Figure 4.31: Zodiac sequence: a) Frame 241; b) PDI; c) Particles distributed by the proposal distribution; d) weighted resampling; e) posterior probability.

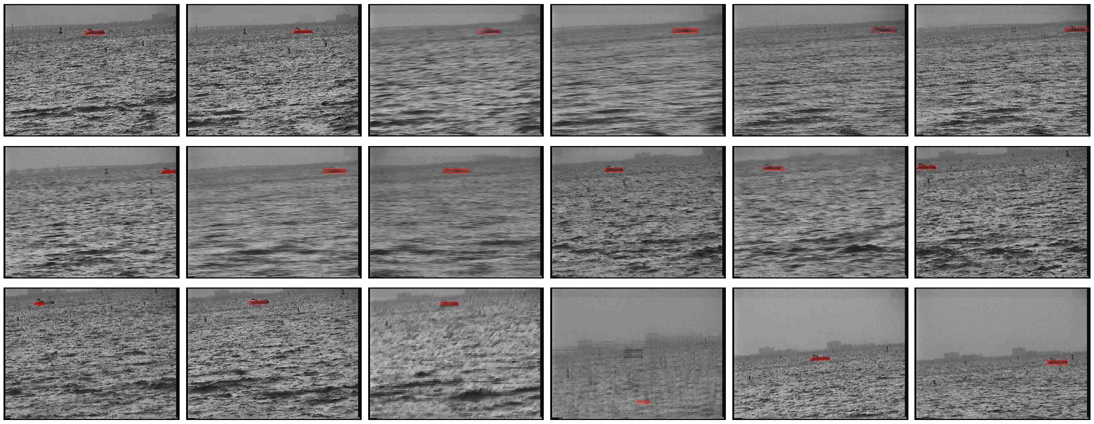


Figure 4.32: Zodiac sequence 7a. Frames: 25, 50,53, 54, 55, 56, 58, 59, 62, 63, 67, 72, 100, 126, 134, 143, 197.

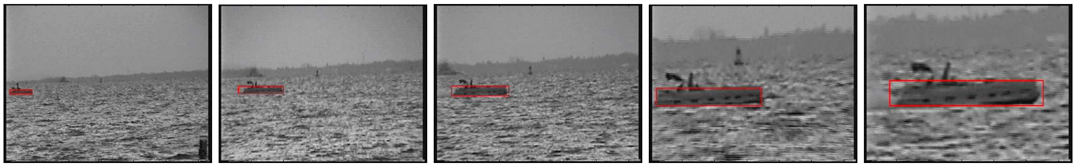


Figure 4.33: Zodiac sequence 7a. Frames: 215, 235, 255, 265, 275, 304

in scale, so the robustness of our method to cope with this situation is highlighted, thanks to the partitioned sampling strategy. Fig. 4.35 shows another example of the performance of the algorithm. In this sequence, the zodiac suffers from a reduction in scale, while it is surrounding by many distracters. In addition, although the target is completely lost for a few frames due to an external occlusion and the image gray levels experiment a great change during the following frames, the tracking system is able to recover the target.

The proposed approach has been compared against other similar colour-based particle filters found in the literature [Nummiaro et al., 2003; Pérez et al., 2002; Porikli, 2005; Okuma et al., 2003]. Table 4.7 shows the results in mean square error and computational cost. As we can see, using Integral Image allows a relevant reduction factor for each particle estimation. Although an initial cost is added in order to create the integral matrix, it is compensated by the large number of particles usually required for the evaluation. Integral histogram method has a similar performance but with a higher initial cost. This fixed computational load is due to the fact that 2^8-bin histograms are used and therefore 256 integral images are calculated. The low number of required particles is not able to make up for this initial cost. Algorithms have been tested using MATLAB running on a Pentium IV 2.4Ghz so that the time costs obtained are only useful to compare one another. Preliminary results in C++ show that our algorithm is able to process a frame and to track 11 targets in 77ms (running on a Pentium IV 3.0Ghz with 2Gb RAM). On the contrary, conventional approaches can not be fast enough for multi-target applications.

Table 4.7: Results obtained comparing our efficient PF with conventional colour-based particle filter (using different number of particles) and integral histograms (I.H.) for the Zodiac sequence.

	N=150	N=500	N=1000	I.H.	Our
MSE [pix^2]	190.2	165.3	171.9	131.6	112.1
Mean N	150	500	1000	314	324
Time PDI [s]	-	-	-	25.3	0.15
Time particles [s]	4.4	23.9	44.3	1.16	0.46
Total Time [s]	4.4	23.9	44.3	26.5	0.61

Total occlusion examples can be seen in Figures 4.35 and 4.36. In the first sequence, the zodiac suffers from a reduction in scale, while it is surrounding by many distracters, but the target is successfully tracked. In addition, although the target is completely lost along a few frames due to an external occlusion, and the image gray levels experiment a great change during the following frames, the tracking system is able to recover the target thanks to the dynamic-guided particles. During the period of time that the object is completely occluded, particles predicted by dynamic and some distracters in the PDI try to estimate the location of the target. Although the resulting estimation is not so good, the proposed methodology allows recovering the correct location of the target as soon as it appears again. The capacity of recovering the target after a failure is also shown in Figure 4.32, specifically in frames 134 and 143. The target is recovered as soon as its measurement is visible or good enough.

4.6.2 Facial Tracking

The uniformity of the skin colour and the presence of the neck make complicate to determine the size and the orientation of the face accurately. By linking and additional region with the t-shirt colour, the performance of the tracker grows considerably. This fact is specially clear in the estimation of size parameters. The explanation of this improvement is due to the fact that the colour density estimation proposed suppresses all information about

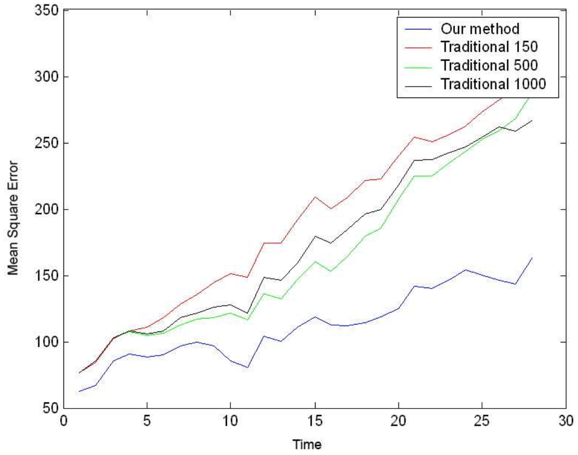


Figure 4.34: Results obtained comparing our efficient PF with conventional colour-based particle filter (Zodiac sequence).

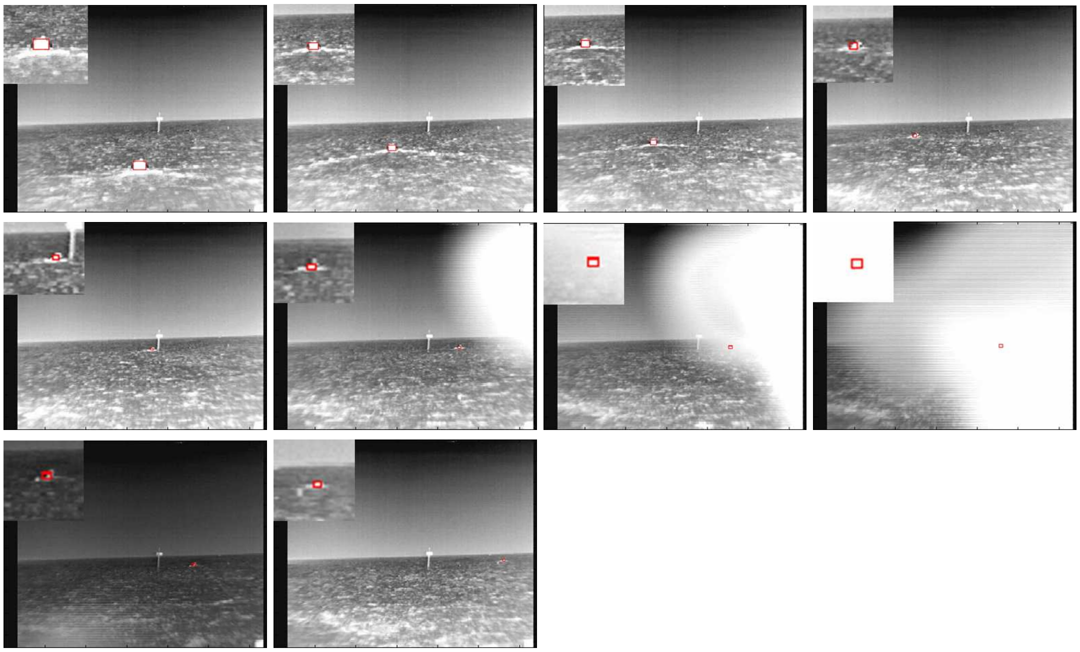


Figure 4.35: PETS Zodiac 2 sequence. Frames: 400, 907, 1132, 1562, 1737, 1899, 1911, 1921, 1931, 2216.

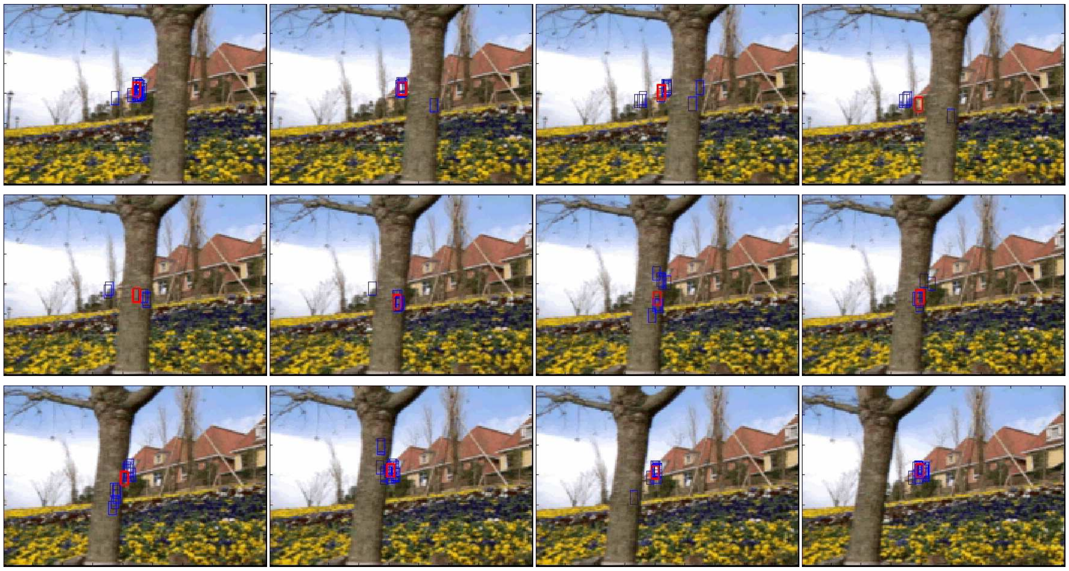


Figure 4.36: Flower garden sequence. Red rectangle represents the estimation and the blue ones are the best particles. Frames: 32, 35, 36, 38, 40, 43, 45, 46, 47, 51, 57.

the relative spatial arrangement of the different patches composing the target. By splitting the target region into sub-regions with individual reference colour models [Pérez et al., 2002; Okuma et al., 2004] this spatial layout is kept, and the tracking performance is improved. The computational cost increase that keeping the spatial layout implies is mitigated by our approach. Each individual evaluation of the sub-regions is really low time-consuming, and the proposal distribution can only be based on the most representative region. Examples of this technique are shown in Figures 4.38, 4.42 and 4.39. Fig. 4.38 and 4.39 show the results for the AVSS'07 face sequences [Maggio and Cavallaro, 2005].

Fig. 4.37 compares traditional PF with our approach. Size error of our method in the *Toni* sequence is larger than conventional PF. However this fact is deceptive: conventional filters loose the target generating a large error in location but a little one in size. This is due to the fact that the tracker without a valid measurement only changes its size due to a size noise of mean equal to zero, which is compensated by the sum of all the particles.

The performance of the algorithm to deal with orientation changes is checked using the *Webcam* sequence, Fig. 4.39. A fast and robust tracking in location, scale and orientation is achieved, even with the presence of partial and total occlusions shown in Fig. 4.35 and 4.39. For the first sequence, the algorithm works quite good till the end, where the PF fails because of a change of the illumination conditions which distorts the colour model. Obviously, the updating of the target appearance model following the approach given in [Nummiaro et al., 2003] is not good enough. A new technique has been considered to cope with this drawback in Section 4.5.

An extra sequence is depicted in Figures 4.40 and 4.41 to check the performance with moving camera and rotation out of the plane of the camera (3D rotations). Two experiments were made: partitioning the state in two stages (location and size) and in three stages (location, orientation and size). Thanks to the partitioned sampling, it is possible to track more parameters with a similar number of particles, obtaining thus a better global accuracy.

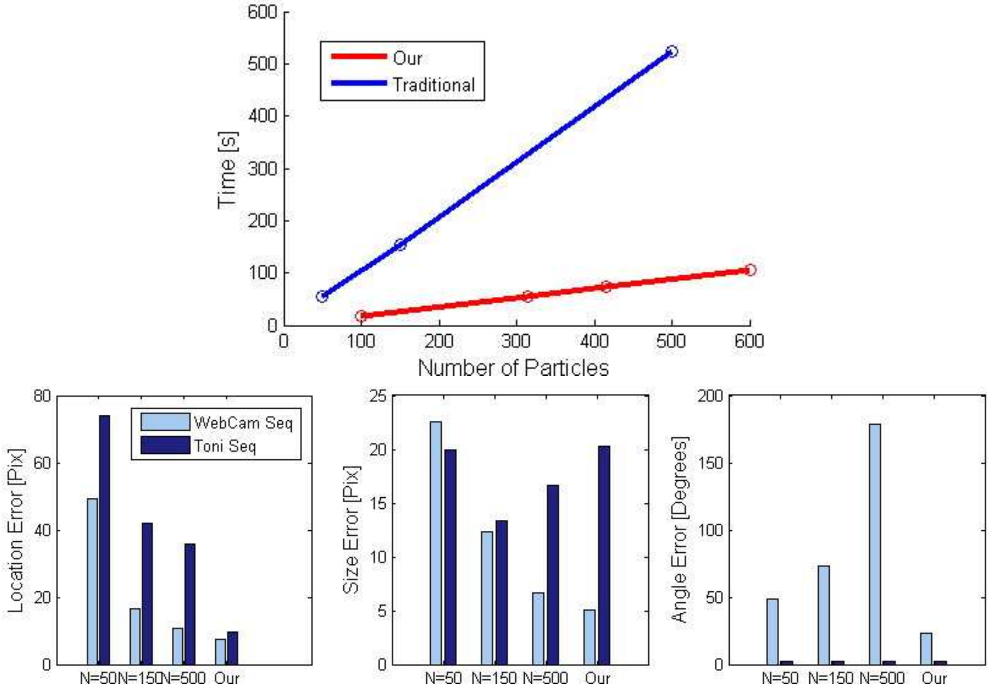


Figure 4.37: Comparison between traditional and our partitioned PF. Up: Time comparison. Down: Error in Location, size and angle in AVSS sequences

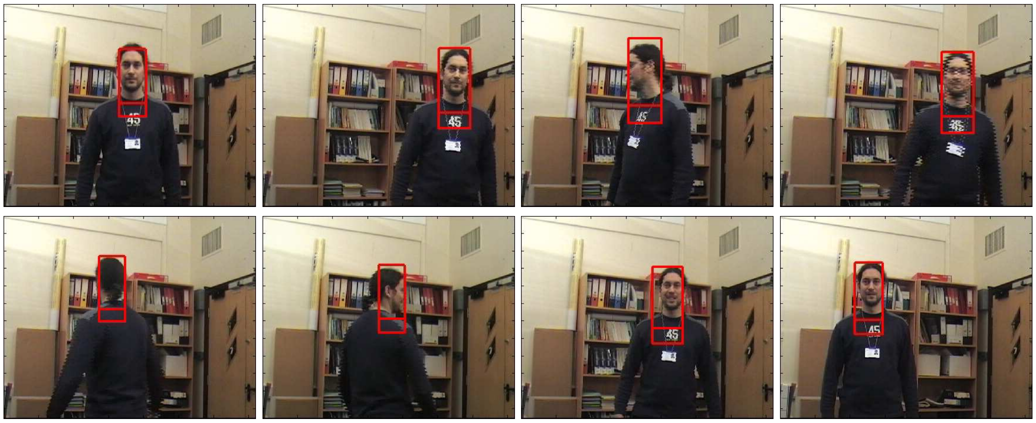


Figure 4.38: Toni sequence. Frames: 1, 44, 106, 185, 211, 221, 302, 428

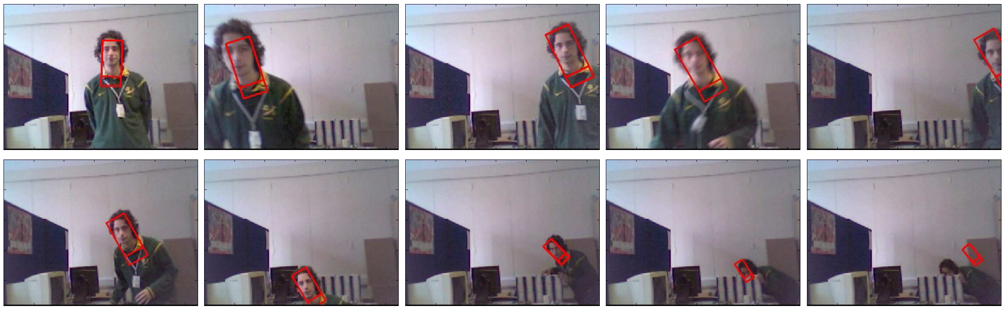


Figure 4.39: Webcam sequence. Frames: 2, 35, 50, 60, 88, 109, 133, 261, 284, 286.

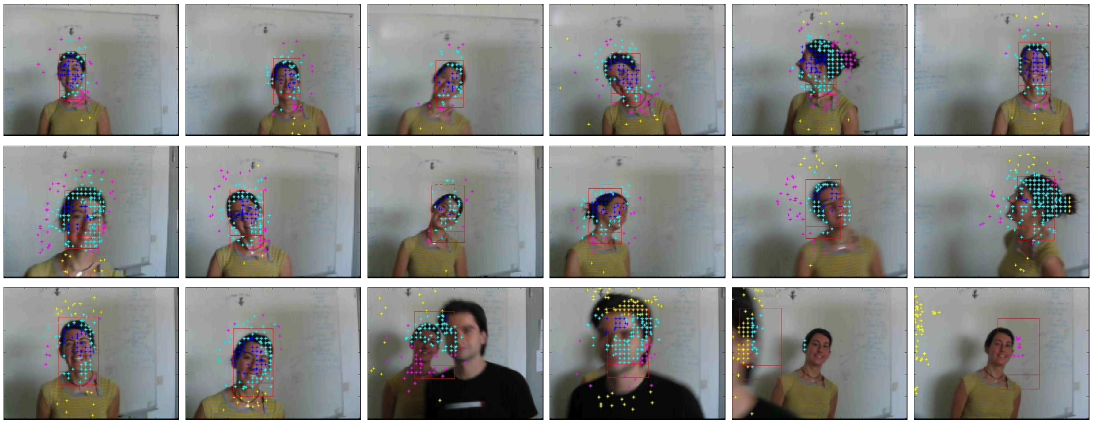


Figure 4.40: Moving face sequence processed using 2 partitioned stages. Location and scale are estimated.

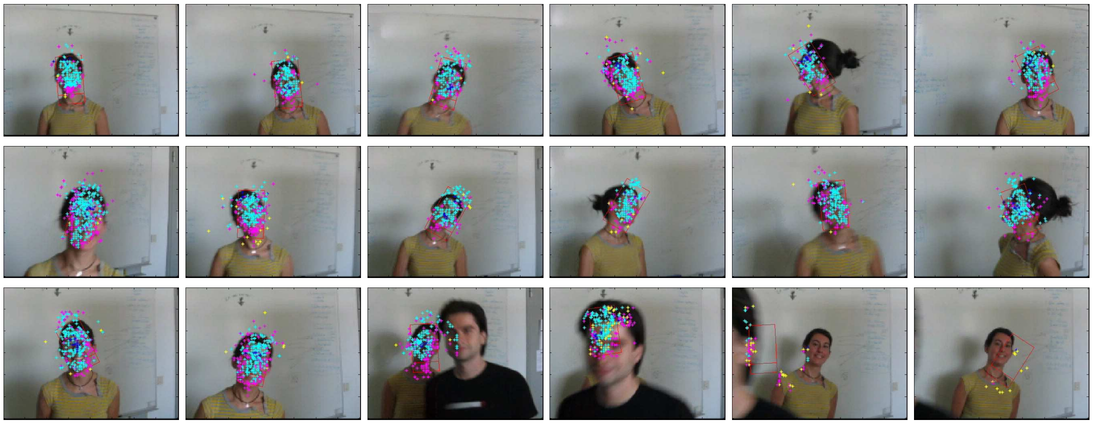


Figure 4.41: Moving face sequence processed using 3 partitioned stages. Location, angle and scale are estimated.

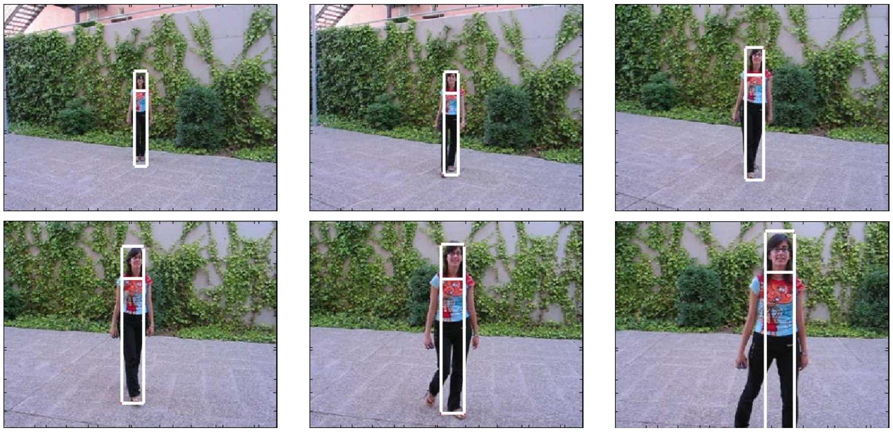


Figure 4.42: Walking sequence1 results.

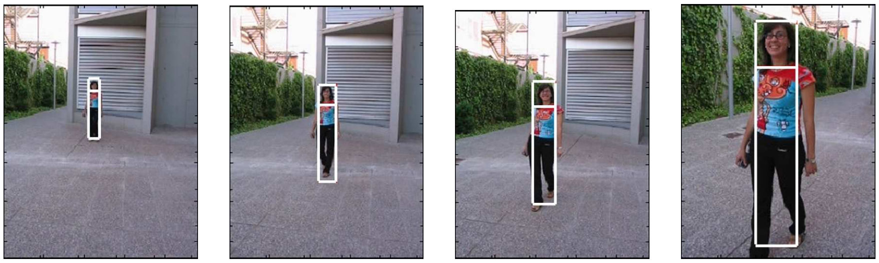


Figure 4.43: Walking sequence2 results.

4.6.3 Pedestrian Tracking

The same strategy shown in the previous section can be extended to cope with pedestrian tracking. The two region model allows us to keep the real size of the target even when it is partially occluded or out of the image. Results for two different sequences using moving camera are shown in Figures 4.42 and 4.43. Numerical result can be seen in Table 4.8 where is also shown a comparison between traditional approach and our proposal using different prior probability functions.

Table 4.8: Results obtained comparing our efficient PF with conventional colour-based particle filter (Walking sequences).

	N=150	N=500	N=1000	Density Mask	χ^2 Mask	Sub-Samp. Mask
MSE [pix]	14.566	12.427	13.208	8.4989	12.84	9.201
Mean N	150	500	1000	34.706	133.9	225
Time PDI [s]	-	-	-	33.602	2.564	3.19
Total Time [s]	76.12	253.75	507.5	66.479	38.71	47.13

Note that the MSE value obtained with 1000 particles is larger than the one obtained with 500 due to the fact that once a minimum number of particles has been launched, the benefits obtained by increasing this number are negligible. An unnecessary number of particles can even produce little errors in the estimation.

4.7 Conclusions

The work developed in this chapter can be divided into three blocks all of them oriented to region tracking domain:

- Feature extraction exploration.
- Efficient colour tracking based on particle filter.
- Colour model update.

We have made an overview to different methods to obtain a robust feature extraction in the first section of this chapter. While in punctual tracking, the measurement extraction consists in detecting possible candidates, being the main problem the presence of false positive and false negative, regional tracking requires to combine the measurements of several areas. For that, a probabilistic measurement process, in which non hard decisions are taken, helps to combine efficiently the measurement of each region and to keep the tracking of the target even though one or several regions are occluded. This characteristic makes the multiple hypothesis tracking more adequate for region tracking. Consequently, particle filter has been proposed for this tracking domain.

As score of the region tracking methodology, an efficient method for particle selection in tracking objects in complex scenes has been introduced. The approach improves the proposal distribution function of the tracking algorithm, including current observation, and reduces the computational cost by dismissing low likely particles. In addition, a partitioned sampling approach decomposes the dynamic state in several stages to manage high dimensional states at a reasonable cost. To represent the colour distribution, the appearance of the tracked objects is modelled by sampled pixels. Based on this representation, the probability of any observation is estimated by colour modelling techniques in the colour space. As a result, a Probability colour Density Image (PDI) is defined, where each pixel points its membership to the target colour model. Finally, the evaluation of all particles is accelerated by computing the likelihood $p(z|x)$ using the Integral Image of the PDI.

The main contribution is the use of successive partitioned estimations to drive the particle set towards high likelihood regions avoiding local maxima in the *a posteriori* probability distribution. Our approach has been focused on location, size and orientation estimation. Once the target probability density based on colour is detected and introduced as proposal distribution, the new set of particles is refined step-by-step through the partitioned dimensions. In order to speed up the observation process an integral image is generated from the colour probability image, allowing the use of rectangular masks for efficient weight computation. Thus, a computing cost reduction is achieved enabling the simultaneous tracking of a large number of trackers as it will be demonstrated in the following chapter. The contributions of this tracking framework are summarised as follows:

- A unified framework for improving sampling process based on Importance sampling and Partitioned sampling is proposed.
- A new proposal distribution by the generation of a Probability Distribution Image (*PDI*), where each pixel of this image indicates the probability of belonging to the target colour model, is introduced.
- By using the Integral Image the computation cost of the likelihood $p(z|x)$ evaluation is substantially reduced. In addition, a new proposal for the evaluation of rotated masks to dealt with different target orientations is set out.

Competitive results have been obtained even in the presence of distracters, occlusions or extremely hard quality conditions, such as zodiac sequences. It is specially worthy of remark the capacity of the framework to recover from failures in the tracking due to occlusions or camera shaking.

The presented tracker is based on colour, both parametric and non-parametric modelling, whose election will depend on the particularities of our application since both of them have their own advantages and drawbacks. Nevertheless, the methodology is not restricted to colour features but it can be extended to other features (gradients, texture, movement, etc.) by applying the adequate likelihood function. Moreover, following this partitioned philosophy it is easy to extend the methodology to estimate feature-based jointly with structure/shape-based parameters, as it will be shown in Chapter 6.

As priority future work, we should deal with the initialisation which has not been tackled in depth. Due to the different nature of the test sequences, required to demonstrate the generality of our tracking framework, it has not been possible to develop an automatic detection method for the first frame. Instead we have assumed the target location in the first frame as known, although no assumptions have been made about its appearance. Given this first location, the framework is able to extract automatically the initial appearance model. Nevertheless, an automatic detection algorithm, based on a previous training, could be useful for specific applications, such as [Okuma et al., 2004]. This methodology is only useful if the target appearance is known a priori, like in sport applications. Another approach consists in defining entrance/exit areas in the image. Thus, when an object is detected there, its model is capture and track. This second approach has not been used in this paper since is useless for moving cameras. Finally, as future improvement, simultaneous segmentation an recognition methods [Fei-Fei et al., 2005] could be applied to improve the results and extend the functionality.

Finally, in the third part we have also addressed a delicate but necessary topic as model updating is. The adaptation of the appearance model over time, simultaneously to the tracking, has received a minor interest in the past and has not been addressed properly. This happens in spite of the fact that feature-based tracking is a very popular topic. In Section 4.5, we have proposed a method which estimates jointly the location of the target as well as the appearance parameters. Rao-Blackwellised particle filter allows us to combine both estimations in a unique framework. This methodology increases the robustness of the updating process, avoiding a fast model degeneration due to inaccurate estimations. Parametric and non-parametric colour models have been tested, obtaining better results for the second ones, even in simple targets, like faces, with apparently unimodal distributions. The main contributions of this section can be summarised as follows:

- We have proposed a methodology that makes possible to track simultaneously location, shape and appearance parameters in a consistent, robust and integrated way. As paradigm, Rao-Blackwellised particle filter has been used.

- Instead of a classical PCA space to model the target's appearance, which limits the possible changes of the appearance, we have proposed a more general approach that allows tracking whatever appearance model provided it can be modelled as one or more Gaussians with mean and covariance. In this manner, the appearance model can change drastically and the algorithm still works.

- In addition, we have extended the methodology to non-parametric techniques by means of the usage of KDE techniques.

- We have introduced spatial information in the appearance updating process, without a prior layout model, for a double purpose, reducing the computational load and making the tracking robust against partial occlusions.

- The inclusion of PDA KF to update appearance parameters allows combining several valid measurements with a unique parameter at each time step. This modification is specially useful to update the appearance when the size of the target changes over time and and the number of pixels corresponding to each parameter of the appearance model varies.

In comparison with classical updating techniques, in which a slow update is not able to assimilate sudden changes and a fast update can fail due to non-accurate location and shape estimations (Figure 4.24), the method proposed is able to achieve a fast and robust update even under extremely changing illumination conditions. This is achieved even under the assumption of smooth changes thanks to the adequate selection of Q and R. As we discussed in Section 4.5.1.2, tested automatisation mechanisms for these parameters have not provided convincing results. As future work, we plan to obtain a better tuning mechanism to estimate automatically these noise parameters and evolve them over time according to the necessities and the environment.

5

Region Domain: Multi Target, Multi Sensor Tracking

"I shall try to correct errors when shown to be errors, and I shall adopt new views so fast as they shall appear to be true views."

-Abraham Lincoln-

In the previous chapter, a region tracking framework was introduced, which applies sampling techniques to improve the prior probability and the calculation of the posterior likelihood. Although the efficiency gain obtained is always welcome, it results especially useful when it can be extrapolated for simultaneous and concurrent processes. One of the main concerns on tracking applications is the treatment of occlusions and distracters. Our framework has proved robust against static occlusions and clutter. The next step consists in tackling the other big challenge on tracking, the multi-target tracking and the dynamic occlusions.

Following the aim of increasing the efficiency of the system, we are going to show the possibilities that our framework provides. Particularly, we cope with multiple tracking of identical targets. That means that the appearance model is exactly the same for all the targets. In this manner, the advantages that our framework provides are enhanced: since the same model is used for several targets, just one probabilistic density image and one integral image are enough for the simultaneous tracking of all of them. On the other hand, indistinguishable targets involve a harder problem due to the fact that no appearance clue can be used to solve the ambiguities, and once the confusion has happened, there is no way to guarantee a correct recovering of the lost tracker.

In order to solve multiple target problems, two approaches have been raised. The first approach is an extension of the multi-layer methodology which allows introducing interaction between independent targets. The second one is the implementation of multi sensor systems to solve the ambiguities that appear due to the multi-target coalescence.

In this chapter, the previously proposed methodology is extended to deal with muti-target tracking. In addition, an incursion in multi sensor tracking is done. Finally, an example of application is presented. One of the best test bench for the multi target proposal is the sports field. Sports sequences mean multiple interacting targets with identical appearances and lots of complex situations. In particular, we work with football sequences as a classic example where multiple target problems appear and as result of a collaboration project with a local football team.

5.1 State of the Art

5.1.1 Multiple Target Tracking

Given the difficulties that multiple target tracking involves, several techniques have been proposed in the literature [Bar-Shalom and Li, 1993; Blackman and Popoli, 1999; Isard and MacCormick, 2001]. Data association between observations and trackers is the problem to be solved, being the coalescence (meaning the tracker associates more than one trajectories to some targets while loses track for others) the most challenging difficulty, especially when similar targets move close or present occlusions. Moreover, cluttered scenarios produce false alarms which introduce confusion in the association algorithm. Fortunately, although in general multi target tracking deals with state estimation of a variable number of targets, assumptions about a constant or known number of targets can be used to constraint the problem. For instance, depending on the application, only one measurement is associated for each target or, on the contrary, several returns are possible candidates. These assumptions will reflect on the data association method.

The simplest solution is the nearest neighbour (NN or NNSF) [Bar-Shalom and Li, 1993; Zhang et al., 1996] in which only the closest measurement is associated to each state. However, this technique has been shown poor to solve complex situations because its performance depends completely on a correct prediction. Probabilistic multiple target tracking techniques try to cope with this problem by performing a global optimisation minimising the statistical distance. The most general technique is the multi hypothesis tracking (MHT) algorithm [Polat et al., 2003], which calculates every possible update hypothesis for each target. Obviously, this fact implies that it is a high time-consuming method. To solve it, a modification of MHT called probabilistic MHT (PMHT) employs a maximum-likelihood method in combination with the expectation maximization (EM) method to reduce the number of hypotheses. Another popular association method is the joint probability data association (JPDA) [Bar-Shalom and Li, 1993]. It estimates the location of each target by a sum over all the association hypothesis weighted by the probabilities from the likelihood. A comparison between the JPDAF and the PMHT is made in [Rago et al., 1995].

Although these algorithms do begin to address the problem of tracking multiple targets, both are prone to error specially in situations involving significant clutter or targets in close proximity. In those situations, it is required to take into account the distribution and location of each target relative to the others. Two examples which try to cope with this problem are MCMC particle filter and Mean-field approaches. Both approaches essentially enforce the rule that more than one target can not occupy the same space at the same time. Markov Chain Monte Carlo (MCMC) [Bergman and Doucet, 2000] method models explicitly the interaction of targets by removing measurements which fall within a radius of other target predictions.

In [Khan et al., 2005] a MCMC particle filter addresses the problem of multiple interacting targets by means of a Markov Random Field (MRF) motion prior that helps to maintain the identity of targets throughout an interaction. The MCMC sampling step shows its potential to produce an efficient filtering. An extension of this paper is presented in [Khan et al., 2006], where an auxiliary variable sampler and a Rao-Blackwellized Markov Chain are implemented to deal with complex target distribution, like merged measurements or more than a measurement per target. Numerical methods allow their computation in real time. Lanz [Lanz, 2006] proposes a Hybrid Joint-Separable (HJS) filter derived from a joint Bayesian formulation of the problem. An optimal representation is obtained by means of Belief Propagation (BP). Finally, a MRF approximation permits to joint dynamics efficiently and an appearance likelihood implements a physically-based model of the occlusion process.

Many other proposals have been discussed, being most of them a variation or a combination of several basic solutions. Thus, Karlson and Gustafson [Karlsson and Gustafsson, 2001] introduce a Bayesian data association method based on particle filter and JPDA hypothesis calculations. Chen et al. [Chen et al., 2001] combine JPDAF with Hidden Markov Models (HMM). While HMM provides a powerful framework to incorporate multiple cues (e.g., edge intensity, foreground region colour and background region colour) by expanding its observation, JPDAF computes the HMM's transition probabilities, taking into account the intercorrelated neighboring measurements. In [Hwang et al., 2003], Hwang et al. present the integration of JPDAF with the Identity Management (IM) algorithm to associate measurements and keep the correct identity of the target using a sensor network. Yu et al. [Yu et al., 2006] address the data association problem by formulating the visual tracking as finding the best partition of a measurement graph containing all detected moving regions. The likelihood is computed using an Adaboost detector and a MCMC method avoids the exponential complexity by sampling the space efficiently.

In order to tackle with the computational cost, Yu and Wu [Yu and Wu, 2004] propose a mean field approximation with a linear complexity, instead the combinatorial complexity of the JPDA-based methods. The proposal introduces a collaborative inference mechanism among a set of low dimensional particle filters, in which the estimate of each target is not only determined by its own observation and dynamics, but also through the interaction with its adjacent targets. This leads to a competition mechanism that enables different targets to compete for the same observations. In [Qu et al., 2005], an analogy with physical forces is used. The merging problem is considered as the effect of an attractive "gravitational force" between observations. To resist the excessive attractions when two targets are close together, a repulsive "magnetic force" is introduced. The history of motion information is also included to solve the labelling problem. In [Dowdall et al., 2007], Dowdall et al. propose a network of independent trackers whose interactions are modelled using game theory.

Nowadays, recent researches try to introduce social concepts into the tracking methodology in order to address the multiple target issue. Even though multi-target tracking generates new problems, it also allows that new information could be included in the global solution. In this way, the presence of other targets helps to estimate those targets with poor likelihood. A clear example is [French et al., 2007]. This paper proposes to improve the tracking of multiple targets in complex scenes by incorporating social motion information into tracking predictions. This is achieved by allowing a tracker to share motion estimates within groups of targets which have been moving previously in a coordinated fashion. The implicit formation of social groups aids the estimates of all the members. Going a step forward in this direction leads to crowd tracking domain.

5.1.1.1 Crowd tracking

The majority of papers in the literature deals with a single person or reduces the amount of people with controlled interactions. Recently, more progresses have been made in this field, as it has been shown in the previous section. Nevertheless, a limitation must be taken into account: under congested situation the problem becomes intractable in monocular sequences and it may require multi-camera systems [Khan and Shah, 2006].

Dong et al. [Dong et al., 2007] present a fast method for estimating the number of humans and their positions from background differenced images obtained from a single camera where inter-human occlusions are significant. They propose a novel example-based algorithm which maps the global shape feature by Fourier descriptors to various configurations of humans directly. By using locally weighted averaging, the best possible candidate configuration is interpolated and extracted. The inherent ambiguity resulting from the lack of depth and layer information in the background difference images is mitigated by the use of dynamic programming, which finds the trajectory in state space that best explains the evolution of the projected shapes.

Maurin et al. use optical flow to track crowd movements both day and night around a sports arena [Maurin et al., 2002]. Haritaoglu et al. track groups as well as individuals by developing different models of pedestrian actions [Haritaoglu et al., 1999]. They attempt to identify individuals among groups by segmenting the heads of people in the group blob.

This kind of crowd analysis can not be considered as punctual or regional tracking, being itself a proper category in our tracking scheme.Nevertheless, this kind of tracking falls out of the scope of this thesis. Although we are introducing it in the regional tracking category, specifically in the multi-tracking chapter, it could compose an extra level of understanding by itself, due to its importance and the specific methodology that it requires. These techniques, closer to fluid mechanics than to conventional tracking algorithm, would need a specific research in that area of knowledge.

5.1.1.2 Football applications

As we have mentioned in the introduction, sports applications are a clear example to show the problems of multi-tracking and the potentiality of our proposals. As consequence, collaborative sports sequences have been used to check the validity of the algorithms. Numerous publications about multi-target tracking applied to sports analysis exist, with many different applications such as highlight extraction, tactic analysis or improvement of the athletic performance.

Highlight extraction is a useful application for TV broadcasting and automatic labelling of recorded video database. Working with a dynamic and uncalibrated camera is an essential requirement. [Ekin et al., 2003; Assfalg et al., 2003; Matsumoto et al., 2000] stand out in this field of application.

In [Ekin et al., 2003], important events such as goals are automatically detected on the basis of camera shot change, referee location or the field-of-view area. Colour segmentation algorithms are applied to detect the predominant colour or the lines which delimit the field. Changes of camera shot and horizontal and vertical histograms are used to detect important events and the referee respectively. In the same line it is the work [Assfalg et al., 2003], which proposes to index a football match using a single uncalibrated moving camera. Different activities or situations are modelled using state machines, whose transitions are conditioned to a set of simple clues such as the zone of the football field where the ball and the players are. Template matching is used to detect the football players and homographic techniques allow locating them on the field. Matsumoto et al. [Matsumoto et al., 2000] present a multi camera system that is able to find automatically the camera with a better viewpoint of the action. For that, the ball is detected by applying template matching and using the fundamental matrices. The cameras where the ball is occluded or too far from the camera are discarded. The best camera is that in which the trajectory of the ball produces a larger and more straight line.

Nevertheless, most papers are focused on tracking players, like a classical multi-target problem, either working with mobile cameras or with multiple cameras. Particle filter has been shown to be the most adequate tracking algorithm due to the advantages it has to manage multiple targets. However, maintaining multi-modality is not an easy task, and tracking failures appear indefectibly. In spite of the fact that particle filter is good maintaining the multimodality for short periods of time, like for example, in self-occlusions or short player collisions, it has obvious problems to maintain this multi-modality over time. Since these long occlusions are quite common in sports analysis, in which players move in pairs or even in small groups, some mechanisms must be applied to solve them. Moreover, this effect is emphasised due to the fact that two players belonging to the same team are virtually identical.

To deal with this problem [Choi and Seo, 2004a], the authors create a synthesised image of the occlusion out of template images saved previously. Particle filter is used as tracking paradigm. Once the background pixels have been segmented by colour, an histogram comparison is employed to assign weights to each particle. However, occlusions between players of the same team can produce labelling errors or even the loss of one of the players, assigning the same measurement to more than one tracker. The proposed solution is to generate a synthetic image of the occlusion combining patches of the appearance of the players before the occlusion and moving them to the predicted location. An "occluded" colour model is extracted from this image and used as reference to obtain the likelihood. In [Vermaak et al., 2003a], authors apply a mixture of Gaussians which maintains the multi-modality. The mixture of Gaussians forces particles to distribute in clusters even when the probability of one of the clusters is much smaller than the others. The system can be assimilated as a mixture of particle filters which interact to calculate jointly the weights of the hypotheses.

An opposite methodology to keep the multimodality is based on differentiating the target to be tracked as much as possible. By generating human models with different techniques, such as neural networks [Okuma et al., 2004] or PDMs [Hayet et al., 2005][Needham and Boyle, 2001], results are improved with regard to the traditional colour likelihood. Okuma et al.[Okuma et al., 2004; Okuma et al., 2003] combine two successful techniques like mixture of particle filters, which assigns a component of the mixture to each player improving in that way the multi-target tracking, and Adaboost, which generates a good proposal distribution capable of managing objects leaving and entering the scene.

Besides the probabilistic solutions, the first and more effective approach consists in the use of multiple sensors, in this case, cameras. Advantages provided by multi-camera systems are evident: better predictions due to a more real and accurate dynamic model, robustness against occlusions, improvement of the accuracy due to the multiple measurement conjugation, 3D estimation of the targets, etc... However, multiple cameras are not always an available solution. Hardware requirements, minimisation of costs or use of preinstalled equipments can limit the availability of resources.

Tracking players in a single mobile camera can operate during short sequences, but requires complex procedures (entry and exit of player, consistent labels) during a complete match. Furthermore, this motion dynamic in the image is affected by the perspective effect. For this reason, a shared reference, like a plan, is commonly used to simplify the problem. Examples are shown in [Yamada et al., 2002; Okuma et al., 2003], where a dynamic calibration is recalculated at each frame. Paper [Yamada et al., 2002] tries to track players and ball. The correspondence between the image and the plan is established dynamically detecting the lines of the field. Once the lines have been obtained using colour, each point receives a mark with the distance to the nearest line point of the plan. Sweeping all the horizontal and vertical angles and the focal distance, the best matching is selected and the homographic matrix is calculated. The system is able to cope with a pan/tilt/zoom camera. T-shirts and shorts of players and referee are detected using colour and a matching based on distance is done between both pieces of the uniform as well as between consecutive frames. Authors assure that a simple linear extrapolation, instead of a tracking algorithm, is enough to track all the players thanks to the homographic mechanism at a frame rate of 30 fps.

A more widespread and successful option consists in the use of several static cameras with a shared reference. This approach has become a de facto standard. An exhaustive approach of the whole problem is tackled in [Xu et al., 2004; Xu et al., 2005], which have been written in the framework of the INMOVE project. A double tracking strategy (image and plane) is employed to track all the players. A multi-camera system detects player candidates in the images using colour and motion, by applying a parametric colour model based on mixture of Gaussians, and background substraction. The correspondences with the measurements corresponding to the same player are matched by a multi-camera tracking algorithm, once they have been translated to the plan as shared reference. In addition, an auxiliary tracker for each player in each camera is running. The employed tracking paradigm is based on Kalman filter with a rectangular model and a first order dynamic model. Another interesting example based on similar principles can be seen in [Sullivan and Carlsson, 2006].

Football is a collaborative sport and the ball is an essential element of the game. However, its tracking is really complex because the movements are three-dimensional, in contrast with the movement of the players, who are always in the same plane. Furthermore, the small size of the ball, in conjunction with the high speed that it can suffer, as well as the multiple occlusion with players, convert the ball tracking in one of the most challenging goals. Several papers tackle with this problem [Ren et al., 2004; Choi and Seo, 2004b; Choi and Seo, 2004a]. In [Choi and Seo, 2004a; Choi and Seo, 2004b] colour and gradients are combined to remove false positives such as lines or pieces of players. Candidates are accumulated over time to generate a trajectory. Those candidates with non coherent trajectories are considered noise and removed. That blob labelled as ball is tracked and, when it disappears, it is assumed in possession to the nearest player. When the ball is in the hands of a player, and meanwhile the ball is not detected again, both player and ball tracking are combined and considered a special case. Yamada et al. [Yamada et al., 2002] use a physical model of the ball to determine accurately its movement. Gravity and friction coefficient are taken into account. Finally, Ren et al. [Ren et al., 2004] tackle this issue dividing it into three fundamental aspects: false positive, ball tracking with lost observations and 3D estimation mono or multi-camera. The ball is classified in three states (rolling on the floor, in the air or out of the field) and it receives a different processing depending on the state in which is classified. When the ball is rolling, it is tracked by a simple Kalman filter and distracters are removing with a temporal filter. To detect a flying ball, a triangularisation combining ball measurements from multiple cameras is made. Once the ball is detected, it is projected on the floor. If only one camera is available, the best fitting between parabolic or rectilinear trajectories determines the location on the floor or in the air. Good results for short sequences have been obtained.

This particular aspect of the sports tracking is not tackle in this chapter. Ball tracking is not considered because it is outside the scope of this thesis. However it will be contemplated in future works.

5.2 Multiple Targets

In the light of previous researches, it should be noted that particle filter has serious problems to track multiple targets concurrently for long periods of time. Although one of the particle filter advantages is, theoretically, the capability for modelling and tracking multi-modality, it has been demonstrated [Vermaak et al., 2003a] that this capacity is limited and it is not consistent enough when there is ambiguity or presence of multiple targets.

Traditionally, models of multiple targets have been tracked simultaneously by concatenating their parameters in just one state vector. Therefore, several clusters corresponding to multiple targets are sampled simultaneously. However, the finite particle set is prone to overpopulate the most probable cluster, producing the loss of the other target clusters if this situation carries on for several frames. This undesirable effect increases specially if clusters are near or collide due to an occlusion. Furthermore, the philosophy of particle filter consists in fairly estimate the clusters in order to have and efficient tracking. This facilitates the probability of confusion during an occlusion.

Once any cluster has been lost, it is unlikely to recover it. This fact happens because the Monte Carlo approximation only guarantees the convergence with an infinite number of samples. In theory, the stochastic term of the motion model produces some non-zero probability everywhere, so even extremely unexpected motions could be tracked. Nevertheless, in practice, the finite nature of samples produces that all of the samples are concentrated near the most likely areas, without samples at all in the rest of the state vector space.

Although increasing artificially the state noise and augmented the particle set can mitigate the problem, by populating empty areas and obtaining a better estimation of the target density, it does not solve it at all. Only efficient sampling techniques with explicit treatment to maintain multi-modality or model interaction between cluster have been proved effective.

5.2.1 Importance Sampling

We tackle this last drawback by means of the importance sampling and the PDI shown in the previous chapter. An efficient sampling guided by the measurement allows placing particles in likely areas in spite of an incorrect motion model and noise parameter. This modification permits to recover from a failure in a multi-target tracking process.

Moreover, the special character of our multi-target approach, in which all the targets have the same appearance model, emphasises the advantages of the proposed methodology. The previous chapter showed the computational cost reduction of evaluating each particle at the expense of having a fixed cost to calculate the PDI and the integral image. This fixed cost is negligible even with a small number of particles. Since all the targets have the same model, both images are reused for every target, increasing the efficiency even more.

5.2.2 Maintaining the Multimodality

Although importance sampling provides an important advantage, it does not ensure to keep clusters separated. To achieve that, a clustering should be implemented either in the sampling or in the weighting process. In this manner, particles would force the target to split each other. Although introducing clustering in the joint sampling process of all the targets has been shown successful to maintain the multi-modality, it has a non-negligible drawback. The traditional joint filter approaches suffer an exponential complexity increase with the number of tracked targets. This makes unusable the joint particle filter for many targets due to the huge number of required particles. If a small number of particles are used in those conditions the tracking quality is low and the number of errors reported is very high.

On the contrary, independent filters for each individual target lead only a linear grow in the computation, but it produces coalescence among targets. For those reasons, and following the efficiency optimisation philosophy proposed in the previous chapter, an approach based on running independent particle filters for tracking each target is presented. Once a initial estimation has been done, we will introduce the interaction explicitly in the hypothesis evaluation stage in order to solve the coalescence.

Let us make a digression in this point. MCMC sampling techniques have shown efficient and successful. By combining this sampling technique with interaction functions like in [Khan et al., 2005], this approach has the appealing property that the filter behaves as a set of individual particle filters when the targets are not interacting, but efficiently deals with complicated interactions as targets approach each other. However, we have adopted the independent filters philosophy due to four factors:

- Although MCMC particle filter has a similar efficiency when targets do not interact, the cost for each target is larger than in the independent filter case and more samples per tracker are needed. This fact has been tested in a previous work [Medrano et al., 2008].

- MCMC sampling has the powerful characteristic of being general and able to model any target distribution. However, when targets are independent, we know accurately the function which fits every target. This knowledge is easily included in the independent filters but must be discard in MCMC-PF for the sake of generality. Moreover, the generality forces to a careful design of the proposal density, which plays an important role in the success of an MCMC algorithm and it is not a easy task in most of occasions.

- Following the incremental learning methodology that this thesis has meant, that is, including the interaction as a function in the weighting stage, allows the integration of all the previously developed techniques and the possible inclusions of new ones. On the contrary, the use of a well-known methodology limited the novelty and the trial-error process that the scientific method implies.

- If the number of targets changes over time, it can cause critical problems when the total number keeps increasing. New tracks have to steal particles from existing trackers so that the number of remaining particles for each target decreases until it is insufficient to correctly approximate the distribution. Independent filters simplify the problem since new tracks are created by assigning a new particle set for each of them rather than stealing particles from existing targets. Therefore, the tracking of the new targets is achieved without affecting the approximation accuracy of the existing tracks.

Please, be aware that we do not want to say that our approach is better than MCMC under any conditions. Nevertheless, we propose another alternative that ensures better efficiency and high possibilities of development and improvement. These assertions will be shown in the next sections as well as in the results.

5.2.2.1 Iterative methodology

The proposal consists of two basic steps: the first one detects all the possible hypotheses for each target, and the second one weighs them on the basis of the joint probability and selects the most adequate for each case. The procedure for including the coalescence in the independent particle filters is indicated in the Algorithm 7.

The high difficulty that tracking identical target implies is the virtually identical appearance of all of them. In these conditions, only dynamics can be used to distinguish the targets. Given this premise, the more recent and accurate information we use, the better results we obtain. For this reason, we propose to take into account the interaction after obtaining a first estimation, instead of applying it to the predicted parameters or even to the previous frame locations.

In order to obtain this first estimation, which ensures a more reliable initialisation of the interactive function, an iterative refinement of the posterior probability is implemented. The methodology is based on dividing the evaluation step in a set of layers. This multi-layer particle filter allows introducing a refinement, where the first estimations helps to discard some hypotheses before costly evaluations should be done. In this way, independent observation can be combined sequentially to give the final estimation.

This layered particle filter has a similar purpose that the annealed particle filter described in [Deutscher et al., 2000]. However the methodology is different. Meanwhile annealed particle filter uses the same measurement through the different layers, our proposal introduces a new factor that was not present in the first estimation and whose value changes with the posterior estimation.

Therefore, by computing the first layer, a first estimation of the state vector of each target can be extracted. This estimation permits to compute the interaction function, whose result is included in the following layers to refine the parameters and model the coalescence.

Algorithm 7: The modified particle filter with an extension to improve the tracking of several objects

```
Given a particle set $\{x^i_{j,t-1}, \omega^i_{j,t-1}\}^N_{i=1}$ which represents the posterior probability
$p(x_{j,t-1}|z_{t-1})$ at time $t-1$ for $j = 1, ..., M$ targets.

  1. Initialisation: $k \leftarrow 1$

     for $j = 1$ to $M$

       • Generate $N_1$ new samples from the importance function in the main partitioned
         dimensions. $x^i_{j,t} \sim g(x_{j,t})$.
       • Propagate $N_2$ samples from the old samples applying dynamic $x^i_{j,t} \sim p(x_{j,t}|x_{j,t-1})$.
       • Weigh the particles
         $\omega^i_{j,t,k} \propto \begin{cases} p(z_t|x^i_{j,t}) \cdot f_t(x^i_{j,t})/g_t(x^i_{j,t}) & i = 1, ..., N_1 \\ \omega^i_{j,t-1} \cdot p(z_t|x^i_{j,t}) & i = N_1 + 1, ..., N \end{cases}$
         and normalize them $\sum^N_{i=1} \omega^i_{j,t} = 1$.

     end

  2. Iteration:

       • Clusterise the target density defined by $\{x^i_{j,t}, \omega^i_{j,t,k}\}_{i,j}$.
       • Assign the correspondences between clusters and targets using the associating
         algorithm and calculation of $E[x_{l,t}]$.
       • Re-weigh: $\omega^i_{j,t,k+1} \propto \omega^i_{j,t,k} \cdot \Pi_{l \in [1..M], l \neq j} \phi(x^i_{j,t}, E[x_{l,t}])$
       • $k \leftarrow k + 1$. Iterate until convergence.

  3. Estimation:

     for $j = 1$ to $M$

       • Assign the final weights: $\{x^i_{j,t}, \omega^i_{j,t}\}^N_{i=1} \leftarrow \{x^i_{j,t}, \omega^i_{j,t,k}\}^N_{i=1}$.
       • Estimate the new position of the state
         $E[x_{j,t}] = \sum^N_{i=1} \omega^i_{j,t} \cdot x^i_{j,t}$
       • Resample $\{x^i_{j,t}, \omega^i_{j,t}\}^N_{i=1}$ to get $\{x^i_{j,t}, \frac{1}{N}\}^N_{i=1}$

     end
```

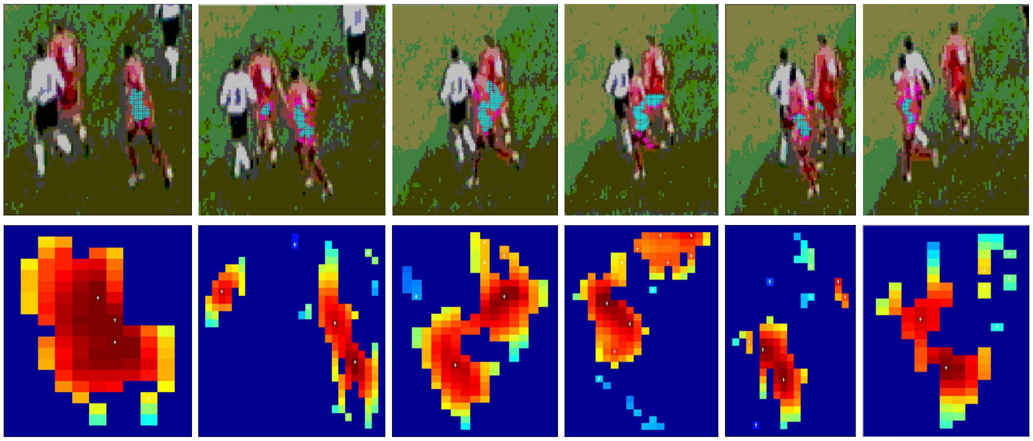


Figure 5.1: First row: Distribution of particles on the image for a short interaction sequence. Bottom row: Continuous map and clusterised candidates (white cross)

The algorithm shown in Algorithm 7 is a general version which allows obtaining stable target locations. However, experiments have shown than one step is enough to obtain accurate estimations without a large computational load. Just in cases where more than 4 targets interact simultaneously it is recommended to include an extra layer for a second interaction step.

5.2.2.2 Clusterisation

The first step consists in detecting local maxima over a map composed by the sample weights. The methodology is similar to the process used in the kernel density estimation technique to transform a discrete and sampled function, like a histogram, in a continuous function.

Once the sampled distribution is reconverted in a continuous map, local maxima are extracted and labelled as candidates. Each one is a possible position for the tracked object, other similar object or a distracter.

5.2.2.3 Association algorithm

Once clusters have been segmented, the association between candidates and targets must be done. We assign weights to the local maxima and select the most satisfactory, that is, we try to associate measurements to tracked objects and classify the rest of them as distracters. This weighting has been made using different methodologies: nearest neighbour, multiple hypothesis tracking algorithm (MHT) [Polat et al., 2003], based on trajectory analysis, and the auction algorithm based on association techniques. A comparison between the different options is shown in Table 5.2.

Nearest neighbour This is the simplest technique [Zhang et al., 1996]. It assigns correspondences taking into account only the distance between pairs. However, several measurements can be associated with the same target, or several targets can catch the same measurement. To avoid these undesirable behaviours, more sophisticated algorithms have been developed in the literature, such as the auction algorithm.

Auction algorithm Auction algorithm is a well-known association technique employed to complex situations, where the presence of observation in excess (distracters) or in absence (bad

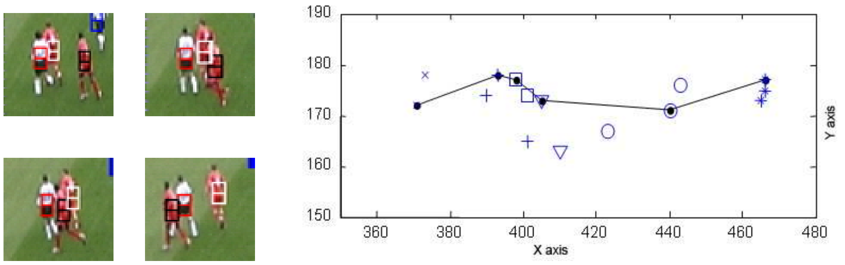


Figure 5.2: Left: tracking of two identical objects using MHT selector. Right: trajectory of the player labelled with a black rectangle, and distracters due to the other players and spurious

detection) produce failures in conventional techniques. It can be used to solve tracking problems by assigning the detected measurements with the established tracks presented in the previous frame. As the number of targets can be variable through the image sequence and distracters can appear in the image, the number of tracks and objects might not be equal. So, an asymmetric assignment problem has to be solved.

The auction algorithm is an iterative process that works similarly to a real auction. It is able to handle symmetric and asymmetric assignment problems. The tracks "bid" for the objects, slowly raising the cost of the objects. A desirable object will get many bids and thereby ending up with a high price while other less attractive objects will have a lower price. The full description of the method is given in [Althoff et al., 2005].

Multiple hypothesis tracking MHT is suitable for solving any data association uncertainty while tracking multiple objects. Assuming the fact that the motion between two frames can not change instantaneously due to inertia and image acquisition rates, a path coherence function Ψ can be defined as follow:

$$\Psi(X_{t-1}, X_t, X_{t+1}) = w_1 \cdot (1-a) + w_2 \cdot (1-b) \tag{5.1}$$

with

$$a = \frac{\overline{X_{t-1}X_t} \cdot \overline{X_tX_{t+1}}}{\|\overline{X_{t-1}X_t}\| \cdot \|\overline{X_tX_{t+1}}\|} \qquad b = \frac{2[\|\overline{X_{t-1}X_t}\| \cdot \|\overline{X_tX_{t+1}}\|]^{1/2}}{\|\overline{X_{t-1}X_t}\| \cdot \|\overline{X_tX_{t+1}}\|}$$

where X_t and X_{t-1} are the position of the tracker in the instant of time t and $t-1$, X_{t+1} is the location of the possible candidates, $\overline{X_{t-1}X_t}$ is the vector between first and second points, and similarly $\overline{X_tX_{t+1}}$ is the vector between second and third. Finally, w_1, w_2 are parameters to assign different importance to the direction and speed information, such that $w_1 + w_2 = 1$.

Hence, we apply this path function for giving a weight to the local maxima, that is, to all the possible candidates including other players and distracters. We choose that combination which minimises the global result of Ψ for all the targets at each time step. An example of application is shown in Figure 5.2.

To solve total occlusions between objects with similar histograms, in which both instances are engaged to the same measurement, we force the system, if it is possible, to select those combinations which do not produce an occlusion, even though they have a worse value. In this manner, once the occlusion has finished, the system must decide which tracker will get the free measurement if there is one available.

5.2.2.4 Interaction function

After associating each target to its corresponding cluster, an interaction function is applied to re-weigh the particles. Many options to define this function are possible. Since the association has been done, it is logical to think that this function should be based on a distance to the chosen location.

To obtain a soft transition, we can apply a function as

$$\phi(x_{j,t}, x_{l,t}) = exp(-g(x_{j,t}, x_{l,t})) \tag{5.2}$$

where $g(x_{j,t}, x_{l,t})$ is a penalty function, for instance, the Euclidean distance.

However, a bad choice of this function could look down the value of the observation in case of occlusion, being the repulsion force of the interaction function the relevant factor to assign a weight to each hypothesis. Given that an agreement between both factor should be the most adequate solution, we apply a binary factor based on this penalty function:

$$\phi(x_{j,t}, x_{l,t}) = \begin{cases} 1 \quad if & g(x_{j,t}, x_{l,t}) < \varepsilon \\ 0 & otherwise \end{cases} \tag{5.3}$$

In this way, the observation is considered to estimate the parameters of the tracker once the effects of the distracters have been removed.

5.2.2.5 Mono target application

This multi target algorithm can also be applied to track only one target in complex sequences. When many distracters appear in the image, or clutter in the background can produce confusions, particles tend to distribute in several modes that steal particles from the target, distorting the estimation and finally producing that the filter diverges.

We can apply the same algorithm proposed for multi target tracking with minor modifications in order to discard those distracters. Following Algorithm 7, once the target density sampled by the particles has been clusterised, the evaluation of each hypothesis is done as

$$\omega^i_{t,k+1} \propto \omega^i_{t,k} \cdot \Pi_c \phi(x^i_t, x^*_{c,t}) \tag{5.4}$$

where x^* is the set of c distracters detected by the cluster algorithm.

The effect of this modification works correctly even in presence of a large amount of distracters, as it is depicted in Figure 5.3.

However, due to the large amount of distracters that can appear, it may be no practical to compute the interaction of the target with all the distracters. Instead, a similar result can be obtain comparing all the hypotheses with the last location of the target. For that, factor ϕ is introduced in the weighting process as follows:

$$\omega^i_t \propto \omega^i_t / \phi(x^i_t, E[x_{t-1}]) \tag{5.5}$$

In this way, the sampling process is guided by the last location of the target. It presents some advantages with respect to using only the last location, like in Kalman filter or some guided sampling techniques. While those sampling techniques guide the distribution of particles, with the risk of loosing completely the target in the event that this prior information would be incorrect, our proposal discards the particles only a posteriori, i.e. once it is probable that the hypotheses near the expected location are valid.

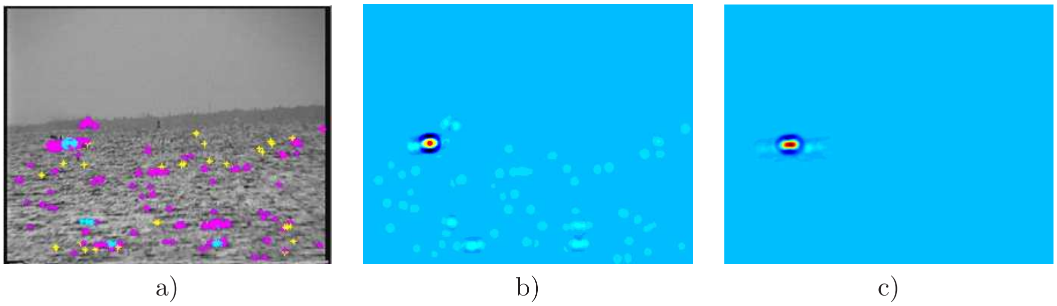

a) b) c)

Figure 5.3: Distracter removal by applying multi-target algorithm to complex mono-target sequences. a) Original frame with particle distribution. b) Continuous map of candidates c) Resulting map after applying the interaction factor

5.2.3 Experimental Results

The performance of the algorithm to deal with multiple targets is checked using *Ant* sequence, (ftp://ftp.cc.gatech.edu/pub/groups/borg/pami05), Fig. 5.4, and sports sequences, Fig. 5.5 (ftp://ftp.pets.rdg.ac.uk/pub/VS-PETS) and 5.6 (http://www.cs.ubc.ca/∼okumak/research.html).

Efficient particle filter presented in Chapter 4 has been combined with the multi-target extension to solve the test sequences. MHT has been chosen instead of auction algorithm for its superior performance, as it is shown in Table 5.2. Moreover, the flexibility of our algorithm allows that other multi-tracking methodologies can be combined with our proposal [Okuma et al., 2004; MacCormick and Blake, 2000; Vermaak et al., 2003a; Khan et al., 2004; Bar-Shalom and Li, 1993].

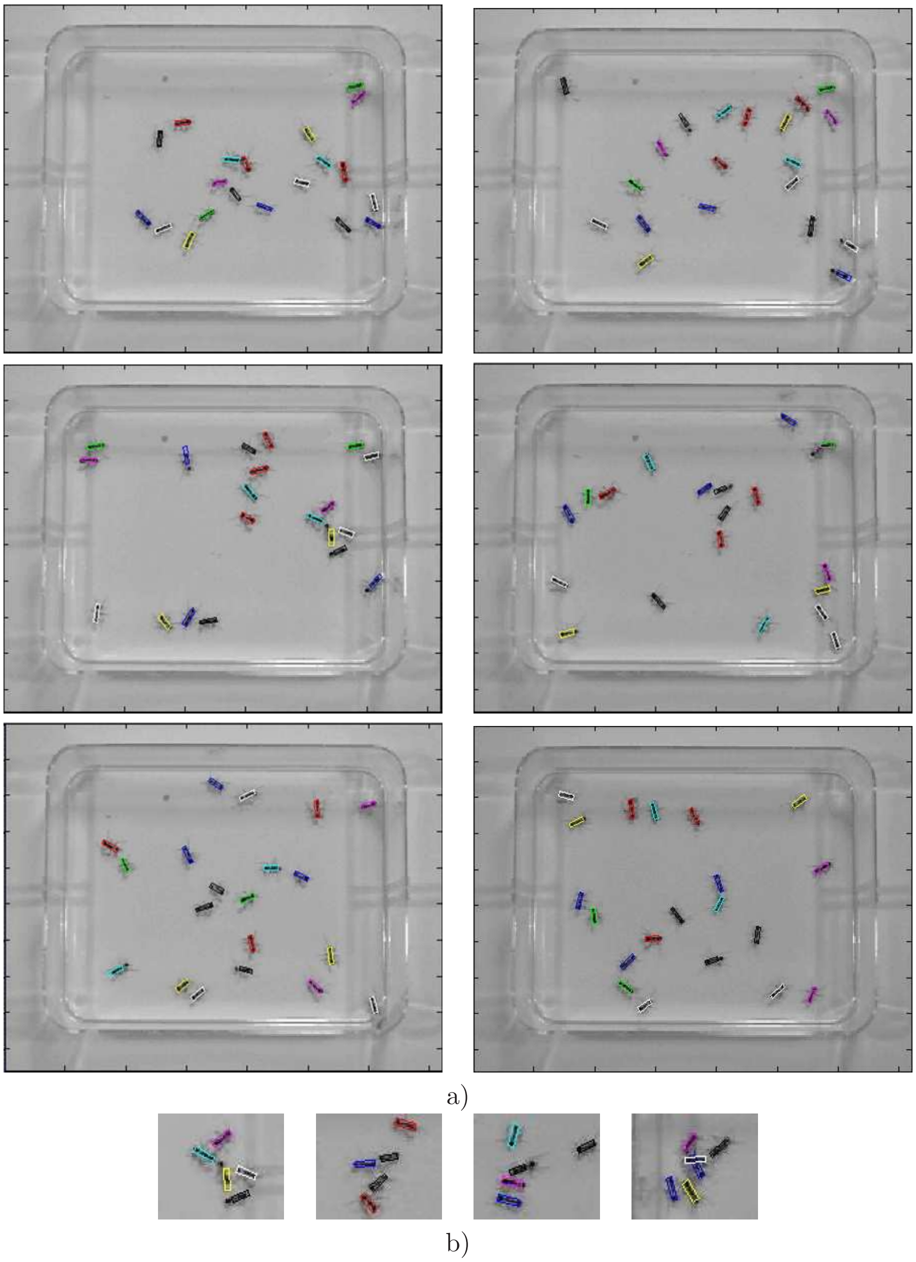
Results are highly satisfactory as shown in the previous mentioned *ant* sequence, Fig.5.4, where there are not any total occlusion between ants. More severe occlusions can be seen in Fig. 5.6 ([Okuma et al., 2004]) as well as in Fig. 5.5, where targets are highlighted in different colours. The simultaneous tracking of a large number of trackers is achieved with a computing cost reduction. Table 5.1 shows the results for the *ant* sequence, where ants move and rotate in a random way. A high accuracy level is achieved in location and angle estimation. Label change bin refers to the number of incorrect labelling situations after occlusions, i.e both ants have been tracked after the collision but with an identity swapping.

Table 5.1: Results obtained for the Ant sequence.

	Location Error [pix]	Angular Error [degrees]	Frames	Label Changes
Mean	2.7286	9.7191	4928	16
Std. Dev	2.2955	29.2229		

Table 5.2: Data association algorithm comparison on the Ant sequence.

	Length [frames]	Location Error [pix]		Angular Error [degrees]		Changes
		Mean	Std. Dev.	Mean	Std. Dev.	
MHT	2850	2.3936	1.868	6.5321	23.2734	3
Auction Alg.	2850	2.334	1.8148	7.5797	25.0922	6



a)

b)

Figure 5.4: Ant sequence (ftp://ftp.cc.gatech.edu/pub/groups/borg/pami05). a) Frames: 1, 101, 301, 601, 1001, 2001. b) Zoom on frames: 301, 538, 1691, 2980.

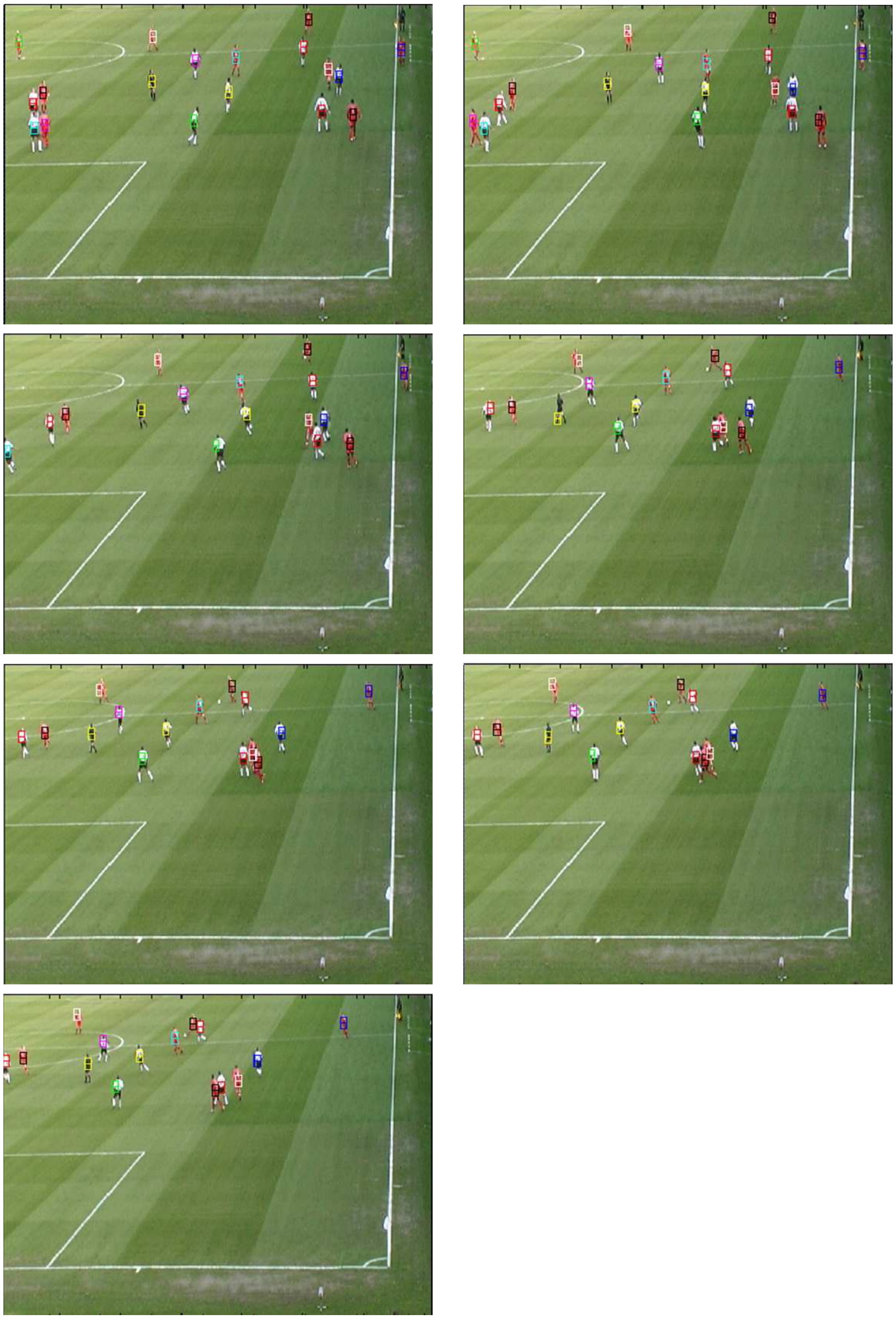


Figure 5.5: PETS football sequence.

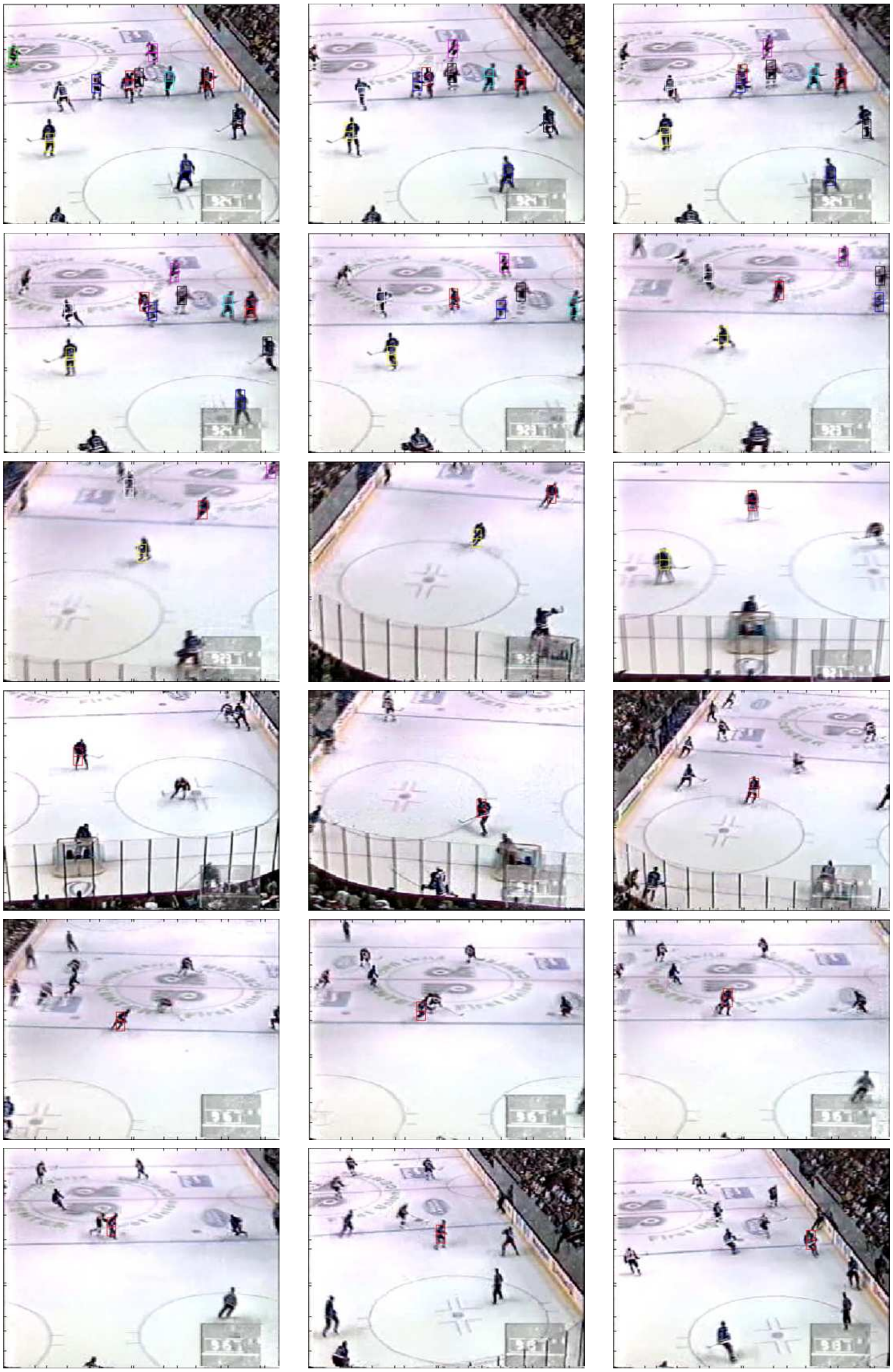


Figure 5.6: Tracking hockey players in the Okuma database with dynamic camera.

5.3 Multi-camera Solution

Even though the algorithm proposed in the previous section shows good results, it can not guarantee the correct solution of complex interactions, with a considerable length. A clear example of these situations can be observed during a sports competition. Patterns of motion like close defence, tackles and so on make impossible to solve them without extra information.

For that reason, most sports applications use multiple sensor to provide the system with the required information for a reliable solution. An application capable of combining information from different sensors (video cameras and ultrasound sensors) is presented in Appendix D. The rest of this chapter has been devoted to the combination of multiple identical sensors, specifically, video cameras. Although the information coming from different cameras can be considered independent if the viewpoint has been chosen carefully, in fact the measurements are correlated because they come from the same kind of sensor. Better results can be obtained using real independent sensors such as ultrasound, laser, accelerometers or infrared to name a few. Nevertheless, it is not always possible to incorporate these kinds of sensors, mainly because they are intrusive and this discards their usage in surveillance.

Multi camera systems give a considerable advantage in relation to monocular ones, specially in the treatment of occlusions. If the multi camera system has been defined correctly, with overlapping and independent viewpoints, the probability of occlusion in several cameras simultaneously is considerably reduced. This fact simplifies considerably the solution of the occlusions.

Nevertheless, some considerations should be taken into account. The usage of multiple cameras involves the necessity of a shared reference to conjugate the different coordinate systems of each sensor. In addition, complex systems of synchronisation and control are required to ensure the temporal coherence of the observations.

5.3.1 Multi-layer Feedback

Applying the same multi-layer methodology, we can take advantage of the tracking filter in one of the camera to help the other. Running independent tracking filters in independent sensors and combining the results provides a better global estimation than the sum of both. An example of this was depicted in Chapter 2, Figure 3.7.

Two alternatives have been evaluated. The first one is the immediate extension of the multi-target layered particle filter. Once the estimation of the first layer has been run in all the independent filters, a joint estimation is re-calculated in a shared reference space. Finally, this estimation is fed back to each one of the independent filters to re-weigh the hypothesis with respect to their correspondences with the joint estimation. In order to achieve that, the estimation is modelled as a mean location and an error (i.e a covariance matrix) before projecting it in the image space.

The projection to the shared reference, which can be one of the cameras or a reference plan, is made by a homographic transformation. The methodology is explained in Chapter 2, Section 3.3.1. Afterwards, the joint estimation is obtained by associating the measurements from all the cameras. Several options are available to do that, such as the association techniques, explained in the previous section, or the MCUKF, explained in Chapter 2 Section 3.2.1, as we will see in Section 5.4.

The algorithm (Algorithm 8) is similar to the one previously described. Usually one iteration is more than enough to obtain the convergence.

This algorithm has the property of maintaining slightly different target parameters for each camera. However, the joint estimation is unique and we can consider it as the real estimation. This fact permits the system recovers from severe failures in one of the cameras, which can

Algorithm 8: The multi-camera multi-layer particle filter for tracking of several objects

```
Given a particle set $\{x^i_{j,t-1,c}, \omega^i_{j,t-1,c}\}_{i=1}^N$ which represents the posterior probability
$p(x_{j,t-1,c}|z_{t-1,c})$ at time $t-1$ for $j = 1, ..., M$ targets and $c = 1, ...C$ cameras.

  1. Initialisation: $k \leftarrow 1$

     for $c = 1$ to $C$

         for $j = 1$ to $M$
           – Generate $N_1$ new samples from the importance function in the main
             partitioned dimensions. $x^i_{j,t,c} \sim g(x_{j,t,c})$.
           – Propagate $N_2$ samples from the old samples applying dynamic
             $x^i_{j,t,c} \sim p(x_{j,t,c}|x_{j,t-1,c})$.
           – Weigh the particles
             $\omega^i_{j,t,c,k} \propto \begin{cases} p(z_{t,c}|x^i_{j,t,c}) \cdot f_t(x^i_{j,t,c})/g_t(x^i_{j,t,c}) & i = 1, ..., N_1 \\ \omega^i_{j,t-1,c} \cdot p(z_{t,c}|x^i_{j,t,c}) & i = N_1 + 1, ..., N \end{cases}$
             and normalize them $\sum_{i=1}^N \omega^i_{j,t,c,k} = 1$.
         end

     end

  2. Iteration:

       • for $c = 1$ to $C$
           – Estimate the new position of the state $E[x_{j,t,c}] = \sum_{i=1}^N \omega^i_{j,t,k} \cdot x^i_{j,t}$ and the
             uncertainty $E[(x_{j,t,c} - E[x_{j,t,c}])^2]$
           – Project them in the shared reference.
         end
       • Assign the correspondences between measurements from different cameras and
         obtain the joint estimation $E[x_{j,t}]$.
         for $c = 1$ to $C$
           – Project back them in the camera $c$.
           – Re-weigh: $\omega^i_{j,t,c,k+1} \propto \omega^i_{j,t,c,k} \cdot \phi(x^i_{j,t,c}, E[x_{j,t}])$
         end
       • $k \leftarrow k + 1$. Iterate until convergence.

  3. Estimation:

     for $c = 1$ to $C$

         for $j = 1$ to $M$
           – Assign the final weights: $\{x^i_{j,t,c}, \omega^i_{j,t,c}\}_{i=1}^N \leftarrow \{x^i_{j,t,c}, \omega^i_{j,t,c,k}\}_{i=1}^N$.
           – Estimate the new position of the state
             $E[x_{j,t,c}] = \sum_{i=1}^N \omega^i_{j,t,c} \cdot x^i_{j,t,c}$
           – Resample $\{x^i_{j,t,c}, \omega^i_{j,t,c}\}_{i=1}^N$ to get $\{x^i_{j,t,c}, \frac{1}{N}\}_{i=1}^N$
         end

     end
```

be easily detected by means of a distance between the camera and the joint estimation. An example is depicted in Figure 5.7.

5.3.2 Unique Filter on the Plan Reference

The second alternative to be evaluated consists in a unique filter per target running on the shared reference. The drawback that this proposal presents is the loss of the guided sampling that importance sampling implies. Since the shared reference, usually a map or a plan, is a virtual space, it is not possible to apply a function $g(x_t)$. To solve this limitation, we propose to run the sampling process in the image world, coming back to the reference world for weighting, estimating and propagating.

In addition, once all the particles have been projected onto the plan, a clustering procedure is applied to reassign particles and deal with occlusions. In this manner, we avoid that near targets could confuse the importance function of the corresponding trackers. After reassigning particles, the final estimation is calculated on the plan.

In this manner, the algorithm can be described as Algorithm 9.

As advantage in relation to the previous version, this methodology requires less particles and we obtain a better estimation of the uncertainty on the plan. Since the clusterisation is made on the plane instead on the image (see Figure 5.8), the gaussian assumption is only made when all the information from the different sensors is available. On the other hand, by running independent filters in each camera we have at our disposal a effective mechanism to recover from failures in any filters. In addition, as we demonstrated in the previous chapter, thanks to the integral image the evaluation time for each particle is not the crucial issue anymore. A comparison between both techniques is made in Section 5.4.

5.4 A Specific Application: Sports Analysis

In this section we propose a complete application capable of tracking multiple objects in an environment monitored by multiple cameras. It encapsulates most of the techniques presented in this thesis to deal with problems such as multi-target, efficient tracking, occlusions, multi-camera applications and so on.

Professional sport is an extremely competitive world. Mass media coverage has contributed to the popularity of sport, increasing its relevance in current society due to the money and fame that it generates. In this environment, in which any assistance is welcomed, video-based applications have proliferated.

Video-based approaches have shown themselves to be an important tool in analysis of athletic performance, especially in sport teams, where many hours of manual work are required to analyse tactics and collaborative strategies. Computer vision based methods can provide help in automating many of those tasks.

Sport analysis can be considered as a classic human activity recognition problem with several distinctive constraints and requirements, for instance, a fixed number of targets. The huge amount of interactions between players during a whole match due to the sport activity makes it impossible to track all players with a single camera. In this application, we have concentrated our efforts in developing a tracking application capable of managing multiple sensors in order to track multiple objects simultaneously.

The research presented in this section examines the task of designing and implementing a sport analysis tool called ASTRO (Automatic System for Tactical Review and Optimization). ASTRO has been built in the framework of a regional project involving the Aragon Institute of Engineering Research (I3A) and the Real Zaragoza Football Club S.A.D, and financed by

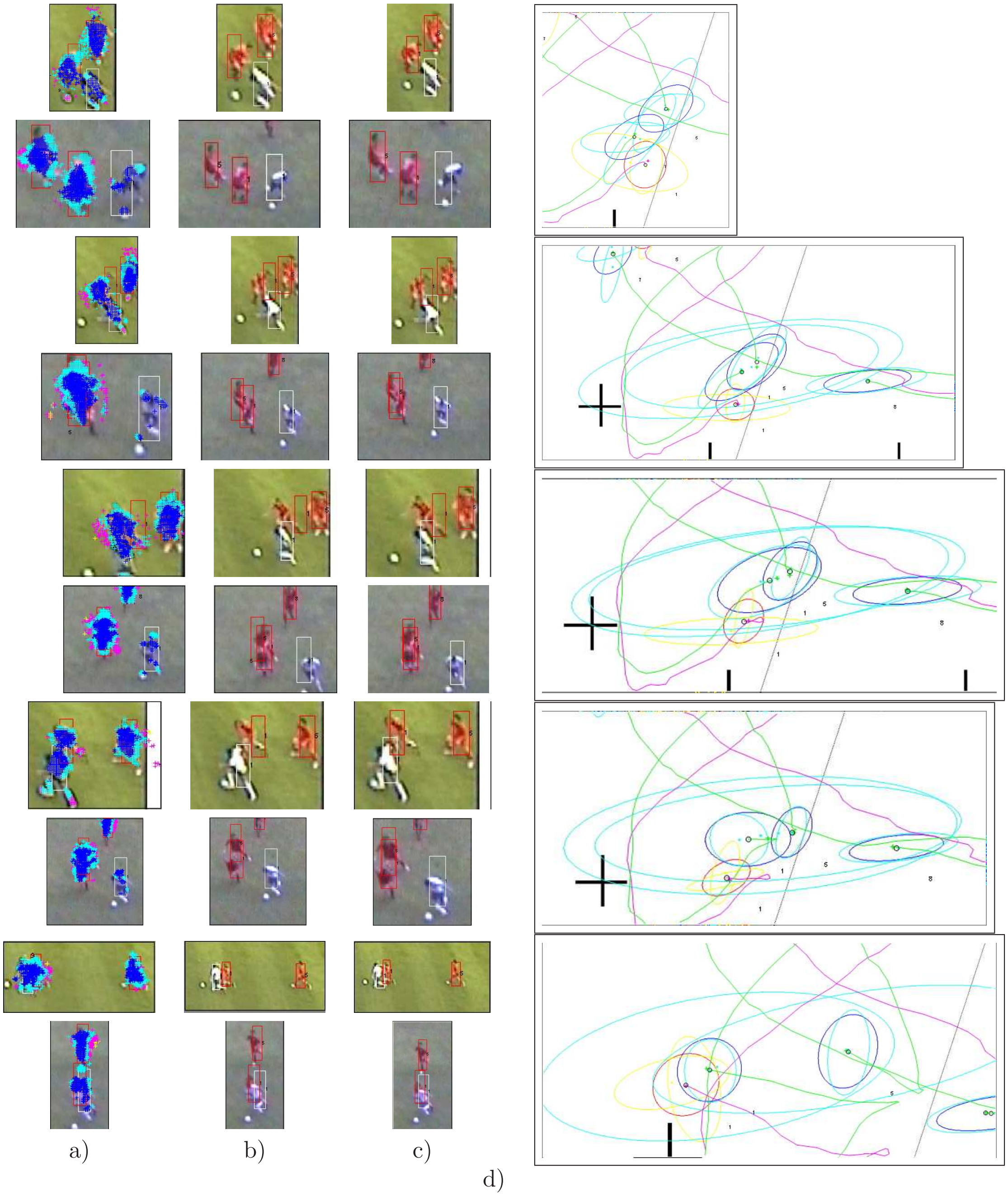


Figure 5.7: a) Particle distribution and first layer estimation, b) & d) estimation on the plan conjugating all camera information, c) estimation after the 2nd layer (feedback).

Algorithm 9: The multi-camera shared reference particle filter for tracking of several objects

```
Given a particle set $\{x^i_{j,t-1}, \omega^i_{j,t-1}\}_{i=1}^N$ which represents the posterior probability
$p(x_{j,t-1}|z_{t-1,1:C})$ at time $t-1$ for $j = 1, ..., M$ targets.

  1. for $c = 1$ to $C$

         for $j = 1$ to $M$
           – Generate $N_1$ new samples from the importance function in the main
             partitioned dimensions. $x^i_{j,t,c} \sim g(x_{j,t,c})$.
           – Propagate $N_2$ samples from the old samples applying dynamic
             $x^i_{j,t} \sim p(x_{j,t}|x_{j,t-1})$.
           – Project back $N_2$ samples in the camera $c$ $x^i_{j,t,c} = H_c^{-1} \cdot x^i_{j,t}..$
           – Weigh the particles
             $\omega^i_{j,t,c} \propto \begin{cases} p(z_{t,c}|x^i_{j,t,c}) \cdot f_t(x^i_{j,t,c})/g_t(x^i_{j,t,c}) & i = 1, ..., N_1 \\ \omega^i_{j,t-1} \cdot p(z_{t,c}|x^i_{j,t,c}) & i = N_1 + 1, ..., N \end{cases}$
             and normalize them $\sum_{i=1}^N \omega^i_{j,t,c} = 1$.
         end
       • Project the particles in the shared reference $\tilde{x}^i_{j,t,c} = H_c \cdot x^i_{j,t,c}$.

     end

  2. Assign the particles and weights: $\{x^i_{j,t}, \omega^i_{j,t}\}_{i=1}^{N'} \leftarrow \{\{\tilde{x}^i_{j,t,c}, \omega^i_{j,t,c}\}_{i=1}^N\}_{c=1}^C$, where
     $N' = N \cdot C$.

  3. Clusterise the target density defined by $\{x^i_{j,t}, \omega^i_{j,t}\}_{i,j}$.

  4. Assign the correspondences between clusters and target's particles using the associating
     algorithm and obtain the joint estimation $E[x_{j,t}]$.

  5. Re-weigh: $\omega^i_{j,t} \propto \omega^i_{j,t} \cdot \phi(x^i_{j,t}, E[x_{j,t}])$

  6. Estimate the new position of the state
     $E[x_{j,t}] = \sum_{i=1}^{N'} \omega^i_{j,t} \cdot x^i_{j,t}$

  7. Resample $\{x^i_{j,t}, \omega^i_{j,t}\}_{i=1}^{N'}$ to get $\{x^i_{j,t}, \frac{1}{N}\}_{i=1}^N$
```

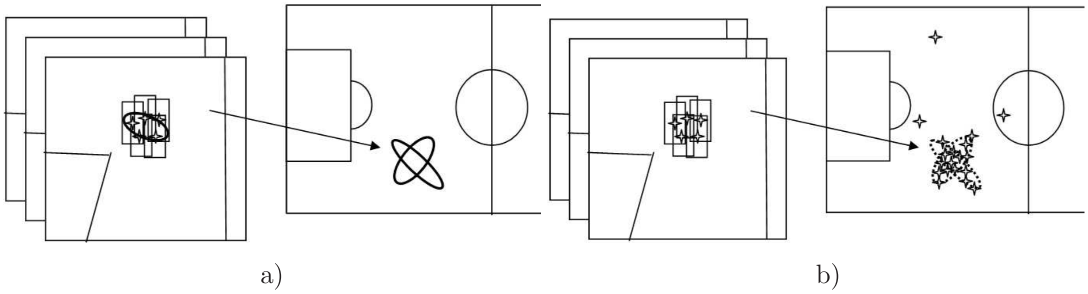


Figure 5.8: Uncertainty estimation for both multi-target methodologies: a) Individual filters with feedback, b) Unique filter on the plan. While the first one makes the Gaussian assumption on the image, the second one runs the clusterisation on the shared reference.

the Government of Aragon. It was our group's task to build the football player detection and tracking system for ASTRO, and the results are presented in this section.

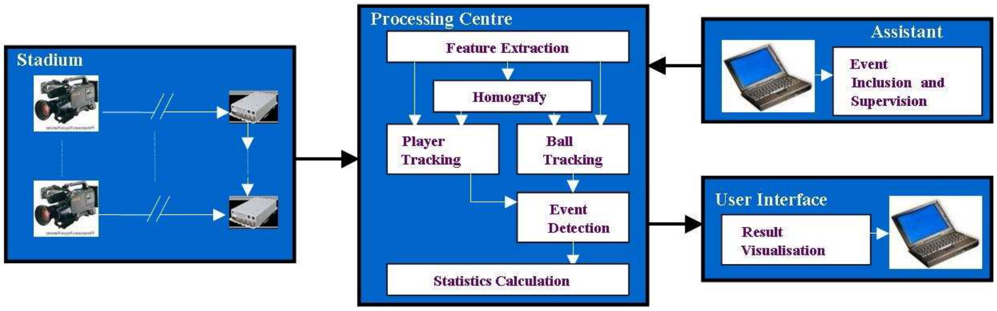


Figure 5.9: ASTRO project general scheme.

The two approaches proposed in Section 5.3 have been tested. In the first approach, a Particle filter is applied to each camera, which permits us to send an indication of the reliability to the plan, as well as more accurate and more robust measurements. In this manner, a non-categorical decision is taken over the image. If the hypothesis is rejected on the plan using multi-camera information, the decision will be corrected. This feedback procedure assures that the final decision will be made using all the available information coming from all the cameras. Thus, each camera can correct its own estimation with the information of the other cameras. Furthermore, the feedback process monitors the entry and exit of players in the scene. As association algorithm, the multi-view tracking algorithm uses Unscented Kalman trackers [Julier and U., 1997][Wan and v. d. M., 2001] to model player's position and velocity on the football field. The algorithm receives multiple measurements from each camera, applies a data association method, which makes the correspondences between measurements and trackers, and estimates the new position of all players. A description of the system is shown in Figure 5.9.

In the second approach, particles are launched in each camera but the conjugation of all of them is made by projecting them in the plan. In addition, a clustering algorithm is applied in the shared reference to correct the coalescence between filters in the image.

5.4.1 System Architecture

The system input is composed of video data from static analogue cameras with overlapping fields-of-view at a football stadium. The cameras are positioned around the stadium at 25 metres in height (on the roof). A compromise between cost and good performance (resolution, overlapping,...) has been sought. A detailed scheme of the locations can be viewed in the Figure 5.10. All cameras have been calibrated to a shared ground-plane co-ordinate system using homographic transformations. [Hartley and Zisserman, 2004].

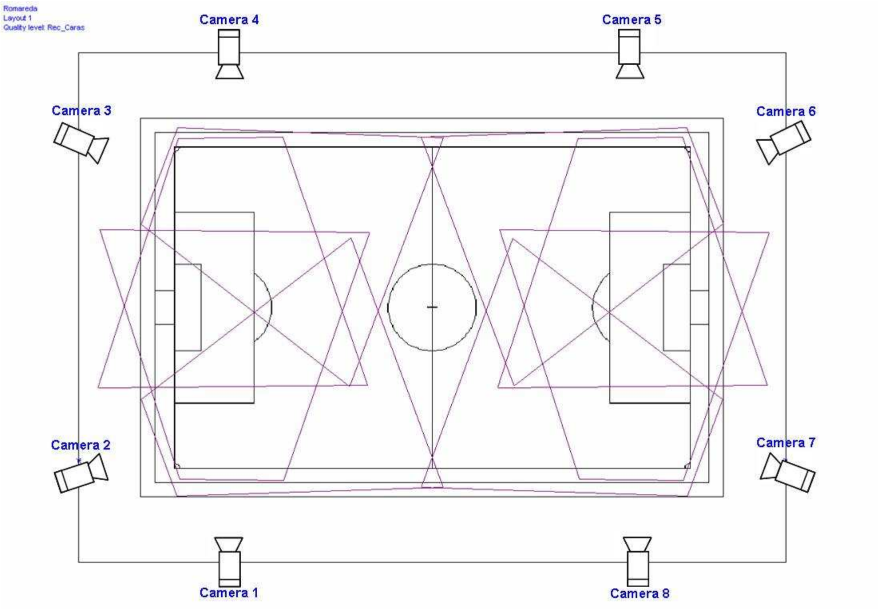


Figure 5.10: Camera distribution and field of view on the football pitch.

The video flow is recorded by a video capture card connected to a PC. Each card has 4 independent channels with 4 DSP, which allow us to record video with a rate of 14 frames per second, using a hardware MPEG-4 video compression codec. In order to record the information provided by 8 cameras, 2 computers have been installed. The system has been designed to work assuming synchronisation between cameras. The 4 cameras connected to the same computer are automatically synchronised by the video capture card, but the synchronisation between both recording servers is obtained using an ad-hoc wireless network (WIFI), which synchronises the system clocks of both computers.

Recorded videos are sent to the processing server. This server comprises 8 single-camera processing computers, a multi-camera integration server which receives data from each camera processor and gives the final estimation. A GigaLAN switch links all computers and makes possible the message transference. The multi-camera integration server directs and controls the process. It is the device in charge of maintaining the synchronisation between cameras as well as obtaining the position of all players at each time step. The infrastructure and its connection can be seen in Figures 5.11 and 5.12.

Each computer associated to each camera processes its corresponding frame obtaining a set of features and a first hypothetical position of targets in the image. When it has finished, data are transmitted to the server. A multi-camera server awaits the responses of all cameras and updates the state estimation of the player on the pitch. Finally, the server sends a message to

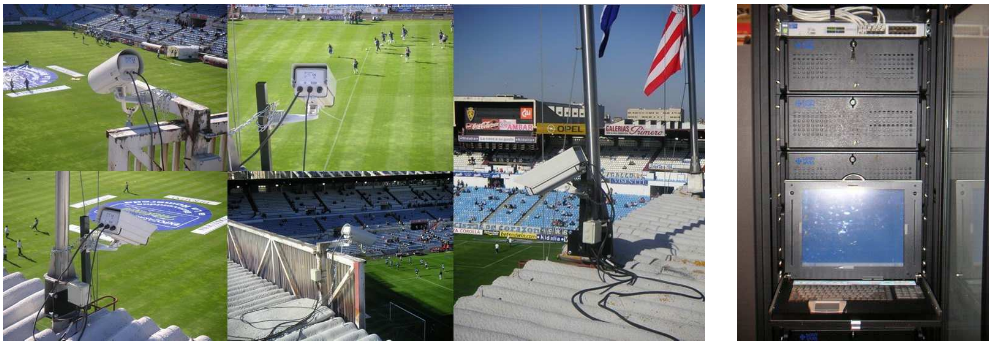


Figure 5.11: Installed cameras and processing cluster.

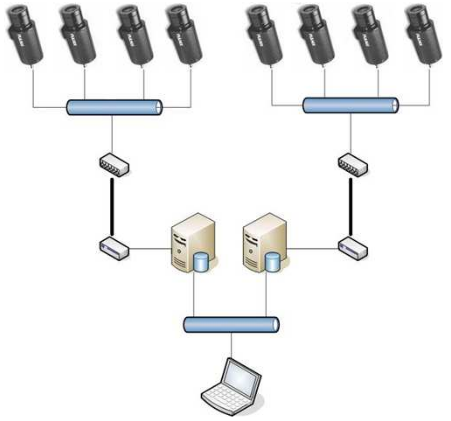


Figure 5.12: System architecture.

permit the camera processor to continue with the following frame. However, this message is not a simple acknowledgement since it has feedback information in order to correct failures in the image.

Algorithms have been compiled in Visual C++ and programmed with a multi-thread philosophy.

5.4.2 General Scheme Option 1: Multi-layer Feedback

The processing algorithm can be divided into two main parts: the single-view processing stage, which is applied to each camera independently, and the muti-view processing stage, which integrates the previous results and gives us the final estimation state.

The system has been specially developed to be applied to sport games, and it has been evaluated in a real football stadium. Each target is tracked using a local importance sampling particle filter in each camera, but the final estimation is made by combining information from the other cameras using a modified UKF algorithm. Multi-camera integration enables to compensate bad measurements or occlusions in some cameras thanks to the other views. The final algorithm results in a more accurate system with a lower failure rate.

The purpose of the single-view processing stage consists in extracting a set of hypotheses which will be considered in the multi-view tracking process. By applying a tracking algorithm, we obtain a robust-to-occlusion method which extracts more plausible hypotheses. Particle filter is a good election due to its advantages in multi-target tracking. Colour, movement and tridimensional information are used to determine the likelihood of the particles, and to weigh them.

Results of this stage are modelled as Gaussians, being the mean the position of each player and the covariance the reliability of this location. Both are sent to the multi-view tracking algorithm as measurements. In this way, we obtain more robust and more accurate measurements with an extra feature: their reliability.

For the multi-view tracking process, Unscented Kalman (see Chapter 3) trackers are used. First, a data association algorithm establishes the correspondences between measurements and trackers. Then, the UKF algorithm combines all the measurements corresponding to each tracker taking into account their reliability.

The output from this process is the 20 player positions per time step (excluding both goalkeepers). The system also indicates the category (team) of each player, and maintains the correct number of players in each category. Furthermore, although the identification of individual players is not possible given the resolution of input data (only team is recognised), a label with the name of each player is assigned in the first frame and it will be tracked during the whole match.

Finally, the output is sent to each camera to correct failures in the image tracking, that is called the feedback procedure previously mentioned.

5.4.2.1 Single-view processing stage

As mentioned in previous sections, targets in the image are tracked using individual particle filters. We apply a particle filter to track each player. The importance function $g(x_t^i)$, shown in Equation (4.1), is introduced in the algorithm to improve the sampling process. This function is obtained extracting the main colours of the object (each colour is a Gaussian) and detecting them in the full image or in the zone surrounding the estimate position (extracted with the last mean state) whose dimensions depend on the position and speed variances (see Figure 5.13). PDIs are calculated using the Mahalanobis distance to the player Gaussians, and Colour Mask is obtained like the cluster closest to the new sample. The number of Gaussians has been chosen automatically using the index MVI (see Section 4.2.3.1), obtaining the best results with 4 foreground Gaussians and 2 background Gaussians.

Moreover, we will also apply the information contained in PDIs to obtain a fast estimation of the posterior density by means of the integral image.

5.4.2.2 Posterior probability

Once prediction has been calculated, the multiple hypotheses generated must be evaluated using an adequate likelihood function. In order to obtain the likelihood function which will weigh each particle, we combine multiple visual clues: colour, movement and height difference. Colour is the most discriminative clue which differentiates between targets and background, but also between different kinds of objects. Movement can not distinguish among players, not being able to identify the player's team, but is very useful to eliminate background areas with the same colours as the targets, for instance, lines which define the field. Height measurement helps to compensate the perspective effect and, thus, to obtain a better estimation of the real size of the object.

Assuming independence between the visual clues, the combination is made as follows:

$$p(z_t|x_t) = p(z_t^{colour}|x_t) \cdot p(z_t^{motion}|x_t) \cdot p(z_t^{height}|x_t) \tag{5.6}$$

Colour probability A new input frame is projected onto the target probability space to generate the Colour Probability Density Image (PDI). We generate a PDI for each kind of object to be tracked. In our particular case, we need two PDIs, one for each team, but more PDIs could be generated: two PDIs for each team if clothes have complex colours, or even one for each person in a video surveillance application. The values of the PDI pixels are taken from the Gaussian mixture models of the targets used as classifiers for each pixel in the input picture. The probability assigned to each pixel is given by the distance (using as metric the Mahalanobis distance) to the nearest Gaussian. More details were given in Chapter 4.

In order to generate the Gaussian mixture model, we extract pixels corresponding to both teams and background (field). Each pixel is projected to the HSV space to reduce the influence of changing illumination and shadows. In order to compensate illumination changes, the mean and covariance of Gaussians classified into background and halo are updated, using an on-line updating algorithm [Elgammal et al., 2002]. Gaussians associated to each team are not updated due to the risk of introducing noise.

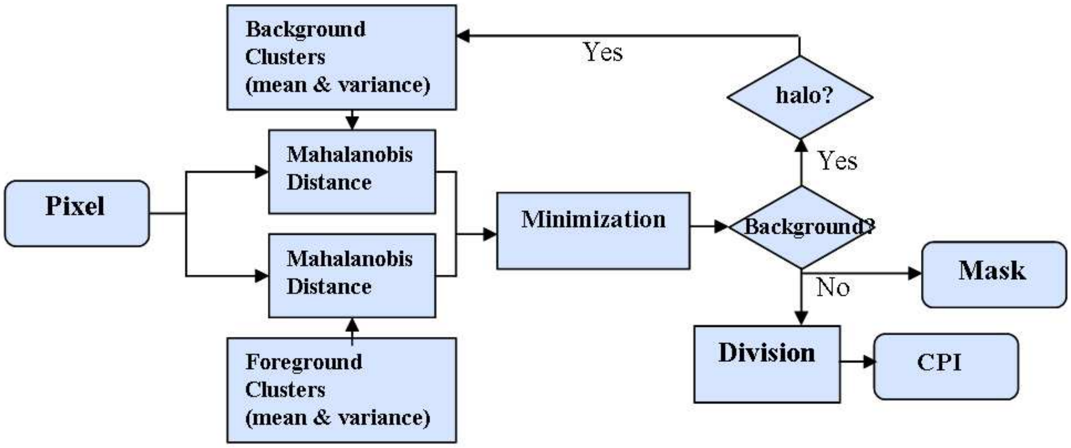


Figure 5.13: PDI and prior mask generation algorithm.

Once PDIs have been calculated (Figure 5.13), the probability associated to each particle can be extracted using the integral image (see Figure 5.14). The state of every particle defines the width **W** and height **H** of the target as well as the position of the centre of the target (x_0, y_0). Using these parameters, a rectangular kernel, with the same dimensions of predicted target, enables us to obtain the posterior probability.

$$\begin{aligned}\tilde{\mathbf{\Pi}}_t^c(i) &= (D + A - C - B)\\ \mathbf{\Pi}_t^c(n) &= \frac{\tilde{\mathbf{\Pi}}_t^c(n)}{\sum\limits_n \tilde{\mathbf{\Pi}}_t^c(n)}\end{aligned} \tag{5.7}$$

where t is the temporal instant and n is the particle number.

Motion probability Motion probability is a weight assigned to each particle and based on the number of pixels of motion that it contains. Their values are computed using the same kernel used for calculating colour probability but, in this case, over the motion detection image I_{mov}.

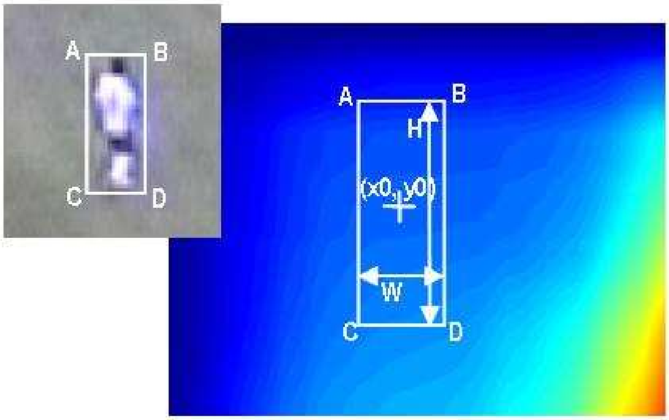


Figure 5.14: Convolution kernel to enhance target candidates on the integral image. **W** and **H** stand for Target predicted Width and Height.

$$\begin{aligned}\tilde{\mathbf{\Pi}}_t^m(n) &= \frac{\displaystyle\sum_{x=x_0-\mathbf{W}/2}^{x_0+\mathbf{W}/2}\sum_{y=y_0-\mathbf{H}/2}^{y_0+\mathbf{H}/2} I_{mov}(x,y)}{\mathbf{H}\cdot\mathbf{W}} \\ \mathbf{\Pi}_t^m(n) &= \frac{\tilde{\mathbf{\Pi}}_t^m(n)}{\displaystyle\sum_n \tilde{\mathbf{\Pi}}_t^m(n)}\end{aligned} \tag{5.8}$$

Height difference probability The last weight is defined as the difference between the height stored in the state vector of the particle and the height which the particle must have because of its new position (given by the propagation stage). This height measurement is given by an algorithm, called Height Estimator, which was explained in Chapter 3 Section 3.4.3. The particularisation for ASTRO project is depicted in Figure 5.15.

$$\begin{aligned}\tilde{\mathbf{\Pi}}_t^h(n) &= (\mathbf{H} - Height(x_0, y_0))^{\alpha} \\ \mathbf{\Pi}_t^h(n) &= \frac{\tilde{\mathbf{\Pi}}_t^h(n)}{\displaystyle\sum_n \tilde{\mathbf{\Pi}}_t^h(n)}\end{aligned} \tag{5.9}$$

where α is a constant which fixes the discriminative power of the weight and depends on the field of view of the camera and the lens parameters.

This weight introduces crucial information to discriminate bad measurements such as merged or cut-up targets, which colour and motion could accept like good observations.

5.4.2.3 Estimation of the new state

Finally, once the last *a posteriori* feature has been obtained, all these features must be combined in order to estimate the new state for the object to be tracked. The score for each candidate is calculated using the multiplication rule, that is, assuming independence between features.

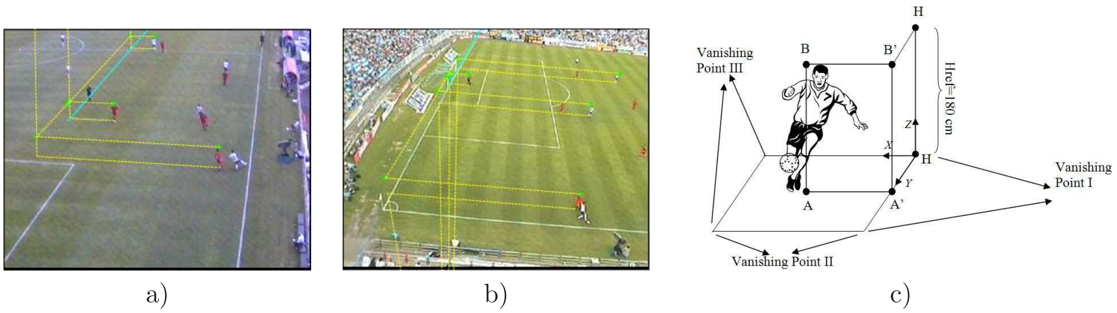


Figure 5.15: a) & b) Estimated height for several player in the image and their projection onto the height reference. c) Height estimation procedure.

$$\mathbf{\Pi}_t(n) = \mathbf{\Pi}_t^c(n) \cdot \mathbf{\Pi}_t^m(n) \cdot \mathbf{\Pi}_t^h(n) \tag{5.10}$$

The particle set corresponding to this candidate is used to estimate the new state by means of a weighted average:

$$E\left[S_t\right] = \sum_{n=1}^{N} \frac{\mathbf{\Pi}_t(n) f_t(x_t)}{g_t(x_t)} \cdot x_t(n) \tag{5.11}$$

where a correction factor must be included in the particle weights due to the importance sampling .

However, this estimation is independent for each camera and it can fail if there are occlusions in the image. Using multi-camera information, this effect is reduced, as we will explain in Section 5.4.2.6. A global view of the algorithm can be seen in Figure 5.16 and in Algorithm 7.

5.4.2.4 Multi-sensor data fusion

Once all players have been detected in each camera, we project all the measurements onto the plan in order to have a shared reference space. Each hypothesis is projected converting it in a single point, the point which is in contact with the floor (the lower point). When these transformations have been made, the multi-camera tracking is applied. For this purpose, we use the multi-Camera Unscented Kalman Filter (MCUKF) (see Chapter 3) as tracking algorithm.

Our reference is a plan of the football field, and a homographic matrix for each camera has been calculated. Thus, we can transform points from the image to the coordinate system of the plan.

MCUKF allows combining different measurements from different sensors assigning a different confidence to each measurement. In Chapter 3, all measurements were equally valid in the image, being the distance to the camera the factor which differentites the observations. To achieve that, an identity matrix was used as covariance in the image and, by homographic transformation, the homographic error was introduce in the covariance matrix. On the contrary, in this system, we can obtain a better covariance of the measurement than a simple identity matrix. A weighted covariance can be calculated by taken into account all the target's particles and their weights. Equation to compute this weighted covariance is:

$$C = \frac{\sum_{i=1}^{N} w_t^i \cdot (x_t^i - E[x_t])' \cdot (x_t^i - E[x_t])}{\sum_{i=1}^{N} w_t^i} \tag{5.12}$$

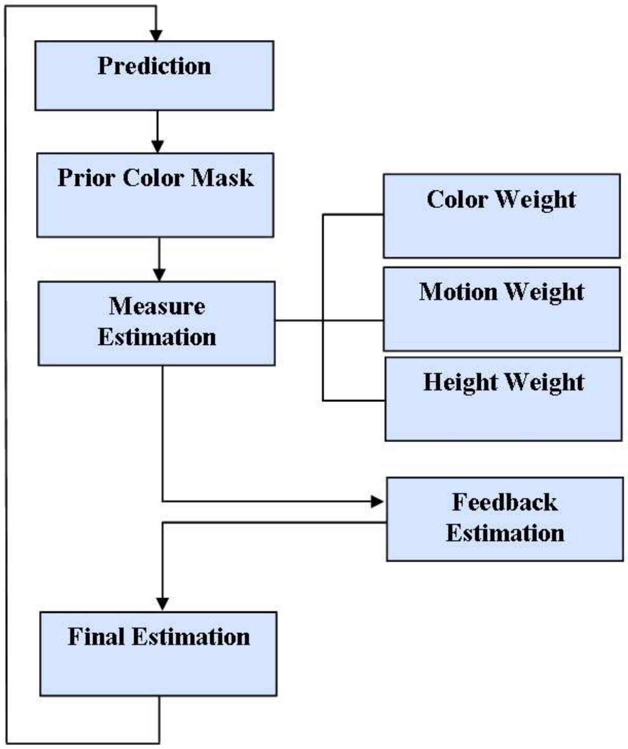


Figure 5.16: Single view tracking algorithm.

where x_t^i are the particles and w_t^i the weights before normalisating.

Using this equation, those samples with higher probability take a higher importance in the covariance calculation. However, an extra factor should be introduced to reduce the reliability of those measurements with a poor global performance. This factor can be computed as the mean of the weights w_t^i, that is $f = \frac{\sum_{i=1}^{N} w_t^i}{N}$. Therefore, the weighted covariance can be formulated as:

$$C = N \frac{\sum_{i=1}^{N} w_t^i \cdot (x_t^i - E[x_t])' \cdot (x_t^i - E[x_t])}{\left(\sum_{i=1}^{N} w_t^i\right)^2} \tag{5.13}$$

Afterwards, the covariance matrix is projected on the plan by means of the equation (3.23).

5.4.2.5 Data association

Since there are several targets moving in the same region interacting with each other, we need to assign the corresponding measurement to each tracker chosen, between all the measurements and the possible existing distracters. This process will be made in the matching stage to take into account the coalescence, which is not properly modelled by using independent filters.

For sake of clarity, the detailed explanation is given in Appendix E. The correspondence between MCUKF trakers and particle filter estimations is based on a modified version of the nearest neighbour algorithm [Zhang et al., 1996]. The number of combinations is limited with respect to the original in order to reduce the computing time, at the expense of obtaining a sub-optimal result.

5.4.2.6 Feedback procedure

Image tracking is conditioned by camera location and player occlusions, which can produce tracking failures. These situations are partially solved by the multi-camera tracking on the plan. In those situations, player locations at the image will be different from locations on the plan. In order to solve these incoherences, state vectors are fed back from the plan to each view.

This information is integrated in the Particle filter as an extra likelihood in a second layer. Particles of each player are re-weighted assuming the position in the plan of this player as a Gaussian or a super-Gaussian and assigning the weight with the distance to the center. In this manner, if the location in both trackers fits each other, the result will not be affected. On the other hand, if they do not fit in, the non-accurate location in the image is corrected. Thus, each camera tracking is assisted by other cameras thanks to the feedback procedure, solving errors which could not be solved with a single view, and improving the results of all cameras.

Therefore, the interaction function is defined as:

$$\phi(x^i_{j,t,c}, E[x_{j,t}]) \propto e^{(-(x^i_{j,t,c}-E[x_{j,t}])' \cdot E[(x_{j,t}-E[x_{j,t}])^2]^{-1} \cdot (x^i_{j,t,c}-E[x_{j,t}]))^{\alpha}} \tag{5.14}$$

where $E[x_{j,t}]$ is the MCUKF estimation of the mean $\hat{\mathbf{x}}_{j,t}$ and $E[(x_{j,t}-E[x_{j,t}])^2]$ is the covariance $\hat{\mathbf{P}}_{j,t}$. α is a factor which allows controlling and tunning the influence of the feedback process in the final estimation. In this system, it has been setting up to be the 70% of the final weight.

5.4.2.7 Player camera transition

Finally, we need an algorithm to administer the transition of players between cameras. Although the number of targets is fixed and, therefore, the number of multi-camera estimations is also fixed, players go in and out of the camera view. In these cases, we have to create or delete, respectively, the local tracker in the image. In the first case, a new tracker is initialised with the label and the particles corresponding to the player.

In addition, in spite of all the previous mechanism to model interaction and avoid tracking failures, due to the small size of players in the image, the measurement is frequently completely lost. For this reason, a reset mechanism has been implemented. This reset can act in two different situations.

- If the location of the player in the plan is too different from the location in the image, a failure in the image will be taken for granted and re-initialised with the location in the plan.
- If two plan trackers catch the same measurement and superimpose to each other, a failure in the plan will be taken for granted. To solve it, extra un-assigned measurements are looked for in the image, and one of the trackers is re-initialised to the nearest measurement.

5.4.2.8 Results

Our system has been tested using several sequences of a football database. This database has been taken in collaboration with the Government of Aragón, the University of Zaragoza and Real Zaragoza S.A.D football team. For this purpose, eight analogue cameras were installed in the football stadium and connected to two MPEG4 video-recorders. In the first frame, the initialisation is given by hand, choosing the players in each camera. We use a constant acceleration model and a motion dynamic which permits objects with variable trajectories. By introducing x and y velocities in the state vector we can solve occlusions between tracked objects

$$\hat{\mathbf{x}}_k = [x \quad v_x \quad y \quad v_y] \qquad \hat{\mathbf{x}}_{k|k-1} = \mathbf{F} \cdot \hat{\mathbf{x}}_{k-1} \tag{5.15}$$

and the dynamic matrix is given by

$$\begin{bmatrix} x_k \\ v_k^x \\ y_k \\ v_k^y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k^x \\ y_k \\ v_k^y \end{bmatrix} \tag{5.16}$$

Once tracking has finished, we can extract statistics from the trajectories of the targets, such as maximum velocity, mean velocity, covered distance or number of sprints, to name a few.

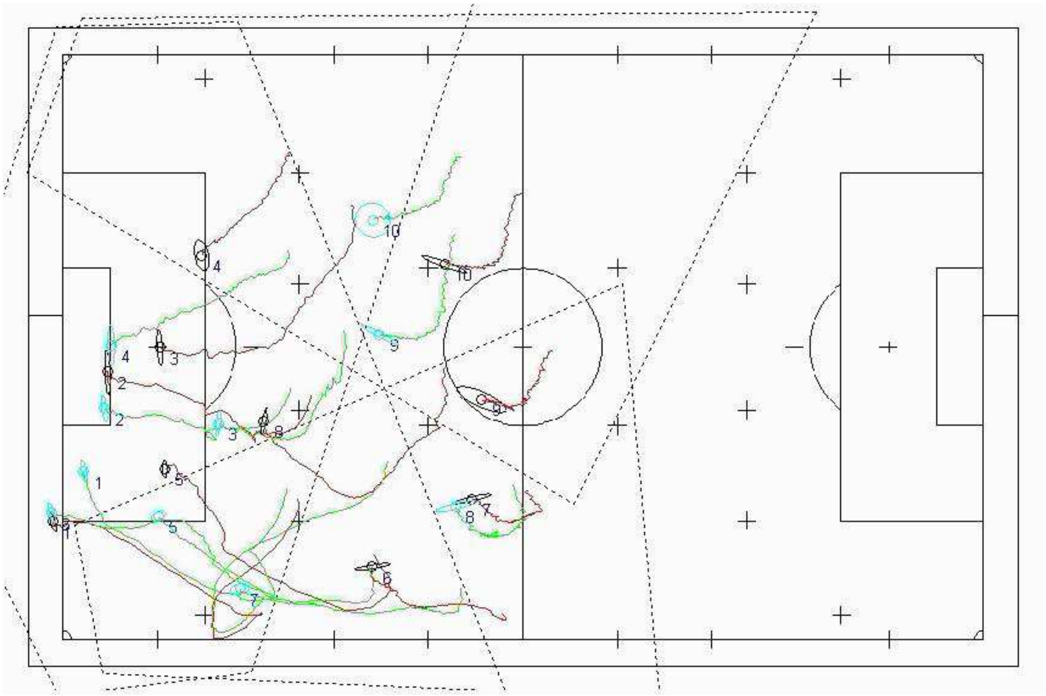


Figure 5.17: Trackers and trajectories during a period of the football match (Sequence 1).

In order to evaluate the system and obtain numerical results, we have manually labelled 2600 frames corresponding to all cameras for 2 sequences of 12 seconds. Sequences have been chosen with many complex interactions between players of both teams, but without any scrums or other special circumstances. In this manner, the accuracy of the system can be checked using the groundtruth of the labelled sequences. Results and statistics are shown in Tables 5.3, 5.4 and 5.5 and Figures 5.17, 5.18, 5.19, 5.20. Reported occlusions are those among players belonging the same team, that is, with the same colour model. Occlusions between different teams, although they are common due to defensive coverages, have been solved automatically by the tracking algorithm in all the observed situations.

A good accuracy level has been obtained for both sequences because of two reasons: the multi-camera tracking which reduces the measurement noise of each camera using the others, and the high precision of the single-camera stage. To estimate the importance of this single-camera stage, we have tested the system without it, that is, sending directly the blob position

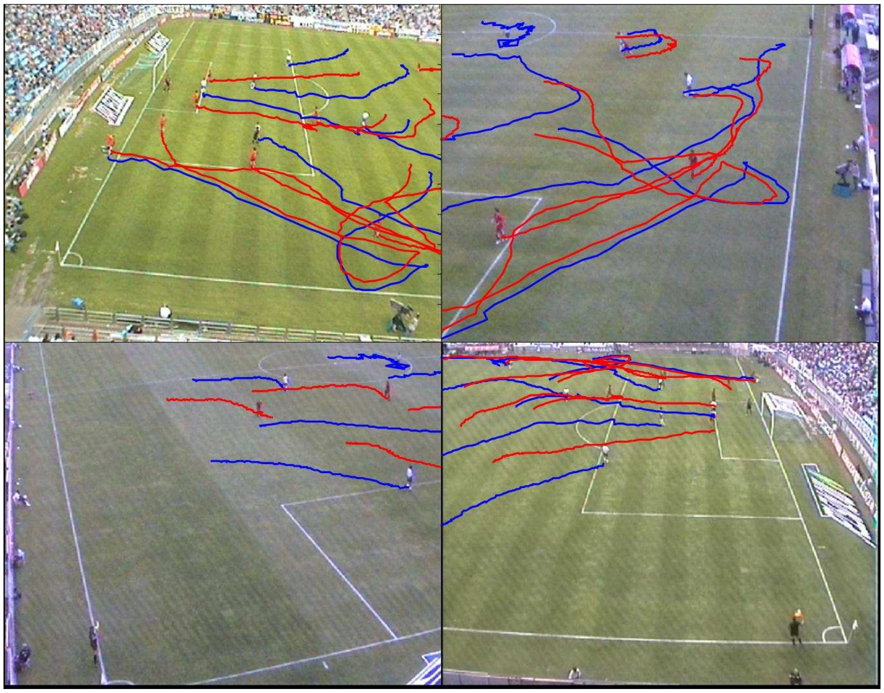


Figure 5.18: Trajectories at each view during a period of the football match (Sequence 1).

from each camera to the plan. This configuration gives us a value of location mean error equals to 1.182 metres, which means an increase of 125 percent in relation to the multi-camera approach.

Sequence 2, more complex than the first one, presents lower performance since most of interactions happen in an area which is not properly covered by any camera due to the poor image resolution (see Figures 5.19 and 5.20). As consequence, the accuracy decreases and one of the targets is finally lost. This lost player (player 7, team 1) has been highlighted in Table 5.4 to show the incoherent data reported, which can provide useful information to easily identify lost trackers. By checking performance statistics, like velocity or distance, trackers with physically impossible results could mean a wrong tracking. This fact is important because error data require an annotated sequence, which is not available during a real performance. This information could be used to generate automatically alarms and act accordingly.

Although excellent results have been obtained, it is not easy to extract conclusions about the performance during a whole match with such short sequences. For this reason, the complete system has been tested uninterruptedly during a long sequence of 8 minutes to abstract conclusions and qualitative statistics. This sequence includes all sort of situations inherent in a football match (Fig 5.21), such as fouls, scrums, goals and the rest of complex interactions. Thanks to this test, results (see Table 5.6) can be analysed, and real conclusions can be extracted.

Table 5.6 encapsulates the advantages of our proposal. The error rate represents the number of manual corrections divided by the length of the sequence and the saving rate is the number of manual corrections divided by all the locations needed to track the players. This means that a human supervisor must act only during the 5.84 percent of the match time in order to correct the system errors. Thus, our system is able to save the 99.4 percent of the work that a

Table 5.3: Sequence 1 statistics. Error measurement has been obtained using a manual-labelled ground-truth.

	Distance [metres]		Max. Velocity [Km/h]		Mean Velocity [Km/h]		Sprints		In Play [min]		Max. Error [metres]		Mean Error [meters]	
Team	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Player 1	54.95	54.09	24.54	23.19	17.05	16.78	2	2	0.19	0.19	1.04	1.07	0.41	0.36
Player 2	33.55	30.11	18.56	12.80	10.41	9.34	1	2	0.19	0.19	1.14	1.35	0.36	0.43
Player 3	39.54	24.53	19.72	10.52	12.27	7.61	1	1	0.19	0.19	1.32	0.72	0.54	0.25
Player 4	27.27	27.89	14.91	12.98	8.46	8.66	1	1	0.19	0.19	1.78	0.94	0.64	0.30
Player 5	40.04	42.88	17.17	15.93	12.43	13.31	1	0	0.19	0.19	0.92	0.85	0.33	0.37
Player 6	25.14	28.50	10.26	16.26	7.80	8.85	1	1	0.19	0.19	2.05	2.45	0.88	0.36
Player 7	18.09	45.63	7.03	17.46	5.61	14.16	2	2	0.19	0.19	0.92	1.35	0.30	0.40
Player 8	37.14	30.98	14.04	14.04	11.53	9.62	1	2	0.19	0.19	1.09	1.42	0.38	0.39
Player 9	20.40	22.70	8.16	8.71	6.33	7.02	1	1	0.19	0.19	1.22	0.93	0.52	0.37
Player 10	19.61	19.02	6.25	7.76	6.06	5.88	0	1	0.19	0.19	1.02	0.95	0.41	0.42
Total											2.45		0.401	

Table 5.4: Sequence 2 statistics.

	Distance [metres]		Max. Velocity [Km/h]		Mean Velocity [Km/h]		Sprints		In Play [min]		Max. Error [metres]		Mean Error [metres]	
Team	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Player 1	50.52	48.16	19.48	17.78	15.16	14.45	1	3	0.2	0.2	2.71	2.08	0.90	0.83
Player 2	44.01	53.55	19.70	23.64	13.20	16.06	1	1	0.2	0.2	1.94	3.27	0.59	0.63
Player 3	39.20	59.67	15.09	29.54	11.76	17.9	3	1	0.2	0.2	1.41	4.48	0.50	0.50
Player 4	27.48	35.77	14.65	17.55	8.24	10.73	3	1	0.2	0.2	3.18	1.81	0.42	0.51
Player 5	20.82	87.35	7.84	37.59	6.24	26.20	2	3	0.2	0.2	1.12	5.34	0.29	1.71
Player 6	26.00	46.95	9.99	24.15	7.80	14.09	1	1	0.2	0.2	2.35	2.68	0.94	0.53
Player 7	**68.28**	28.06	**41.59**	10.64	**20.48**	8.42	**2**	2	**0.2**	0.2	**21.8**	3.84	**13.0**	0.38
Player 8	43.88	29.09	16.33	10.55	13.16	8.73	2	1	0.2	0.2	1.11	1.05	0.55	0.37
Player 9	35.78	42.20	14.47	18.73	10.73	12.66	2	1	0.2	0.2	2.07	1.74	0.46	0.51
Player 10	36.78	30.17	16.03	14.35	11.03	9.05	3	1	0.2	0.2	5.70	1.62	1.21	0.45
Total											21.8 (5.7)		1.26 (0.646)	

Incorrect tracked target, which must be re-initialised manually, is marked in **bold**. () shows resulting errors without this tracker

Table 5.5: Sequence complexity and results obtained (Option 1).

	Duration [sec]	Occlusions*		Length of Occlusions	Transitions between cameras				Lost Players	Mean Error [metres]
					Entry		Leaving			
		Total	Solved		Total	Solved	Total	Solved		
Seq. 1	11.625	2	2	24	9	9	9	9	0	0.40
Seq. 2	12.0	5	5	84	11	10	12	11	1	0.65
Total	23.625	7	7	108	20	19	21	20	**1**	**0.52**

*Only occlusions between players of the same team have been scored since colour models solve the others

Table 5.6: Long sequence results (Option 1).

	Duration		Error Rate	Saving Rate	Conflicts		
	Time	Frames			Total	Solved	Rate
Long Seq.	7min 46sec	11661	5.84%	99.37%	759	608	80.1%

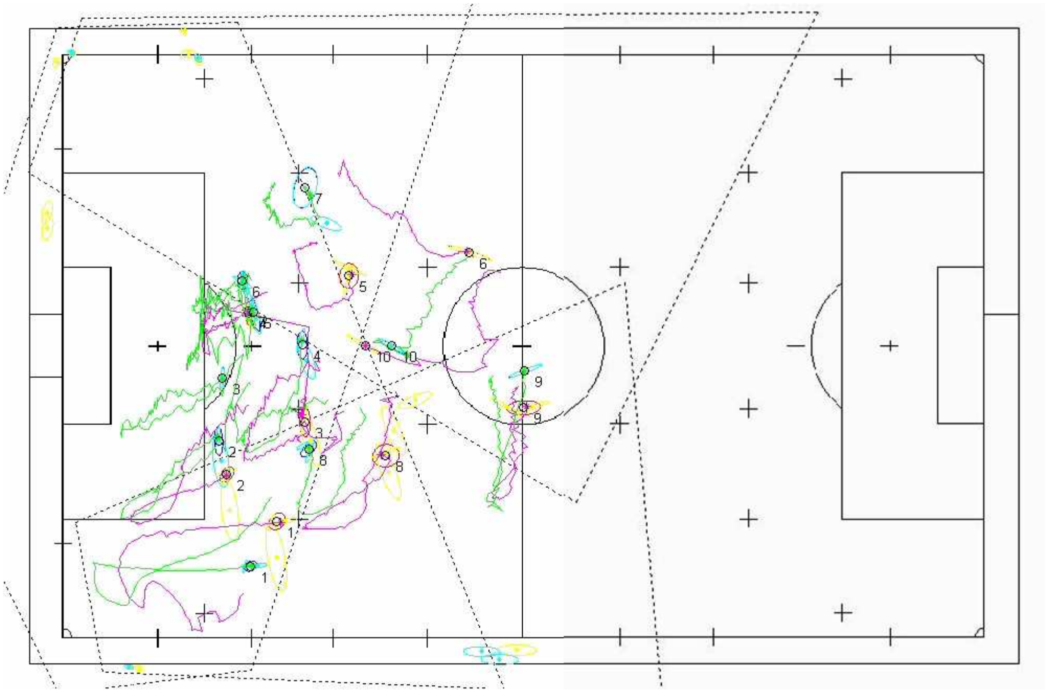


Figure 5.19: Trackers and trajectories during a period of the football match (Sequence 2).

human user must do to complete the annotation of the whole match. Finally, the conflict rate consists in the number of solved potentially dangerous situations divided by all these situations. Therefore, the system involves a substantial improvement with regard to current commercial systems, which solve most of the conflicts manually.

If we analyse the unsolved errors in depth, we can see the distribution of these errors (Fig. 5.22). Most of them are situated in zones 1, 2 and 3 due to the poor resolution of the players in these areas for all the cameras. As future improvement, it would be advisable to place three extra cameras in order to increase the system redundancy, reducing thus the error rate. The importance of this redundancy is also corroborated by the conflicts distribution: the 57.32 percent of conflicts appear when only one camera could be used to obtain useful observations (due to occlusions, extremely low resolutions or compression losses), and the density of conflicts per metre is reduced from 2.44 to 1.98 when these conflicts are observed by more than one camera. A more detailed study which remarks the importance of the camera overlapping is explained in Appendix F.

5.4.3 General Scheme Option 2: Unique Filter on the Plan Reference

This second version shares many characteristics with the previous one, such as importance function, observation process or posterior probability. The main difference resides in the global estimation and the conjugation of the information from the different cameras.

Figure 5.23 shows the projections of the particles coming from different cameras in the same reference, and the probability distribution that they form. An ellipse remarks the uncertainty of the resulting estimation on the plan. The third image shows an occlusion and how the clustering allows maintaining both trackers separated.

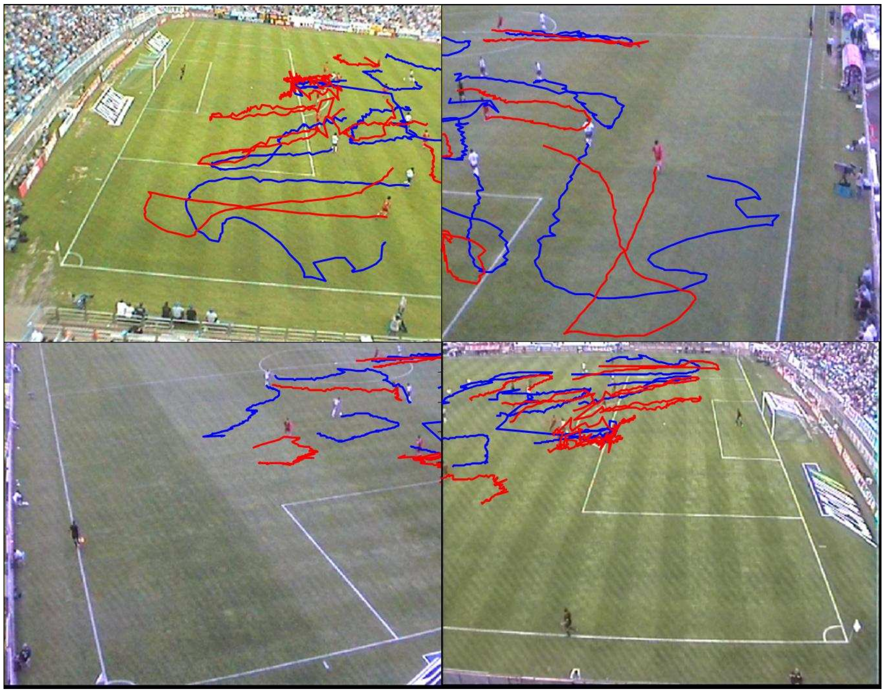


Figure 5.20: Trajectories at each view during a period of the football match (Sequence 2).

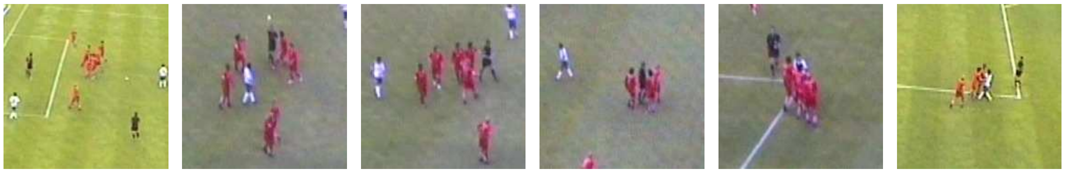


Figure 5.21: Complex situations that appear during a football match.

Similar results are obtained for this second option regarding the first one. Results for sequence one can be seen in Table 5.7 and Figure 5.24.

5.4.4 Comparison Between Option 1 and 2

Even though both methods give encouraging results, the first option apparently seems a better paradigm due to a better accuracy and a higher robustness. This last assertion is corroborated by the fact that this algorithm has the property of maintaining a slightly different target parameters for each camera. However, the joint estimation is unique. This methodology allows the system to recover from a severe failure in one or several cameras, which can be easily detected by means of a distance between the camera and the joint estimations. An example is depicted in Figure 5.7.

However, the second alternative has an advantage which should not be discard. It allows keeping the multi modality until the calculation of the global estimation. Although the first option models the uncertainty before projecting the estimation on the plan, this process implies a gaussian assumption in each camera. That is one of the reasons for having two different

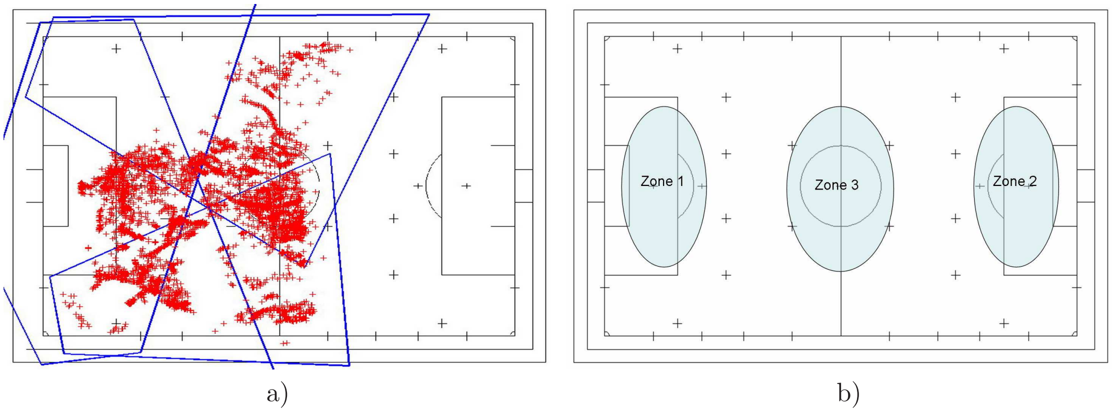


Figure 5.22: a)Conflict distribution. b)Main zones of unsolved errors

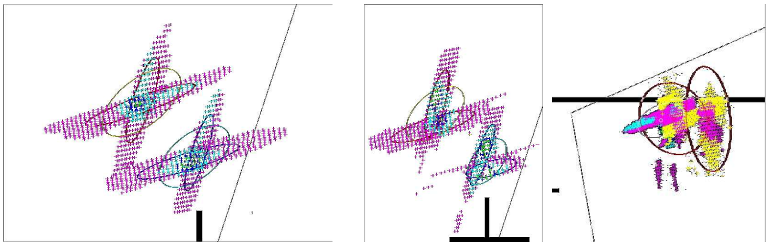


Figure 5.23: Examples of particles projected on the plan and extracted estimation.

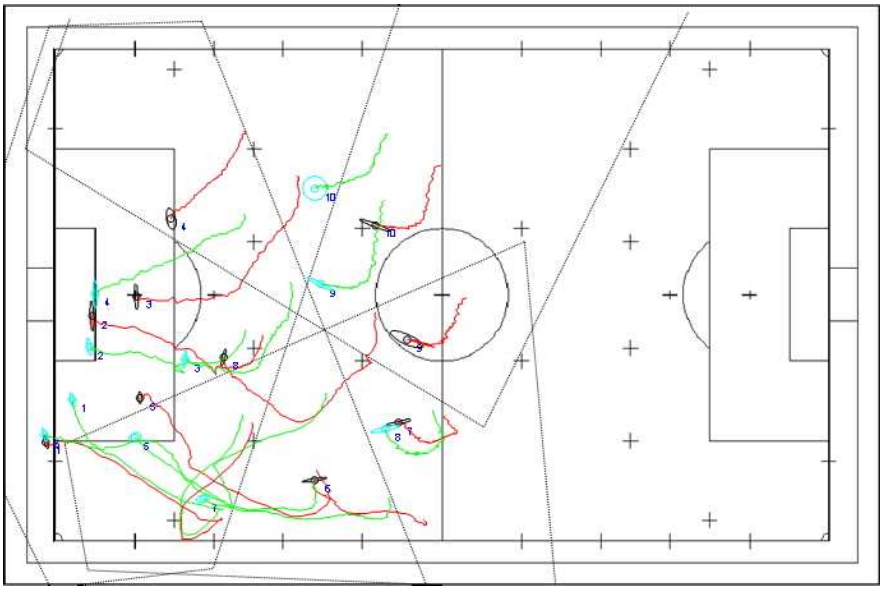


Figure 5.24: Trajectories at each view during a period of the football match (Sequence 1).

Table 5.7: Sequence complexity and results obtained (Option 2).

	Duration [sec]	Occlusions*		Length of Occlusions	Lost Players	Mean Error
		Total	Solved			
Seq. 1	11.625	2	2	24	0	1.0456

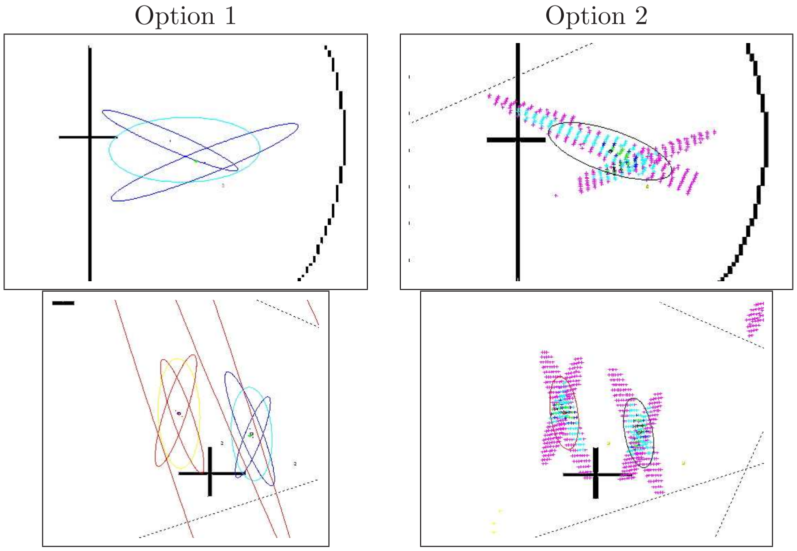


Figure 5.25: Comparison of the estimation obtained with both techniques in two particular situations.

tracking algorithms: PF obtains a very good quality tracking on the images and maintains the multi-modality for the next frames, while representing the measurements by their Gaussian approximations allows us to perform a quick and reliable data association of the measurements with trackers, applying an UKF, and combine the information from different sensors.

Avoiding the Gaussian assumption might provide a better estimation. Nevertheless, the theoretical improvement is not reflected on the results because the clustering algorithm and the association technique applied to deal with occlusions on the plan introduce similar assumptions. More effort must be applied in order to obtain a method which allows a compromise between computational complexity and solution of occlusion. In addition, it is worthy to note the relative importance of the gaussian assumption of the first option. It is only temporary, in order to obtain the provisional estimation, since the particles, and therefore the multi-modality, are propagated to the next time step.

Figure 5.25 shows the estimation of the location and error obtained with both methods. Although similar state parameters are obtained, a more accurate uncertainty representation is depicted.

As extra requirement of the first option, an algorithm to model the transition between cameras, capable of creating and destroying trackers is required, as pointed in Section 5.4.2.7.

5.5 Conclusions

In this chapter we have presented techniques to extend the efficiency obtained in previous chapters to multi-target tracking, specially useful when they share the same or similar appearance models. Techniques to model the interaction and remove the coalescence have been shown.

Nevertheless, the so-complex problem that this kind of tracking implies makes unfeasible the solution of highly interactive targets. Examples can be found in sports sequences. To address these applications, additional information should be included by the combination of several sensors or cameras. Although additional information helps to cope with multi-target tracking, it also introduces new challenges, such as sensor synchronisation or the development of a methodology for combining the observations.

In this sense, a computationally efficient multi-camera algorithm for tracking multiple targets in a sporting match has been presented as representative example. Most of the techniques presented in this thesis are encapsulated in this application to achieve positive results.

Therefore, we consider that ASTRO project implies a significant advance in sports application. It bases its performance on the combination of cameras. The robustness of multi-view methods lies in the multi-view integration which makes the method insensitive to occlusions in some of the cameras. However, a sporting match is an environment so complex that it is not enough in many situations. For this reason, we have developed a double tracking methodology combined with a feedback procedure. This technique increases the strength of the system with a feasible computational cost, thanks to several efficient modifications of the particle filter.

Results obtained are satisfactory, and involve a substantial improvement with regard to current commercial products. Our system allows processing a complete match in a few hours. In this way, a football trainer will have the match at one's disposal in order to analyse it carefully and extract conclusions. The fact of obtaining results as soon as possible means an important competitive advantage in an increasingly competitive environment.

As future work, it would be important to develop a method to automatise the initialisation of the players on the map and the generation of the colour models. Furthermore, we must improve the data association algorithm in very complex situations (fouls, corner kicks) while we maintain the multi-modality and a reasonable computational cost. Thus, we are working on a better relationship between the particle filter and the plan in order to obtain a better estimation of the uncertainty and, therefore, improving the multi-camera integration.

Finally, the hardware which composes the system is in a renovation process to incorporate megapixel high-resolution digital camera, connected by a Gigabit Lan. In this manner, better observations could be obtained, decreasing the number of errors of the global system.

6

Articulated Model Tracking

"Vision looks inward and becomes duty. Vision looks outward and becomes aspiration. Vision looks upward and becomes faith."

-Stephen S. Wise-

Articulated tracking is the most complex tracking domain. It implies not only estimating the global location of the body, but also the relative movement of the regions that compose the model. Therefore, an extra degree of complexity is introduced with the consequent increase of dimensionality. In order to address the problem, prior knowledge of the movement and the target's morphology must be introduced. While the observation process or the tracking filter were the cornerstones in the previous domains, the model is basic in this level of understanding to achieve a successful result.

Human motion modelling can be defined as the ability to estimate, at each frame of a video sequence, the positions of each joint of a human figure which are represented by an articulated model. The subject is considered as a complex object, articulated or deformable. Their applications include analysis of human activity, entertainment, ambient intelligence and medical diagnosis to name a few.

A compromise between simplicity, generality, accuracy and robustness has to be achieved. Although the more accurate and detailed the prior information is, the better pose estimation is obtained, this is not our objective. This thesis is focused on surveillance, with all the implications that this entails. While tracking methods based on 3D anthropomorphic articulated models have proved the most effective, they imply a high computational cost not suitable for real time and generally rely on data capture synchronously by several cameras which have been accurately calibrated.

Therefore, we have chosen simple models, such as 2D models or stick figures. These models are not comparable with 3D volumetric models used in pose estimation and animation, but are accurate enough for activity recognition and robust enough for surveillance applications. Unlike solutions whose constraints restrict the type of sequences they can handle, our approach is based on an constrained 2D model designed to tackle 3D motion patterns such as changes in the pose of the object with respect to the camera. Thus, they should be able to deal by themselves with variations in rotation and depth, that is, with the intrinsic ambiguity of projected 3D postures, self occlusions and distortions introduced by camera perspective. To achieve this without introducing strong motion constraints which would restrict the application of our system, we propose to use some specific knowledge about biomechanics and human gait analysis. Because of the high complexity of this task, we assume just a person to be evaluated or at least a scenario where the different subjects do not interact or occlude each other. These situations will be consider as future work.

In this chapter we will propose two different techniques to extract and track an articulated model. As a requirement, the proposed methods should be able to recover human poses with data from a single uncalibrated camera. The first one is based on the use of morphologic information about the human shape. By extracting the silhouettes of the target and applying a model to establish correspondences with predefined human poses, a stick skeleton with the pose is obtained.

The second approach consists in an articulated set of patches. They join in a set of connection points with a higher or lower degree of freedom. The layout of the patches and connections help the system to introduce constraints which simplify the problem. Finally, results and conclusion for each proposal are shown.

6.1 State of the Art

Although the location of a person as well as its trajectory are crucial information in tracking and human motion analysis, the motion of joints provides vital information to motion estimation and recognition of the whole figure. Therefore, the location of the limbs of the human body must be extracted if a deeper analysis is required.

This domain does not discard the whole body tracking. Quite the contrary, the motion of the whole body can be employed as a guide for estimating the movement of body parts. For that reason, a combination of both methodologies is usually employed, being the human body tracking only responsible for estimating the relative motion of the human parts.

Approaches to vision-based human motion analysis can broadly be divided into generative and discriminative. The first category explicitly uses a human body model [O'Rourke and Badler, 1980; Hogg, 1982; Rohr, 1994; Rehg and Kanade, 1995; Gavrila and Davis, 1996; Kakadiaris and Metaxas, 1996; Deutscher et al., 2000] that describes both visual and kinematic properties of the human body. Discriminative approaches [Elgammal and Lee, 2004; Li et al., 2006; Safonova et al., 2004; Rahimi et al., 2005; Urtasun et al., 2005a; Urtasun et al., 2006; Urtasun et al., 2005b; Sminchisescu and Jepson, 2004; Hou et al., 2007; Lee and Elgammal, 2006] learn the mapping from image space to pose space directly from carefully selected training data. While this may improve the pose recovery results, it comes at the cost of computational complexity. In discriminative approaches, variations can either be encoded implicitly, like variation in body size dimensions, or explicitly, like viewpoint. The fact that not all parameters can efficiently be encoded explicitly causes discriminative approaches to perform less accurately compared to generative work. The reason is simple, they only work for the trained motions: any other pose will cause an error which grows with the difference between the sample and the training dataset. However, discriminative approaches are computationally much less expensive, can potentially be applied in real-time and are more robust to noise or occlusions. Furthermore, the discriminative approaches allow us to recover the pose with less information, being more suitable for monocular application

- Generative approaches:

  The motion in these approaches is constrained by body kinematics and dynamics as well as the dynamics of the activity being performed. Such constraints are explicitly exploited to recover the body configuration and motion in model-based approaches, like [O'Rourke and Badler, 1980; Hogg, 1982; Rohr, 1994; Rehg and Kanade, 1995; Gavrila and Davis, 1996; Kakadiaris and Metaxas, 1996; Deutscher et al., 2000], through explicitly specifying articulated models of the body parts, joint angles and their kinematics (or dynamics) as well as models for camera geometry and image formation. Therefore, the generation model models the data generation process, although is also possible that it can learn from unlabeled data.

- Discriminative approaches:

  The huge number of parameters which model the human motion, in conjunction with the multiple degrees of freedom of an articulated object, as the human body is, make useless the efforts of tracking algorithms to recover the human pose. This is due to the fact that tracking algorithms are inefficient managing high dimensionality spaces. The probability of falling in a local minimum is high in these situations. Nevertheless, as it has been shown in the literature [Elgammal and Lee, 2004; Grochow et al., 2004; Safonova et al., 2004], the space of possible human motions is intrinsically low-dimensional and several techniques can be implemented to reduce the dimensionality.

Among the discriminative approaches, many methods have been implemented, like PCA [Ormoneit et al., 2005]. However, the human pose space, by definition, is highly non-linear. Because PCA can only learn linear subspaces, the nonlinearity of human motions is often not well modelled in a linear subspace, and it will not be able to effectively reduce the dimensionality of complex and highly non-linear motion data. Nonlinear dimensionality reduction algorithms such as Isomap [Tenenbaum, 1998] and Local Linear Embedding (LLE) [Roweis and Saul, 2000] can be used to find more effective low dimensional embedding of motion data [Wang et al., 2003; Li et al., 2006; Elgammal and Lee, 2004; Rahimi et al., 2005] than PCA. However, these techniques do not provide a direct method to come back to the human pose space, which is necessary if we want to recover the pose from a sequence. Lately, a number of existing dimensionality reduction methods provides inverse mappings, such as Charting [Brand, 2002], Locally Linear Coordination (LLC) [Teh and Roweis, 2003] and the Gaussian Process Latent Variable Model (GPLVM) [Lawrence, 2005].

Regarding to the modelling, the structure of the human body can be addressed by the location and motion of the torso, the head and the four limbs. It has been represented in many different ways, being 2D contours, sticks, articulated patches and volumetric models the most common ones.

Before going on with the description of the human model based tracking, it is worthy to discuss about the two lines to take into account on analysis of human body parts: 2D versus 3D approaches. The first option involves working directly with the 2D features derived from the images, using discriminative approaches or generative ones. Both alternatives have been successfully applied for constrained types of movements, such as lateral view i.e. walking parallel to the plane of the image or simple and periodic motion. Nevertheless, their performance decreases significantly for unconstrained and complex human actions, with movements out of the plane of the camera (wandering, making gestures, turning, ...) and under the presence of incessant self-occlusions. Even though 2D discriminative approaches allow us to cope with self-occlusions in a more robust way, they only work if some a priori knowledge about the movement or the viewpoint is assumed, in order to apply the correct 2D model. They are indeed the best and easiest solution for several application in which the computational cost, the environment or the resolution are a limitation.

On

the other hand, 3D methods [Deutscher and Reid, 2005; Felzenszwalb and Huttenlocher, 2005; Lee and Cohen, 2003; MacCormick and Blake, 2000; Sminchisescu and Triggs, 2001] can be considered a more general purpose approach. They do not have the problem of being an ill-posed technique and enable to take advantage of a large available a priori knowledge about the kinematics, shape properties and biomechanics of human body and human gait. All this prior information makes the problem tractable and permits to predict events such as self-occlusions. On the contrary, they do not have a direct access to the 2D features of the images and they must be projected into the images. This fact have two consequences: the first one is that the projection, in addition to the larger dimensionality of the model, make the 3D tracking a computationally expensive methodology, not suitable for real time tracking. The second one is the necessity of a constrained environment, where all the cameras have been calibrated and the transformation between the images and the 3D world is known. Therefore, the initialisation process is much more complex in comparison with 2D models. This second limitation can not be available for some application like video surveillance field, where the idea is re-utilise the cameras already installed.

Therefore, although 3D tracking has obvious advantages in principle over 2D, last ones do not have to be discard in advance, because of their possible application in certain fields.

6.1.1 Stick Figure

A stick model represents the human body as a combination of line segments linked by junction points. The stick figure is probably the simplest representation of the human body, but it is enough to provide a key to activity recognition and human motion analysis. This concept was introduced by Johansson [Johansson, 1975] who marked the joint points as moving light displays. This idea was improved by Rashid [Rashid, 1980] who presented a method capable of recovering a human structure by assuming the points connected and belonging to the same object. Unluckily, these methods imply the use of markers, which avoids any kind of segmentation problem, but constraints their use to non-real environment and artificial conditions.

Many different non-marked based methods have been used to approximate a human being by a set of sticks. Guo et al. [Guo et al., 1994] represent a human by a ten stick 6 joint articulated model. Joint motion is predicted to reduce the complexity of the estimation, and angle constraints are introduced to avoid impossible poses. The final estimation is obtained minimising the energy in a potential field. Karaula et al. [Karaulova et al., 2000] propose a novel hierarchical model of human dynamics for view independent tracking of the human body in monocular video sequences. Dynamics are encoded using Hidden Markov models and kinematics using Hierarchical Principal Component Analysis. The top level of the hierarchy contains information about the whole body, the lower levels about each subpart of the body. Cheen and Lee [Lee and Chen, 1984] recover the 3D stick configuration of a moving subject composed of 17 sticks and 14 joints. All the possible configurations are projected in 2D to check their feasibility, which implies a huge computational cost. Baratkumar et al. [Bharatkumar et al., 1994] model the legs with a stick skeleton. Medial axis transformation extracts the 2D stick figures of the legs, although the method is sensitive to change of the viewpoint. Huber [Huber, 1996] proposed a flexible stick model, where the segments have a certain degree of relaxation like a spring. The skeleton is fitted to the image using motion and stereo measurement, which are confined in a space of possible motions. Iwasawa [Iwasawa et al., 1997] extracts the stick figures in monocular and pre-calibrated thermal sequences. The torso is firstly fitted using the main axis of the silhouette and then the extremes of the limbs are heuristically located as the farthest points. Finally, articulated joints are obtained by using genetic learning. As drawback, the system is view- and gesture-dependent.

6.1.2 Silhouettes

This representation models the shape of the human body as a projection in the image plane. The goal is to obtain the bounding contour of the object and keep updating it dynamically over time.

This methodology has many links with region-based tracking and it can be seen as an extended version of regions. Just like regions, one can keep tracking even in the presence of partial occlusions. Contours allow us to track not only articulated or rigid objects but also deformable targets. Moreover, information related with the pose can be estimated thanks to the external silhouette. In contrast to region tracking, active contours reduce the computational complexity. However they require a good initialisation and a model which should be trained in an off-line stage.

Isard and Blake [Isard and Blake, 1996] employ the Condensation algorithm, in combination with deformable templates, in order to model complex movements. An edge detector is used to obtain the likelihood function. Impressive results are obtained even in presence of cluttered backgrounds. Paragios and Deriche [Paragios and Deriche, 2000] present a new variational framework for detecting and tracking multiple moving objects. Motion detection is performed using a statistical framework for which the observed interframe difference density function is approximated using a mixture model. By employing a geodesic active contour function, tracking and detection are addressed together. Peterfreund [Peterfreund, 1999] explored active contours based on Kalman filter for tracking of non-rigid objects, like people. Proposed active contours combine gradients and optical flow measurements along the curve to improve robustness to clutter and occlusions.

Contours are, therefore, a valid method for non-rigid object tracking which give better results than traditional method against clutter and occlusions. In addition, they provide a way to estimate the pose of the subject. Thus, they can be included just in the frontier between tracking and activity recognition. However, for a more detailed and deeper analysis of these disciplines, they have been shown insufficient, being articulated models much more useful.

Some works trying to join both worlds have been published. In the work by Leung and Yang [Leung and Yang, 1995], two processes are applied. The first process uses a novel technique to extract the outline of moving objects. The second process interprets the outline of different types of posture and produces a 2D human body stick figure for each frame of the image sequence. A new human body model is used in the second process. The model provides information of the human structure, shape and posture for the labelling process. Niyogi and Adelson [Niyogi and Adelson, 1993] show a method for recovering a stick model of a human by spatiotemporal analysis of gait patterns. A walker is a translating blob which has braided spatiotemporal patterns in the lower half of his body. By recognising these spatiotemporal signatures, a model for subsequent spatiotemporal analysis is imposed. The contour of a walking subject was outlined by utilising joint trajectories, and a more accurate gait analysis was carried out using the outlined 2-D contour for the recognition of human beings.

6.1.3 Planar Patches

In the same way that contours can be seen as an extension of region-based tracking approach, the same happens with planar patches. In this case, each patch can be assumed as a region with its own properties. The difference consists in that the relations among regions can change dynamically by means of the inclusion of new parameters in the model.

Ju et al. [Ju et al., 1996] propose a cardboard model in which the human limbs are modelled by a set of connected planar patches. By constraining the parameterised motion of the patches in the image, the articulated motion is reinforced. Optical flow is used as feature to track the limbs as well as to estimate the viewpoint. Results confirm that 2D patches are able to track a limb without occlusions and once the viewpoint has been determined.

Rehg et al. [Rehg et al., 2003] describe a two-dimensional scaled prismatic model (SPM) for figure registration, which deals with variations in rotation and depth. SPM reduces significatively the number of singularities that appear due to the bidimensional projection of the 3D pose and does not require detailed knowledge of the 3D kinematics. Although they demonstrate the application of the model for motion capture from movies, only certain types of movements can be tracked and the system fails for quick movements, being unable to recover the right pose.

In [Wu et al., 2003], an approach that analyses subparts locally is proposed for visual tracking of articulated models while reinforcing the structural constraints among different subparts. It combines a dynamic Markov network which characterises the dynamics and the image observations of each individual subpart as well as the motion constraints with a Mean Field Monte Carlo (MFMC) in which a set of low dimensional particle filters interact with each other and solve the high dimensional problem collaboratively. In [Mcallister et al., 2002], Random Sample Consensus (RANSAC) and Maximum Likelihood Estimation Sample Consensus (MLESAC) algorithms are incorporated in a planar patch tracker like feature weights to perform robust tracking. In [Noriega and Bernier, 2007], authors propose a planar-patch articulated model, which is a loose limbed model, including attraction potentials between adjacent limbs and constraints to reject poses resulting in collisions. The compatibility between model and image is estimated using one particle filter for each limb and the compatibility between limbs is represented by interaction potentials. The joint probability is obtained by belief propagation on a factor graph.

6.1.4 Volumetric Models

To solve the intrinsic drawbacks of 2D models, many authors try to depict the geometric structure of human body using 3D models such as ellipses, cylinders, truncated cones and/or spheres.

In one of the first works about 3D volumetric tracking, O'rourke and Badler [O'Rourke and Badler, 1980] propose a 24 segment model covered by overlapping spheres. The location of the body parts are predicted and located carefully into the image. After that, a parser and a simulator check this fitting with linear function and knowledge about the human body.

Rohr [Rohr, 1994] uses 14 elliptical cylinders to model a human body. Eigenvector line fitting is applied to outline the subject in the image. Afterwards, the similar distance measurement is used for projecting the 2D shape in 3D. Hogg [Hogg, 1982] applies the same model with a computer program to recover the 3D description of a human walking. In a work by Wachter and Nagel [Wachter and Nagel, 1999], a correspondence between a 3D elliptical cone model and the real image of a monocular sequence is established. As tracking algorithm, an iterative extended Kalman filter is used, and both edge and region measurements are incorporated in the likelihood function to determine the degrees of freedom of joints and camera orientations. Akita [Akita, 1984], Perales and Torres [Perales and Torres, 1994] applied a combined stick and volumetric model to process the sequence in a coarse to fine manner. Stick model indicates the motion and relationships between the body parts and volumetric shapes provide knowledge of the external body shape.

Some researches have been focused on the recovery of specific body parts. This can be useful to recognise gestures and, for example, identify the symbols of the sign language, if we are tracking a hand. Rehg [Rehg and Kanade, 1995] is interested in tracking the fingers of a hand fitting a 3D hand composed of cylinders. Goncalves et al. [Bernardo et al., 1996] track a human arm in 3D in a calibrated environment. The model is composed of truncated cones and spheres. By projecting the shape of the arm and comparing with the real image, the location can be extract by error minimisation in an iterative process. Recently, Bray et al. [Bray et al., 2007] have combined particle filtering and a SMD optimisation method to produce smart particles to track high-dimensional articulated structures (hands in their case) with far fewer samples.

Kaadiaris and Metaxas [Kakadiaris and Metaxas, 1996] consider a multi camera system to cope with 3D model-based body part tracking. Kalman filter is applied to predict the location of each limb. The correspondence between the contour in the image and the projection of the 3D shape is used as likelihood function. Gavrila and Davis [Gavrila and Davis, 1996] extended this methodology to a 22 degree-of-freedom model. Matching between the model and the actual image is done based on the chamfer distance. Hunter et al. [Jain, 1997] build a model composed of 5 ellipsoids with 14 degrees of freedom. An Expectation-Maximization algorithm solves the mixture estimation of the low-level segmentation images.

The development in computation machines has allowed combining the particle filter with 3D articulated models. Even though this combination does not always work in real time, it is possible its use in pose recovery applications. Probably one of the most important papers in this field is the one presented by Deutscher, Blake and Reid [Deutscher et al., 2000]. It is not only the most important generative approach but also the reference method to compare all the new algorithms. A modified version of particle filter is proposed for estimating efficiently the multi-modal distribution of human body articulated model in a huge dimensional space. By dividing the estimation process of the posterior probability in a set of layers, the final estimation is correctly modelled with a reduced number of particles even in the presence of narrow peaks. Motion and edges are used as measurements. The main drawback is the huge computational load for processing each frame. Sidenbladh, Black and Fleet [Sidenbladh et al., 2000] present another relevant probabilistic method for tracking 3D articulated human figures in monocular sequences. It is based on a generative model of appearance, a robust likelihood function which works out a gray level difference, and a priori probability distribution which introduces knowledge about human gait and joint angles. Particle filter is the paradigm used for predicting and propagating the hypothetic poses. Thus, the posterior probability distribution over model parameters can be modelled in spite of being multi-modal because of kinematic singularities, depth ambiguities, occlusion and ambiguous image information. The valid 3D human motion is constrained by prior probability distribution over the dynamics of the human body.

Recently, discriminative approaches based on latent spaces and manifolds have achieved a high popularity. This is mainly because they reduce the computational cost by constrained the space of possible poses with prior information. Elgammal [Elgammal and Lee, 2004] proposes a manifold to relate silhouettes with 3D poses. A different 1D manifold is learned per view and activity. The framework is based on learning three components:

1. Learning Manifold Representation: using nonlinear dimensionality reduction, an embedding of the global deformation manifold that preserves the geometric structure of the manifold is obtained. Given such embedding, the following two nonlinear mappings are learned.

2. Manifold-to-input mapping: a nonlinear mapping from the embedding space into visual input space.

3. Manifold-to-pose: a nonlinear mapping from the embedding space into the 3D body pose space.

In more recent works, some authors have employed Gaussian Process Latent Variable Models (GPLVM) [Lawrence, 2005; Urtasun et al., 2005a; Urtasun et al., 2005b; Urtasun et al., 2006;

Tian et al., 2005; Hou et al., 2007] as tool for reducing the dimensionality of the feature space, making easy the modelling of different kinds of movements. One of the advantages of this method is that it provides an efficient probabilistic method and allows estimating the uncertainties of the low dimensional embedding.

In [Sminchisescu and Jepson, 2004], two different regression algorithms are used for the forward mapping (dimensionality reduction) and inverse mapping. The representatives used in the regression are chosen in a heuristic manner. In [Urtasun et al., 2005b], GPLVM and a second order Markov model are used for tracking applications. The learned GPLVM model is used to provide model prior. Tracking is then done by minimising a cost of 2D image matching, with the negative log-likelihood of the model prior as the regularisation term. Both [Sminchisescu and Jepson, 2004] and [Urtasun et al., 2005b] advocate the use of gradient descent optimisation techniques; hence, the low-dimensional space learned has to be smooth. An alternative approach [Tian et al., 2005] employs the GPLVM in a modified particle filtering algorithm where samples are drawn from the low-dimensional latent space. The GPLVM model in this case is used as a good non-linear dimensionality reduction algorithm. The smoothness enforced in the low-dimensional space by the learning algorithms in these three papers works well for tracking small limb movements, but may fail when large limb movements occur over time. In case of using gradient descent optimization techniques, good initialisation is required for the success of such techniques.

The strength of the GPLVM (global smoothness) may be its weakness. As GPLVM ensures temporal smoothness, but it may learn an over-smoothed density function and consequently fail to capture large pose changes over time. This is due to the fact that GPLVM is a static model; it has no intrinsic dynamics and does not produce smooth latent paths from smooth time-series data. Thus, even with an additional dynamical model, our GPLVM-based people tracker often fails due to anomalous jumps in the latent space and to occlusions. Another solution to include the temporal coherence is the Gaussian process dynamical models (GPDM). The GPDM is a latent variable dynamical model, comprising a low-dimensional latent space, a probabilistic mapping from the latent space to the pose space, and a dynamical model in the latent space [Urtasun et al., 2005a]. While GPDM has advantages over GPLVM, usually producing much smoother latent trajectories ,it can still produce large gaps between the latent positions of consecutive poses.

6.2 Silhouette Tracking

In this section, we propose a Rao-Blackwellised Particle Filter for addressing the problem of human pose estimation and tracking. The advantage of the proposed approach is that Rao-Blackwellisation allows the state variables to be splitted into two sets, being one of them analytically calculated from the posterior probability of the remaining ones. This procedure reduces the dimensionality of the Particle Filter, thus requiring fewer particles to achieve a similar tracking performance. In this manner, location and size over the image are obtained stochastically using colour and motion clues, whereas body pose is solved analytically applying learned human models. Point Distribution Models (PDMs) are applied for obtaining analytic solutions to the articulated body model parameters. These models are able to estimate 2D body pose whatever the viewpoint.

Our contribution proposes a framework in which learning models are combined with non-linear non-gaussian tracking algorithms. While Bayesian filter allows modelling of multimodalities of the distribution, analytic solutions reduce the dimensionality of the problem. The pose variables are solved applying a set of view dependent Gaussian Mixture Models (GMMs), as in [Cootes and Taylor, 1997; Jaeggli et al., 2006], and each motion state is modelled by means of PDMs [Baumberg and Hogg, 1994; Bowden and Sarhadi, 2000]. RBPF combines the advantages of stochastic and deterministic methods in a unique framework.

In Section 6.2.1, we study in depth the PDMs and their application to human modelling, as well as different solutions to deal with the common problems that appear, such as change of view or fragmented observations. We propose a statistical model for detection and tracking of human silhouette and the corresponding 2D/3D skeletal structure in gait sequences. We follow a PDM approach using a Principal Component Analysis (PCA). The problem of non-lineal PCA is partially solved by applying a different PDM depending on pose estimation; frontal, lateral and diagonal, estimated by Fisher‘s linear discriminant. Additionally, the fitting is carried out by selecting the closest allowable shape from the training set by means of a nearest neighbor classifier. To improve the performance of the model we introduce concepts related to human gait analysis which take into account temporal dynamic to track the human body. The incorporation of temporal constraints on the model increases reliability and robustness.

Afterwards, in Section 6.2.2 the Rao-blackwellised framework which integrates the PDM with the space tracking is presented. We change the solution proposed in section 6.2.1 in order to a better integration in the probabilistic framework. Instead of a solution based on nearest neighbour, we choose a solution based on GMMs.

6.2.1 Active Shape Models

The present approach can be divided into two phases: an off-line stage to generate the shape model from a training set using them to extract the mean shape and variation modes applying Principal Component Analysis (PCA); and an on-line stage consisting in locating and fitting the model in the image, and correcting it to the closest plausible pose. The 2D silhouette of a moving human and the corresponding 3D skeletal structure are encapsulated within a point distribution model (PDM). Some previous works in learning deformable models for tracking human motion are applied in [Koschan et al., 2003; Baumberg, 1995; Foresti et al., 2000].

The fitting process begins with a roughly human figure detection based on image difference and background subtraction. Then, we determine the pose of the figure to apply a specific model depending on the view. Fitting is an iterative process, where the model uses suggested movement from control points to find natural resting place. Once we have the silhouette we obtain automatically from the training set the skeletal structure corresponding to the matched contour of the database.

6.2.1.1 Training

Our main goal is to construct a mathematical model which represents a human body and the possible ways in which it can deform. Point distribution models (PDM) are used to obtain a complete silhouette of a non-rigid object, as well as the corresponding skeletal structure.

The generation of a 2D deformable model of the human contour follows a similar procedure as described in [Koschan et al., 2003; Baumberg, 1995]. For the training stage we use the CMU Motion of Body (MoBo) database [Gross and Shi, 2001]. Good training contours, which were taken manually, are located combining two different grids (see Figure 6.1), one rectangular and other semicircular used for taking point between the legs, to extract 49 landmark points for the human silhouette. Other different kinds of grids were tested, such as rectangular or elliptical grids. However they were discarded due to their inability to extract the leg contours, which contain most of the information concerning the movement.

Simultaneously, we extract 13 points corresponding to a stick model. This fact allows us to establish a correspondence between silhouettes and skeletons, which will be used to extract the

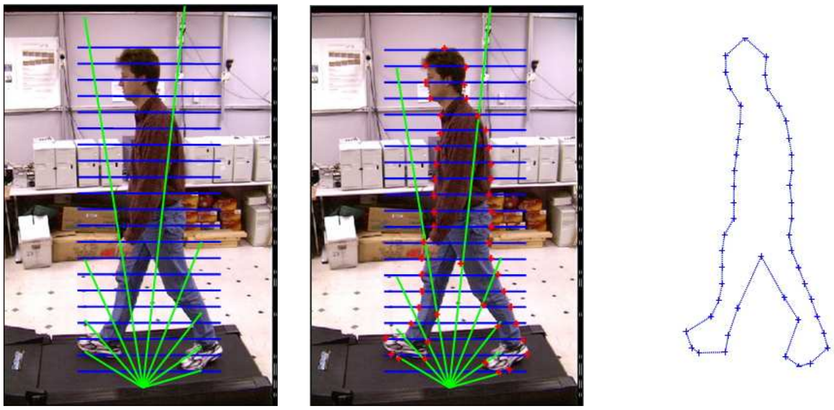


Figure 6.1: Training grid and contour extracted

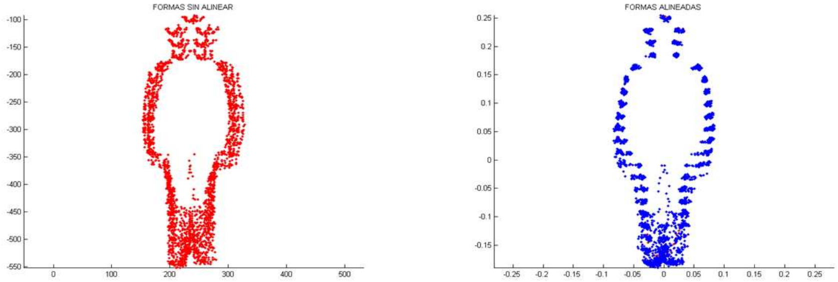


Figure 6.2: Frontalview contour dataset before and after the alignment

pose.

So, we obtain 3672 contours with their respective stick models, corresponding to 9 subjects in 4 different views (frontal, lateral, diagonal backwards and diagonal forwards). We align all the training set using Procrustes analysis [Goodall, 1991] to free the model of position, size and rotation. In this way, the PCA can extract the real variation of the contour and reduce the number of parameters. Each view must be independently aligned. Algorithm is explained in 10 and results in Figure 6.2.

6.2.1.2 Pose estimation

However, the possible shapes generated by the combination of the primary variation modes are not indicative of the training set due to their inherent non-linearity. In order to produce a model more accurate for practical applications, we divide the global training set in several cluster attending to the pose of the human figure. It is clear that the silhouette of a human figure taken from a frontal view is quite different from a lateral one. To reduce the non-linearities of a global PDM, we first estimate the most probable pose of the human view, and then we apply a specific PDM for this view, as depicted in Figure 6.3. The eigenvectors (Figure 6.4) catch the 98% percent of the set's variance.

We have chosen four different viewpoints: frontal, lateral, diagonal backwards and diagonal forwards, which encompass most of the silhouettes we can find in real situations. To carry out this pose classification, we first determine a global PDM taking into account all different views

Algorithm 10: Procrustes Analysis

Given a set of M contours $X = [x_1|x_2|\dots|x_M]$, where each contour x_j is a vector composed of n points $x_j = [\mathrm{x}^1, \mathrm{x}^2, \dots, \mathrm{x}^n, \mathrm{y}^1, \mathrm{y}^2, \dots, \mathrm{y}^n]$, the normalised set $\hat{x}_j$ is obtained as follows:

1. Location normalisation: The centroid of the mean contour $C = [\bar{x}, \bar{y}]$ is obtained and its value is subtracted to each training sample:

$$\bar{x} = \sum_{j=1}^{M}\sum_{i=1}^{n} x_j^i \qquad \bar{y} = \sum_{j=1}^{M}\sum_{i=n+1}^{2\cdot n} x_j^i \tag{6.1}$$

$$\dot{x}_j = \begin{cases} x_j^i - \bar{x} & \forall i \in [1, n] \\ x_j^i - \bar{y} & \forall i \in [n+1, 2\cdot n] \end{cases} \qquad \forall j \tag{6.2}$$

2. Scale normalisation: Every sample is scaled to a common size, usually with norm equal to 1.

$$\ddot{x}_j = \dot{x}_j / s_j \quad \text{where} \quad s_j = \|\dot{x}_j\| \quad \forall j \tag{6.3}$$

3. Rotation normalisation: The mean square error between the sample and a reference is minimised. Although a mean contour is usually used as reference, if there is a high variability in the dataset, it could be better to use one sample as reference.

$$S_{xx} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{x}_j^i \cdot \bar{\mathrm{x}}^i \qquad S_{xy} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{x}_j^i \cdot \bar{\mathrm{y}}^i \qquad S'_{xx} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{x}_j^i \cdot \mathrm{x}_j^i \tag{6.4}$$

$$S_{yy} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{y}_j^i \cdot \bar{\mathrm{y}}^i \qquad S_{yx} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{y}_j^i \cdot \bar{\mathrm{x}}^i \qquad S'_{yy} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{y}_j^i \cdot \mathrm{y}_j^i \tag{6.5}$$

$$\begin{bmatrix} \hat{\mathrm{x}}_j \\ \hat{\mathrm{y}}_j \end{bmatrix} = \begin{bmatrix} \frac{S_{xx}+S_{yy}}{S'_{xx}+S'_{yy}} & \frac{S_{yx}-S_{xy}}{S'_{xx}+S'_{yy}} \\ \frac{S_{xy}-S_{yx}}{S'_{xx}+S'_{yy}} & \frac{S_{xx}+S_{yy}}{S'_{xx}+S'_{yy}} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{x}_j \\ \mathrm{y}_j \end{bmatrix} \tag{6.6}$$

where $\mathrm{x}_j = \ddot{x}_j^i$ with $i = 1, \dots, n$ and $\mathrm{y}_j = \ddot{x}_j^i$ with $i = n+1, \dots, 2\cdot n$.

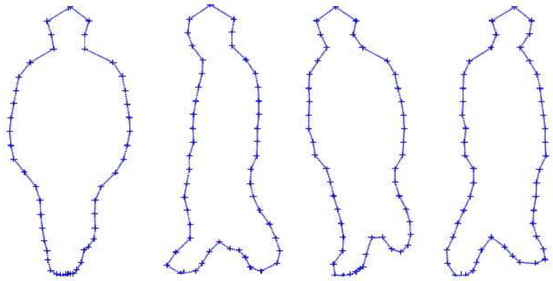


Figure 6.3: Mean contour for every view

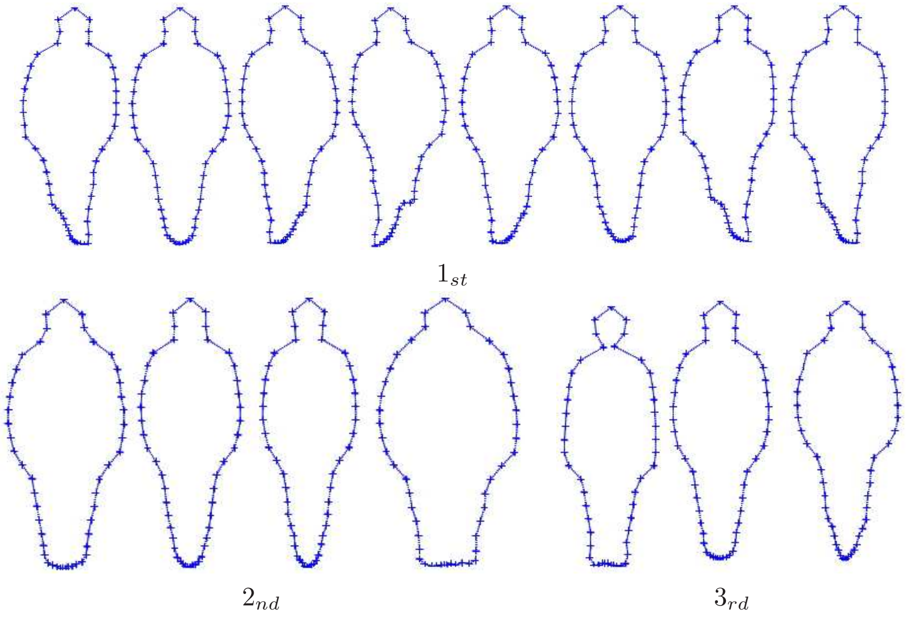


Figure 6.4: Three first frontalview eigenvectors

from the training data set. Better results are obtained if this PDM is generated just using the upper half of the body (head, shoulders and body). This is because the high variability of the legs introduce confusion.

Thus, once this global PDM has been fitted and the contour extracted, by means of a Linear Discriminant Analysis (LDA), we make a supervised (cluster-based) classification of the four distinct poses. To classify a new silhouette shape we use the Mahalanobis distance onto the LDA space (Figure 6.5)

We have tested the pose classification method with other images taken from the MoBo database obtaining a mismatch error lower than 2.78%. By introducing a temporal filtering that discards those changes shorter than 3 frames, the error decreases to 0.35%. This assumption is more than reasonable, since a person can not change its position with respect to the camera instantly due to the inertia.

6.2.1.3 Model fitting

As model fitting we call the process by means we find the position of the silhouette and skeleton points in a new image. It is made by matching the model generated off-line in the blobs detected in the current image. Then we determinate the pose of the human, so we can apply a specific PCA model to the blobs that have been found in the previous state. Finally, once we obtain an acceptable silhouette, the skeleton of the figure is located since we have correlated silhouette with skeleton in the training process.

Blob detection A widely used technique for separating moving objects from theirs background is based on subtraction and thresholding. Assuming the camera is stationary with fixed lighting conditions and good contrast, the method can be used to segment moving objects in a scene. In this approach an image $B(x, y)$ of the background is stored before the introduction of a foreground object. Then, given an image $I(x, y)$ from a sequence, feature

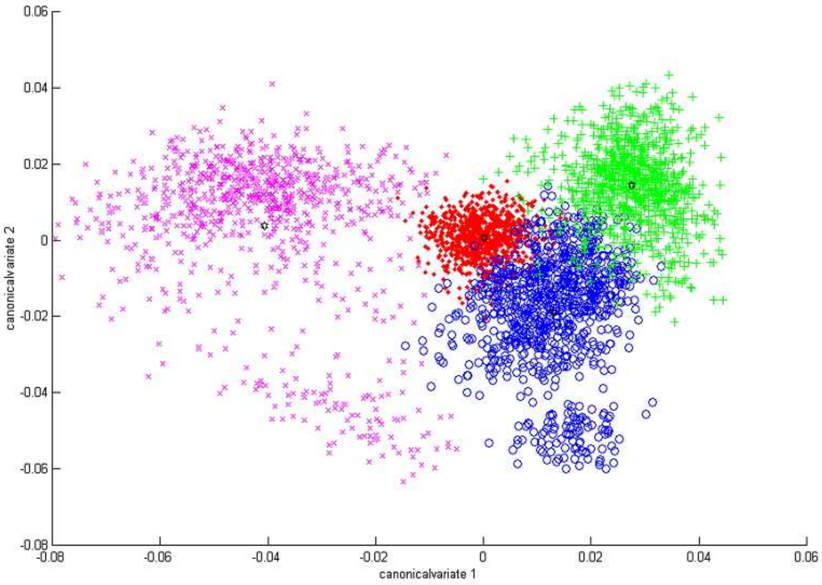


Figure 6.5: Dataset representation onto the LDA space. Red dots are the lateral view, green dots the forward diagonal, blue dots the frontal view and magenta ones the backwards diagonal.

detection of moving objects are restricted to areas of $I(x, y)$ where $|I(x, y) - B(x, y)| > \sigma$, where σ is a suitably chosen noise threshold. A shadow removal algorithm plus some morphological operations are used to improve the results and fill the holes (see Figure 6.6).

Afterwards, a vertical histogram (Figure 6.7) allows extracting the centroid avoiding the influence of the legs which could move this centroid horizontally. The bounding-box determines the size of the model.

Silhouette matching PCA model is used to constrain the shape of the PDM when applied to an image. The fitting process is an iterative one, where the model uses suggested movement from control points to find natural resting place. For that, perpendicular lines to the contour are drawn for every model point. The new location for each point is determined as the intersection of these lines with the blob. Movement of the model is allowed through the relocation of the model within the image plane using translation, rotation and scaling.

Deformation of the model is also permitted by finding the closest allowable shape as determined by the bounds of the mathematical deformation model. Given a new shape x, the closest allowable shape from the model is constructed by finding b such as: $b = \Phi^T(x - \hat{x})$. As pointed in [Foresti et al., 2000], each component of vector b can be clipped in a range given by $-3\sqrt{\lambda_i} \leq b_i \leq +3\sqrt{\lambda_i}$, obtaining vector b^*. Following this approach, the closest allowable shape can then be reconstructed as $x \simeq \hat{x} + \Psi b^*$. An example is shown in Figure 6.8.

Acceptable results are obtained, as it can be seen in Figure 6.9.

To this point, we have used a linear PCA; however, non-linearities are present in a model (see Figure 6.10) where parts of the contour display rotational characteristics in the image plane around some pivotal point. In Figure 6.11 we can see how impossible contours, which tend to appear when the blob is fragmented, are allowed by the PCA correction. This effect is

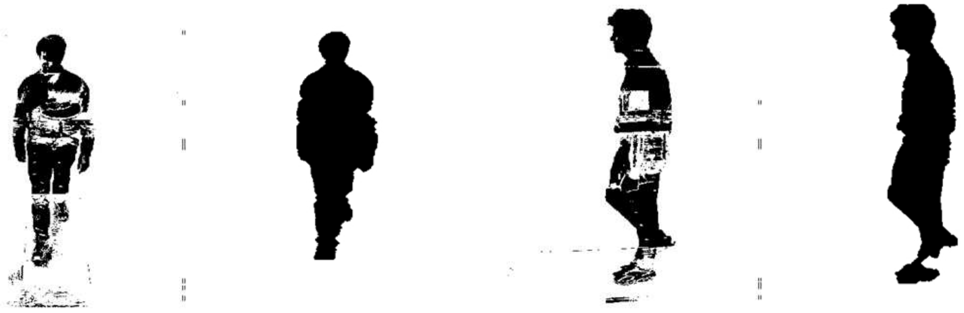


Figure 6.6: Blob detection and completion

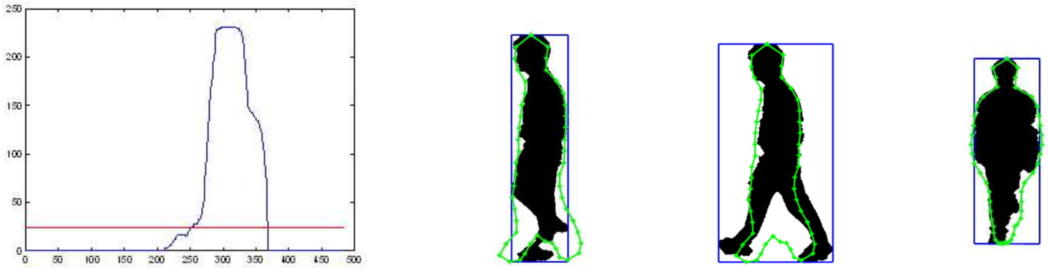


Figure 6.7: Vertical histogram and initialisation of the mean shape on the blob

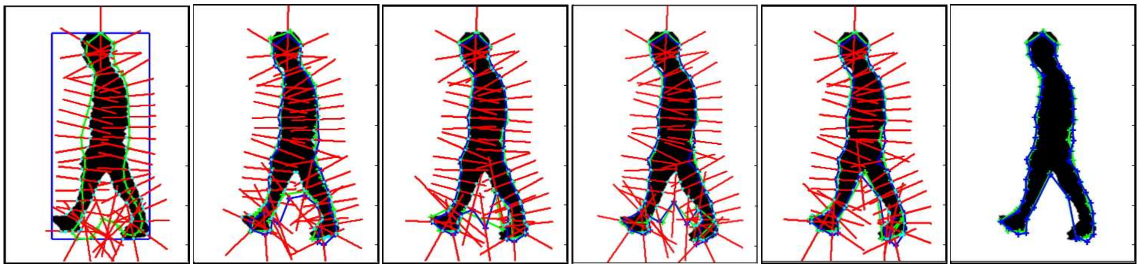


Figure 6.8: Iterative fitting process. Green contour is the result of moving the points along the normals (red lines). Blue contour is the closest allowable contour after limiting the maximum deformation using PCA

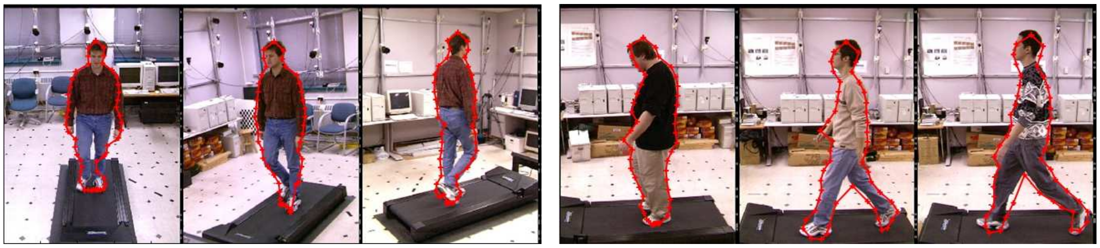


Figure 6.9: Results for different views and different subjects

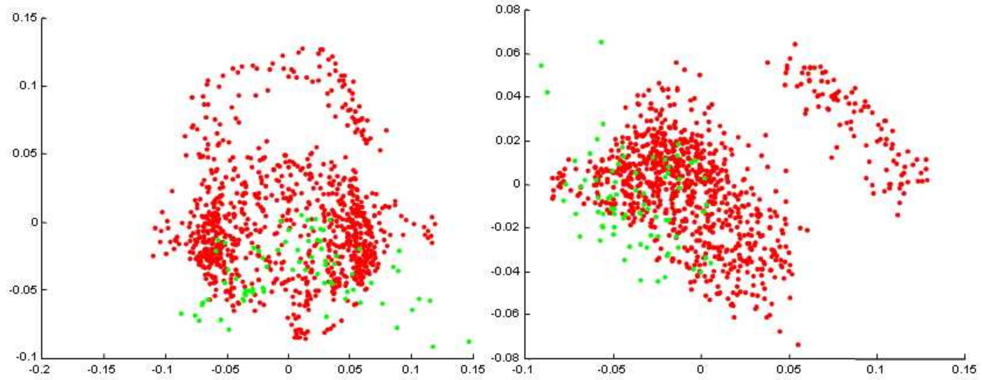


Figure 6.10: Frontal view 2 first feature space. Red dots are the training examples. Green dots are the erroneous contours

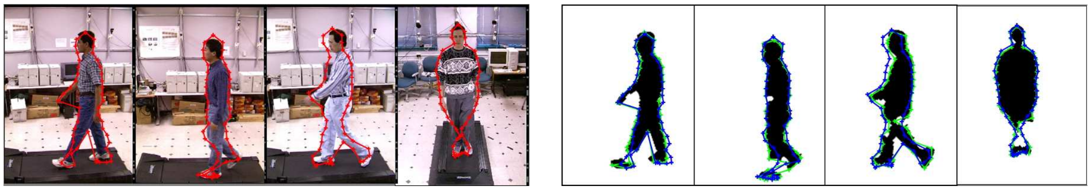


Figure 6.11: Results for fragmented blobs

encourage if we include the skeleton points into the model, due to the fact that they introduce a higher degree of non-linearity (see Figure 6.12). This point is quite important since our goal is to extract the skeleton to know the human pose.

To reduce this undesirable effect, several strategies are available: the usage of non-linear methods, a linearisation of the non-linear characteristic space by applying mixture of gaussians, etc... The first approach has been to use a Nearest Neighbour strategy. As mentioned previously, given a new shape x, the closest allowable shape from the model is constructed by finding b, but we select in each iteration the closest allowable shape from the training set by means of a nearest neighbor classifier. This technique always warranties acceptable contour determination, as it can be noticed in Figure 6.13, reducing noise effect errors in the blob detection. Later in this chapter, we will introduce another strategy based on MoG.

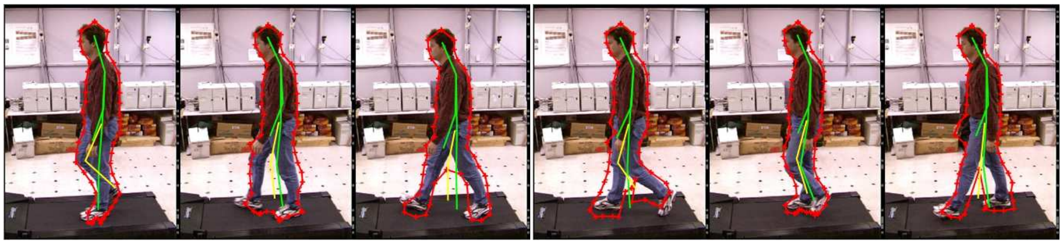


Figure 6.12: Results for a joint contour-skeleton model

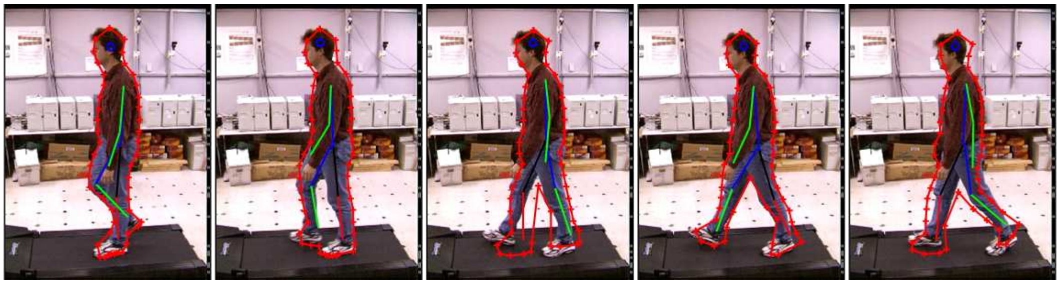


Figure 6.13: Results for a joint contour-skeleton model using Nearest Neighbour

Selective thresholding using anthropometric information Blob detection is the most critical state of the model fitting. The selection of a suitable threshold is a delicate decision. While a low threshold introduces noise and shadows, a high value splits the blob and creates holes. Experimentally we have usually found distorted and fragmented blobs corresponding to a single person. To improve the blob segmentation, we use the anthropometric information that the contour gives us.

Since even poor fittings provide a rough approximation, we first apply a model fitting to obtain a roughly contour. Then we reduce the threshold value inside to detect more foreground pixels. At the same time, we increase its value outside to remove noise and shadows. This process is repeated iteratively to convergence in a more reliable silhouette, as depicted in Figures 6.14, 6.15 and 6.16. In this manner, the threshold selection can be considered a less crucial issue.

Skeletal structure of human body During the training state we have selected manually 49 points corresponding to the contour in four different views, and simultaneously, the 13 points of the skeleton in the four views. We have correlated all points: silhouette and skeleton, so, once we have located the contour of the human figure, using the nearest neighbour strategy, the skeleton is recovery immediately from the corresponding silhouette of the training set (to one silhouette corresponds a skeleton).

Moreover, as we have correlated the skeletal points between different views, we are able to build a 3D skeleton model, just inverting the camera calibration process carried out in the generation of the MoBo database [Gross and Shi, 2001]. Since the calibration method is the Tsai camera model [Klette et al., 1998], the relationship between the coordinates in the image (x_f, y_f) and in the 3D system of the camera (X, Y, Z) can be estimated as:

$$s_x^{-1} \cdot d'_x \cdot (x_f - c_x) \cdot (1 + k_1 \cdot r^2) = f \cdot \frac{X}{Z} \tag{6.7}$$

$$d_y \cdot (y_f - c_y) \cdot (1 + k_1 \cdot r^2) = f \cdot \frac{y}{Z} \tag{6.8}$$

where d_x, d_y, c_x, c_y, s_x, f and $k1$ are the camera intrinsic parameter and

$$r = \sqrt{\left(d'_x \cdot (x_f - c_x) \cdot s_x^{-1}\right)^2 + \left(d_y \cdot (y_f - c_y)\right)^2} \tag{6.9}$$

By introducing the extrinsic parameters, we can obtain the 3D coordinates in the real world

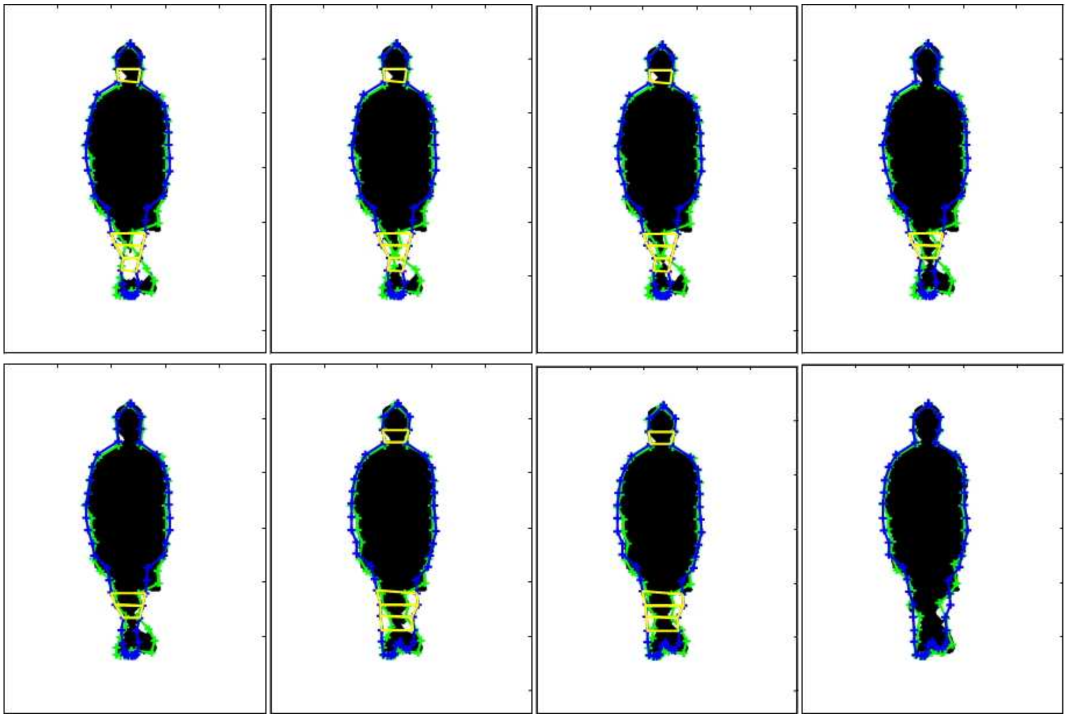


Figure 6.14: Selective thresholding for frontal view. Yellow boxes are those areas where the threshold value is decreased in each iteration. Green contour is the fitted silhouette. Blue contour is the corrected shape using PCA

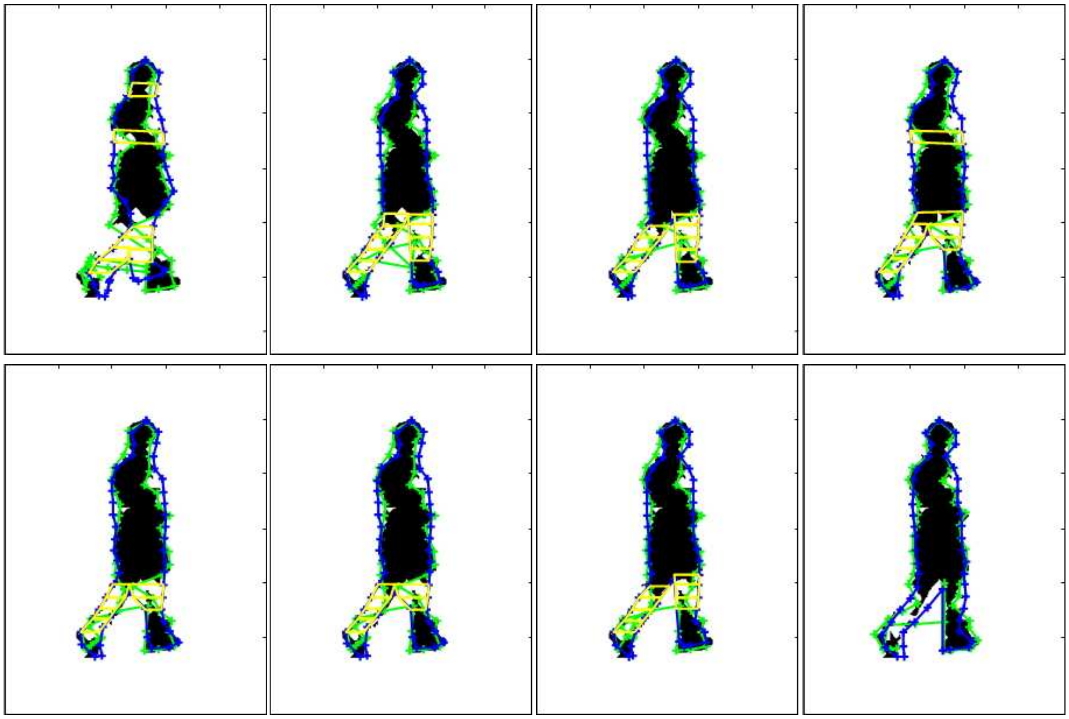


Figure 6.15: Selective thresholding for diagonal view. Yellow boxes are those areas where the threshold value is decreased in each iteration. Green contour is the fitted silhouette. Blue contour is the corrected shape using PCA.

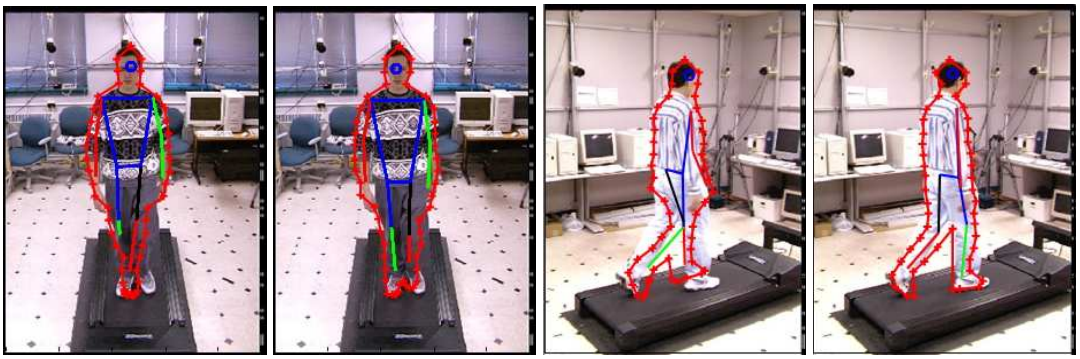


Figure 6.16: Selective thresholding results versus conventional results.

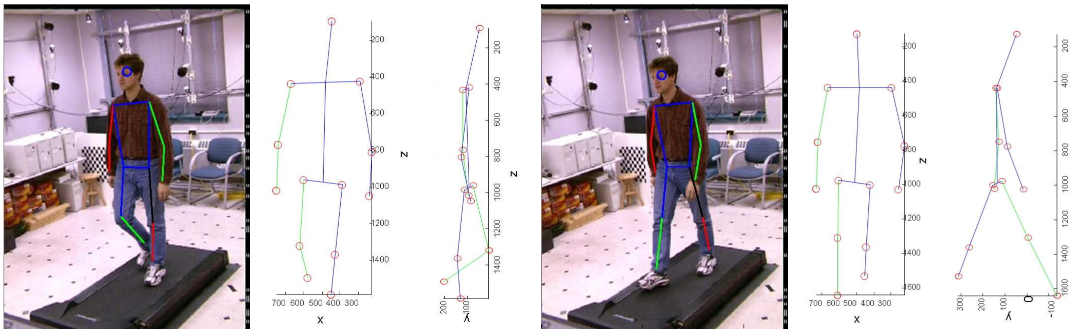


Figure 6.17: 2D skeletons and their corresponding 3D stick figure

(X_w, Y_W, Z_w) as

$$s_x^{-1} \cdot f^{-1} \cdot d'_x \cdot (x_f - c_x) \cdot (1 + k_1 \cdot r^2) = \frac{X}{Z} = \frac{r_1 X_w + r_2 Y_W + r_3 Z_w + T_x}{r_7 X_w + r_8 Y_W + r_9 Z_w + T_z} \tag{6.10}$$

$$f^{-1} \cdot d_y \cdot (y_f - c_y) \cdot (1 + k_1 \cdot r^2) = \frac{y}{Z} = \frac{r_1 4 X_w + r_5 Y_W + r_6 Z_w + T_x}{r_7 X_w + r_8 Y_W + r_9 Z_w + T_z} \tag{6.11}$$

$$\tag{6.12}$$

where R and T are the rotation and movement parameters, so that

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} = \begin{bmatrix} \cos R_z & -\sin R_z & 0 \\ \sin R_z & \cos R_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos R_y & 0 & \sin R_y \\ 0 & 1 & 0 \\ \sin R_y & 0 & \cos R x_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos R_x & -\sin R_x \\ 0 & \sin R_x & \cos R_x \end{bmatrix} \tag{6.13}$$

The result for each point is a system composed of 2 equations and 3 unknown variables. Therefore, this system can not be solved using a single point in a monocular application. Hopefully, the database provides another synchronised view of the same 3D point. In that way, an extra equation is available to solve the system. Therefore, by applying the system of equations to the points that compose the skeleton we can obtain a 3D reconstruction, as shown in Figure 6.17. In Figure 6.18 we represent a walking cycle corresponding to one subject in several views.

Besides the advantage that placing the subject in a 3D environment implies, in terms of action recognition and pose recovery, this method also allows us to solve self-occlusions and determine the location of hidden limbs.

Multi-camera extension Although our goal is obtaining a robust method in a monocular scenario, this methodology can be easily extended to multi-camera application. By linking the PDMs of c cameras, we can compose a unique model so that

$$x = \{x_1^1, \ldots, x_n^1, y_1^1, \ldots, y_n^1, \ldots, x_1^c, \ldots, x_n^c, y_1^c, \ldots, y_n^c\} \tag{6.14}$$

In that way the accuracy is improved substantially regarding the individual PDMs, as it is depicted in Figure 6.19. The robustness of the method is also improved: even when half of the cameras are completey occluded, the information coming from the others are enough to reconstruct the shape correctly. By filling the hidden points in the vector with the mean shape for that view, the correction step of the PCA places those points in the most probably location. In Figure 6.20, we can notice as the result does not degenerate too much even using only one camera to reconstruct the whole model.

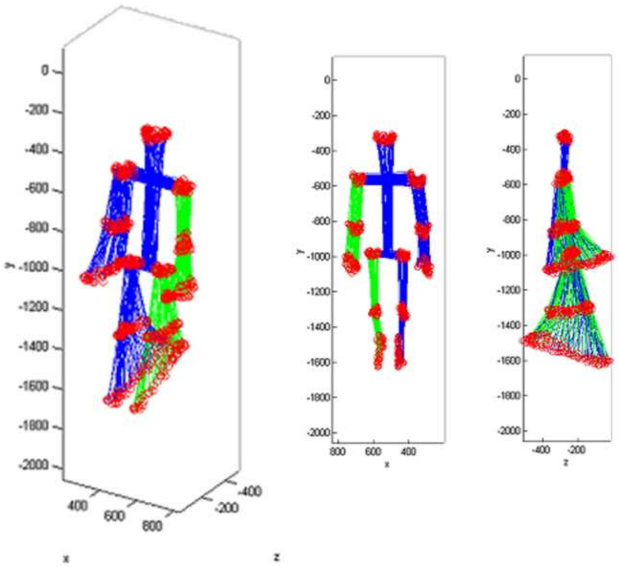


Figure 6.18: 3D skeleton model

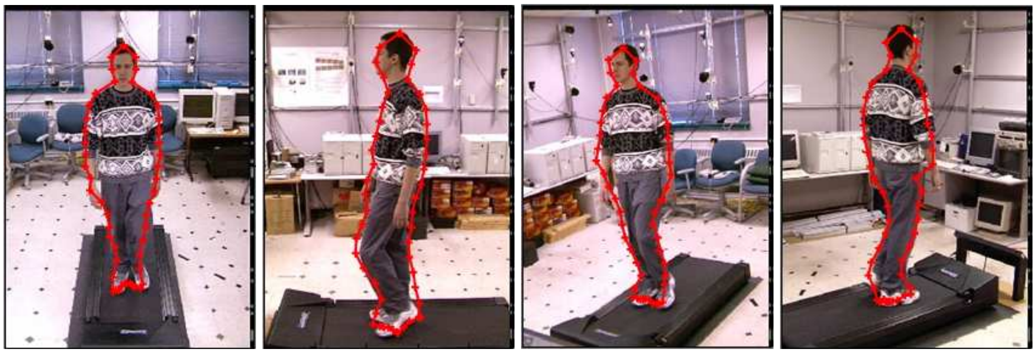
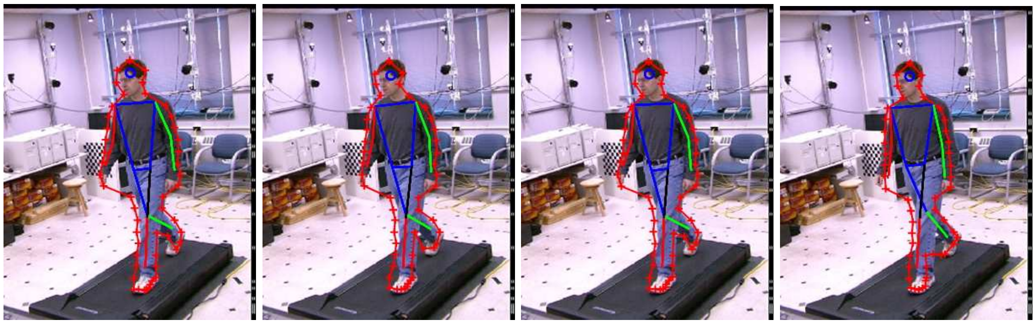


Figure 6.19: Results of applying the multi-camera PDM. The monocular result for the same frame can be seen in Figure 6.13.



a) b) c) d)

Figure 6.20: Results of applying the four camera PDM using information coming form: a) four cameras, b) three cameras, c) two cameras, d) one camera.

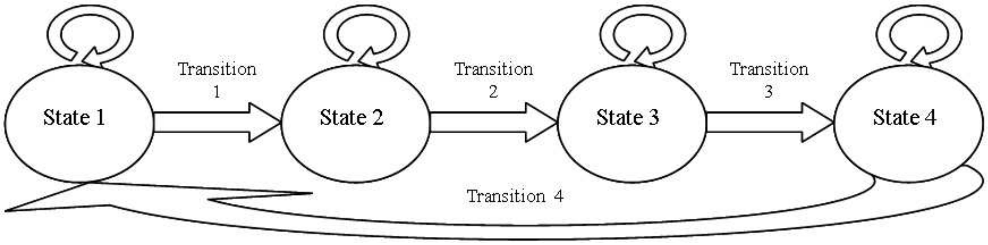


Figure 6.21: State diagram and transitions

6.2.1.4 Human gait analysis

Up to this point we have presented a statistical model of the human figure based on a single image, without taking into account the dynamic of the gait encompassed in a temporal sequence. It is obvious that temporal information will improve the model fitting reducing the number of possible candidates from the database to match. Considering the human gait, we have divided a natural cycle into four states, as it is depicted in 6.21: left foot stop and right foot moving behind, left foot stop and right foot moving ahead; and vice versa, right foot stop and left moving behind or ahead. We have used a k-means clustering algorithm to the skeleton points to divide the whole space for every pose into four sub areas in relation to the previous steps defined in a gait cycle. In the clustering analysis we have considered only the six points of the skeleton corresponding to the hips, knees and ankles. The result is shown in Figure 6.22.

Algorithm 11: K-means Algorithm

```
1. Initialisation: Given $k$ random samples as initial seeds, they are considered the initial
   centroids of the $k$ classes.
2. Iteration:
   • Every sample is classified into a class using a minimum distance criterion to the
     centroids.
   • Centroids are recalculated using the samples assigned to each class.
   • If the distance between the old and the new centroids is lower than a threshold,
     we stop the iteration.
```

Once we have all skeleton classified we apply a LDA analysis to separate the feature space into disjoined regions, see Figure 6.23. The sequentiality can be noticed observing one cycle in the LDA feature space (Figure 6.24).

This state diagram, with four steps and four transactions, produces a more accurate and robust model and at the same time reduces the neighborhood space used to fit a new figure. Once the contour has been fitted and the skeleton extracted, we can project it in the LDA space to be classified. Known its state, we limit the plausible deformation (or the possible nearest neighbours) at the next time step, to those ones which belong to the same state and the possible transitions. If the contour is inside the transitions, the next contour could belong to that transition and the previous and following states.

This procedure avoids errors due to the ambiguity that a pose with opened legs in the lateral view contains, for instance. That pose could be easily labelled as belonging to the states with

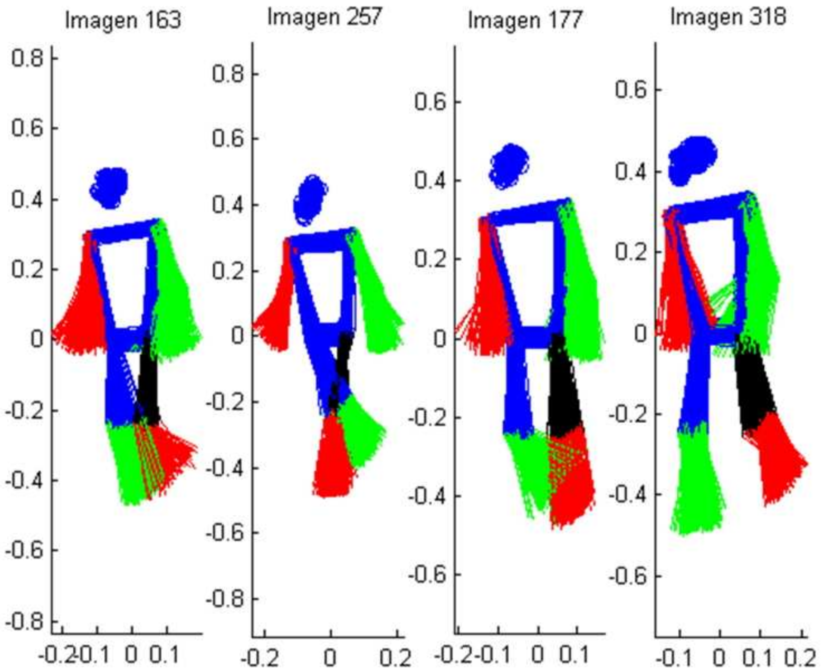


Figure 6.22: Human gait k-mean clustering results for diagonal view

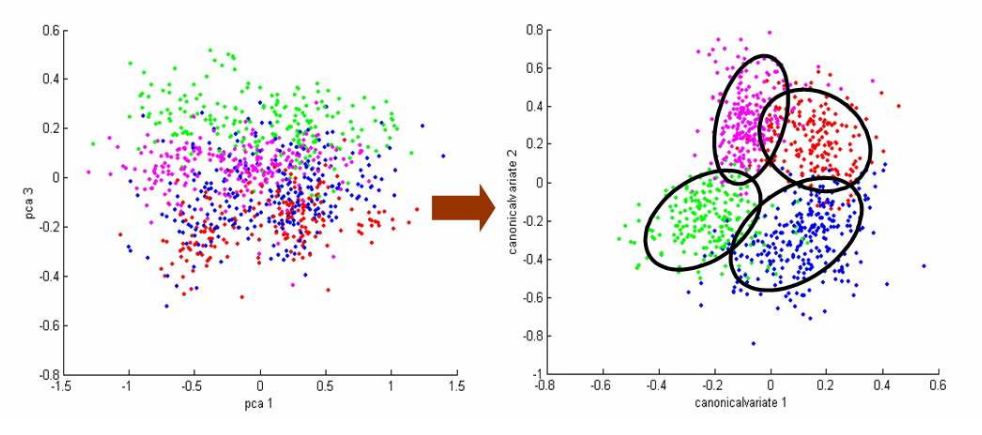
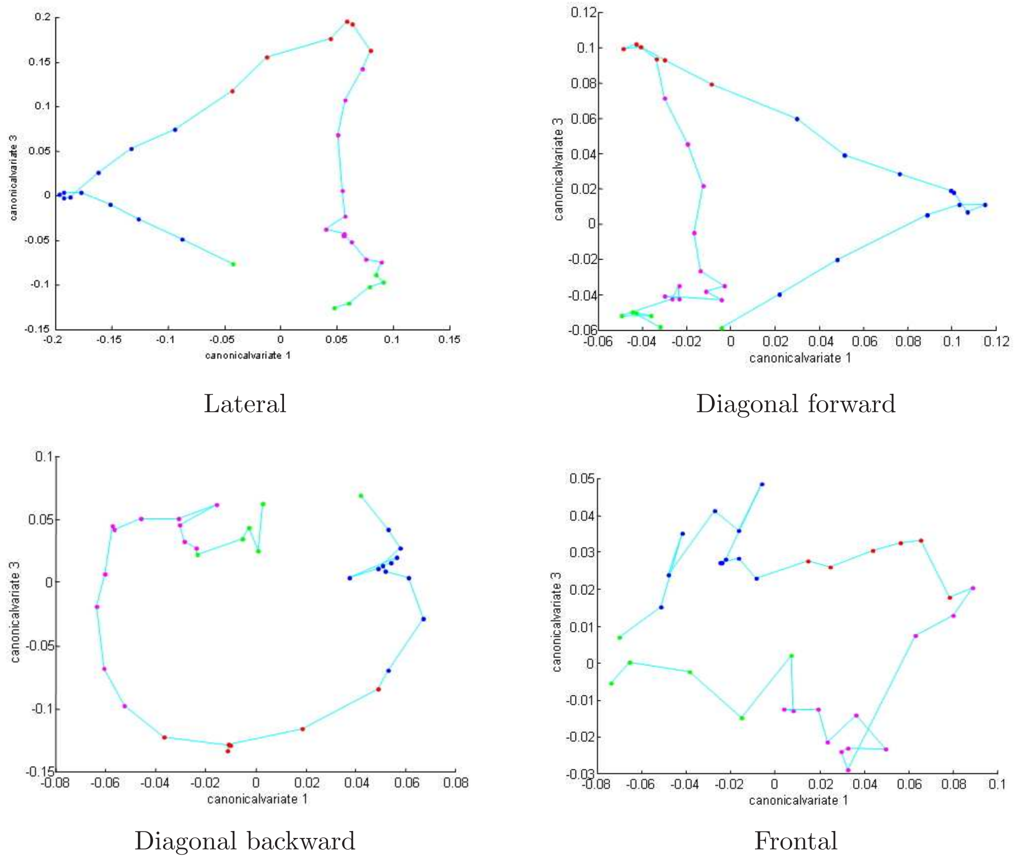


Figure 6.23: Human gait clustered into four steps



Lateral

Diagonal forward

Diagonal backward

Frontal

Figure 6.24: Cyclic sequentiality onto the LDA feature space.

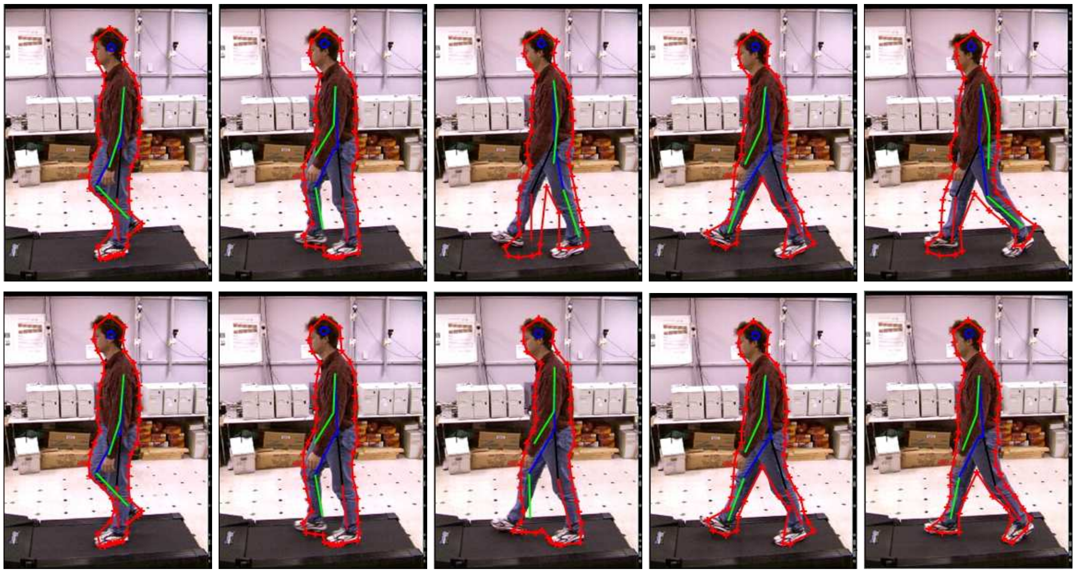


Figure 6.25: Upper row: Results without introduce information about the sequentiality. Bottom row: Results using the human gait improvement.

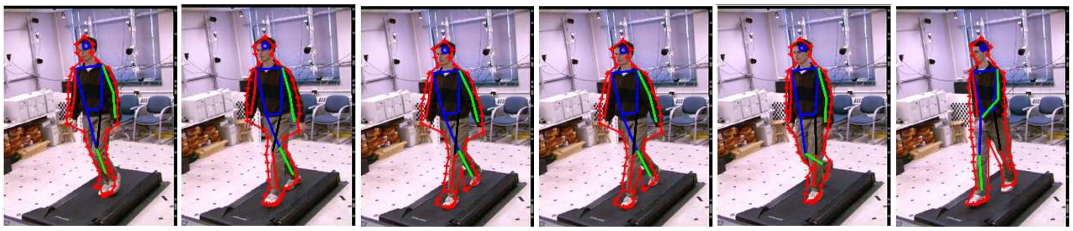


Figure 6.26: Silhouette and skeleton fitting

the right leg forward or with the left leg forward, it does not matter which. Therefore, jumps in the sequentiality of the movement are avoided and better results are obtained, as depicted in Figure 6.25.

6.2.1.5 Results

In our experiments we have verified that the human contour is a good indicator to determinate the skeleton of the person. Obviously, the more precise the silhouette is, the more exact the corresponding stick figure will be as depicted in Figure 6.26.

As we have generated a 3D stick model of the human figure, we can employ this model to recovery the position of some hidden arms and legs in natural movements. In Figure 6.27 we show some results obtained in different camera positions where an arm is completely occluded by the body.

In Figures 6.28 and 6.29 we present the silhouette and skeleton of a person in a completely different sequence in relation to the pictures from the MoBo database, with different camera points of view, frame rates and image sizes. To improve the segmentation process we have used a Kalman filter to track the bounding box corresponding to the human blob. We track the centroid

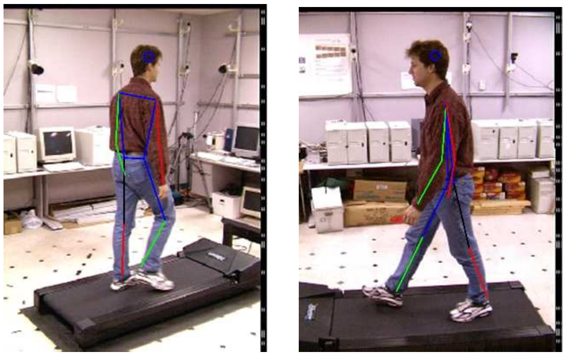


Figure 6.27: Hidden limb detection by 3D skeleton projection

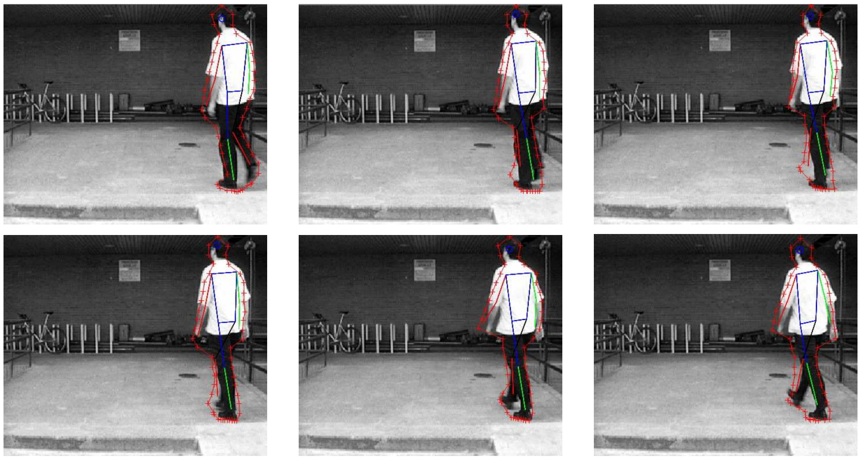


Figure 6.28: Skeleton detection and tracking

of the blob and the width and height. This tracking process reduces significantly problems related to the blob location in very poor contrasted foreground in relation to background, as well as computational cost.

However, the nearest neighbour (NN) strategy has an important drawback. Only learned poses can be obtained, without possibility to generalised new poses, even inside the admissible feature space. This fact implies the use of a large training database to obtain good results and a growing computational time. To solve that, a linearisation can be done. By applying mixture of gaussian the feature space is linearised. This option was presented in paper [Rogez et al., 2006] and it is explained in the next section.

Although more versatile, this option is prone to diverge more easily than the NN strategy. Therefore, extra motion clues should be introduced by means of an integral tracking framework. As we have seen, the AMS methodology requires an external tracking capable of providing the area and the blob where the PDM is fitted. In this section, a simple Kalman filter has been applied and tracking and model fitting have been considered as independent processes. In next section we propose a joint framework which take advantages of the information provided for each system to help the other.

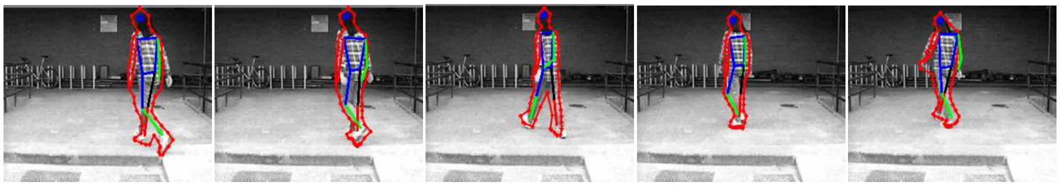


Figure 6.29: Skeleton detection and tracking

6.2.2 Rao-Blackwellised Human Tracking Framework

Full-body articulated modelling and tracking (either deterministic or stochastic) by unconstrained search through the complete space, are not viable because of the excessive computational complexity. Nevertheless, it is possible to tackle this problem by introducing prior assumptions about dynamics, which can be modelled as geometrical equations, or view restrictions.

In contrast with stochastic methods, deterministic body models based on learning techniques relate automatically image observations with body poses, avoiding exhaustive space searches due to the knowledge of dependencies and constraints of the human body. On the other hand, deterministic methods require some assumptions such as image location, viewpoint estimation, dynamic knowledge, to name a few. By contrast, these pre-requisites (or needed prior information) are known in stochastic methods.

The combination between learning techniques with stochastic methods makes feasible the dimensionality reduction limiting the prior probability as well as providing the prior information needed to make effective the learning methods. In this work, Rao-Blackwellised particle filter is proposed as overall framework for integrating both stochastic and analytic techniques.

Rao-Blackwellised particle filter (RBPF) has been applied in other state estimation problems to reduce dimensionality. It has been used in applications so different as SLAM [Doucet et al., 2000], non-linear regression [Doucet et al., 2000], multi-target tracking, or appearance and position estimation [Ba and Odobez, 2005; Khan et al., 2004; Jaeggli et al., 2006]. This technique was already introduced in this thesis in Chapter 4 Section 4.5 as part of a methodology to obtain a robust update algorithm.

6.2.2.1 Rao-Blackwellisation

Particle filters can be inefficient sampling in high-dimensional spaces. However, the state can be separated into tractable subspaces in some cases. If some of these subspaces can be analytically calculated, the size of the space over which PF samples will be drastically reduced. The technique which allows marginalising out some of the variables is called Rao-Blackwellisation [Casella and Robert, 1996]. In particle filtering, Rao-Blackwellisation refers to integrating out part of the state analytically. In this manner, fewer samples (particles) are needed for obtaining the same level of performance. This is due to the fact that part of the posterior over the state can be calculated exactly, instead of using a more expensive and noisy sample set.

6.2.2.2 Rao-Blackwellised particle filter

Let us consider a state space model where y_t are hidden variables which respond to a set of observed variables z_t. We assume z_t as a Markov process of initial distribution $p(y_0)$ and transition equation $p(y_t|y_{t-1})$. Observations are assumed to be conditionally independent given the process y_t of marginal distribution $p(z_t|y_t)$. The aim is to estimate the joint posterior distribution $p(y_{0:t}|z_{1:t})$ or some of its characteristics such as the filtering density $p(y_t|z_{1:t})$. The pdf can be written in the recursive way

$$p(y_{0:t}|z_{1:t}) = \frac{p(z_t|y_t)p(y_t|y_{t-1})p(y_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})}, \tag{6.15}$$

where $p(z_t|z_{1:t-1})$ can be considered as a proportionality constant.

In multi-dimensional spaces, integrals are not always tractable. However, if the hidden variables had a structure, we could divide the state y_t into two groups, r_t and s_t such that $p(y_t|y_{t-1}) = p(s_t|r_{t-1:t}, s_{t-1})p(r_t|r_{t-1})$. In such case, we can marginalise out $s_{0:t}$ from the posterior, reducing the dimensionality problem. Following the chain rule, the posterior is decomposed as follows

$$p(r_{0:t}, s_{0:t}|z_{1:t}) = p(s_{0:t}|z_{1:t}, r_{0:t})p(r_{0:t}|z_{1:t}), \tag{6.16}$$

where the marginal posterior distribution $p(r_{0:t}|z_{1:t})$ satisfies the alternative recursion, and

$$p(r_{0:t}|z_{1:t}) \propto p(z_t|r_t)p(r_t|r_{t-1})p(r_{0:t-1}|z_{1:t-1}). \tag{6.17}$$

6.2.2.3 Human pose RBPF

As mentioned before, human body tracking requires information about the local pose of the subject as well as the global location of the target in the image. In order to support multiple hypotheses for the tracking problem, a particle filter approach can be easily adapted: the state space can be extended with body pose parameters. In this way, the state $y_t = [s_t, r_t] = [x_t, \alpha_t, l_t]$ consists of two parts: the location part $r_t = l_t$ which models the target position, and the body pose variables s_t, composed of the global orientation of the body relative to the camera α_t and its local pose coefficients x_t. We divide the body pose parameters into two sets due to the fact that x_t are obtaining applying a view-dependant model which requires orientation knowledge. Assuming that the camera axis is approximatively parallel to the ground, at shoulder level, the global orientation α_t can be formulated with a single parameter: the person orientation with respect to the camera, which can be simplified as the direction of motion in the image.

In a RBPF, the posterior can be written by the chain rule of probability as

$$p(x_t, \alpha_t, l_t|z_t) = p(x_t, \alpha_t|l_t, z_t) \cdot p(l_t|z_t). \tag{6.18}$$

Given this formulation, we divide the problem into two parts (Fig. 6.30), $p(x_t, \alpha_t|l_t, z_t)$ which can be solved analytically using a learned parametrical model explained in Section 6.2.2.4, and $p(l_t|z_t)$ which does not exhibit an obvious analytic solution, but can be handled by a PF because of its low dimensionality.

In contrast with previous works [Jaeggli et al., 2006; Doucet et al., 2000] where only stochastic weights were used for resampling, we compute the final importance weights including the quality of the analytic estimation. Thus, we guarantee a correct resampling not only in the dimensions of the stochastic parameters, but also in remaining dimensions of the space.

6.2.2.4 Human modelling

As analytic model, we apply a probabilistic 2D model for pedestrian motion analysis in monocular sequences [Bowden and Sarhadi, 2000; Rogez et al., 2006]. This model is a probabilistic evolution of the model presented in the previous section. Now, we discretise the viewpoint and construct a global framework, composed of 8 dynamical view-based models

Algorithm 12: Rao-Blackwellised Particle Filter for human tracking

For each time step t, starting from the distribution $\{l_{t-1}^i, \Pi_{t-1}^i, \; w_{t-1}^i(x_{t-1}, \alpha_{t-1})\}_{i=1}^N$, the posterior probability $p(l_t, x_t, \alpha_t|z_t)$ is recursively approximated as a set of N weighted hybrid particle set $\{l_t^i, \Pi_t^i, \; w_t^i(x_t, \alpha_t)\}_{i=1}^N$

1. Sequential importance sampling state

   - For $i = 1, ...N$, sample

   $$\hat{l}_t^i \sim q(l_t|l_{t-1}^i, z^t)$$

   - For $i = 1, ...N$, calculate the importance weights

   $$\pi_t^i \propto \pi_{t-1}^i \frac{p(z_t|l_t^i)p(l_t^i|l_{t-1}^i)}{q(l_t^i|l_{t-1}^i, z_t)}$$

2. Analytic step

   - For $i = 1, ...N$, calculate the posterior density $w_t^i(x_t, \alpha_t) = p(x_t, \alpha_t|z_{0:t}, \hat{l}_{0:t}^i)$ on the subspace coefficients x_t,α_t using the analytical solution. We denote as $z_{0:t}^i$ the image descriptor computed at sampled location $\hat{l}_t^i$

   $$w_t^i(x_t, \alpha_t) = p(x_t, \alpha_t|z_{0:t}^i) \propto p(z_t^i|x_t, \alpha_t)p(x_t, \alpha_t|z_{0:t-1}^i) = L_t^i \cdot p(x_t, \alpha_t|z_{0:t-1}^i)$$

   - For $i = 1, ...N$, reweigh the samples

   $$\Pi_t^i = p(z_t|\hat{l}_t^i, x_t, \alpha_t) = \pi_t^i \cdot L_t^i$$

3. Resampling step

   - Multiply/suppress samples $\hat{l}_t^i$ with high/low normalised importance weights $\tilde{\Pi}_t^i = \frac{\Pi_t^i}{\sum_{i=1}^N \Pi_t^i}$, respectively, to obtain N samples l_t^i.

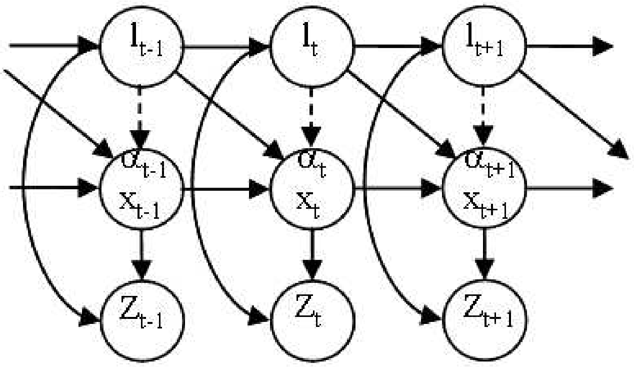


Figure 6.30: Graphical structure of the Rao-Blackwellised particle filter obtained by partitioning the search space into 2 parts.

that can respond robustly to any change of direction during the sequence using the parameter α_t.

Each human model encapsulates within a Point Distribution Model (PDM) $x_t = [co_t^1, ..., co_t^{200}, sk_t^1, ...sk_t^{26}]$ the information of the full body silhouette (given by the 2D Shape made of a series of landmarks $co_t^{n_1}$ located along the human contour) and the structural information $sk_t^{n_2}$ (given by the corresponding 2D stick figure, also called Skeleton).

The method is based on learning dynamical models. Correspondences between several different views of the same walking sequences have been established previously during the capture stage. A clustering in a global Shape-Skeleton feature space where all the views considered are projected together is performed. The different clusters correspond in terms of dynamics or view-point. For each cluster, a series of local motion models is learnt by clustering the stick figure subspace. Gaussian Mixture Models (GMM) are used to cope with the problem of non-linearity of the model as proposed in previous works [Cootes and Taylor, 1997]. GMM are fitted to the total Shape-Skeleton training data using the Expectation Maximization (EM) algorithm.

In this way, the pose pdf for a certain view can be expressed as

$$p(x_t, \alpha_t) = \frac{1}{C+R} \sum_{c=1}^{C} p(c) \sum_{r=1}^{R} p(r) N(\mu_{r,c}, \sigma_{r,c}), \tag{6.19}$$

being $\mu_{r,c}$ and $\sigma_{r,c}$ the parameters of the Gaussian components, and $p(c) \cdot p(r)$ the weights estimated by the EM algorithm during the learning stage. R and C are the number of states in which the viewpoint and the motion cycle have been divided respectively.

Temporal and spatial constraints are considered to build a Probabilistic Transition Matrix (PTM) depicted in Figure 6.31. Temporal constraints enable a frame to frame prediction $p(C_t^i|C_{t-1}^j)$ (corresponding to the probability of being in cluster i at time t conditional on being in cluster j at time $t-1$) of the most probable local models from the GMM that have to be considered. This probability is modelled using a transition matrix learned using the training dataset (see Table 6.1).

Spatial constraints select the adequate model to be applied $p(R_t^k|R_{1:t-1}^{m_{1:t-1}})$. Since the subject can change the direction arbitrarily, but this transition will be considerably slow, this probability is modelled as a Gaussian centered in the prediction calculated from the last direction [Rogez et al., 2006]. However, while in that paper the probabilities are static, here the error between the estimation an the prediction of the tracking controls the variance of the

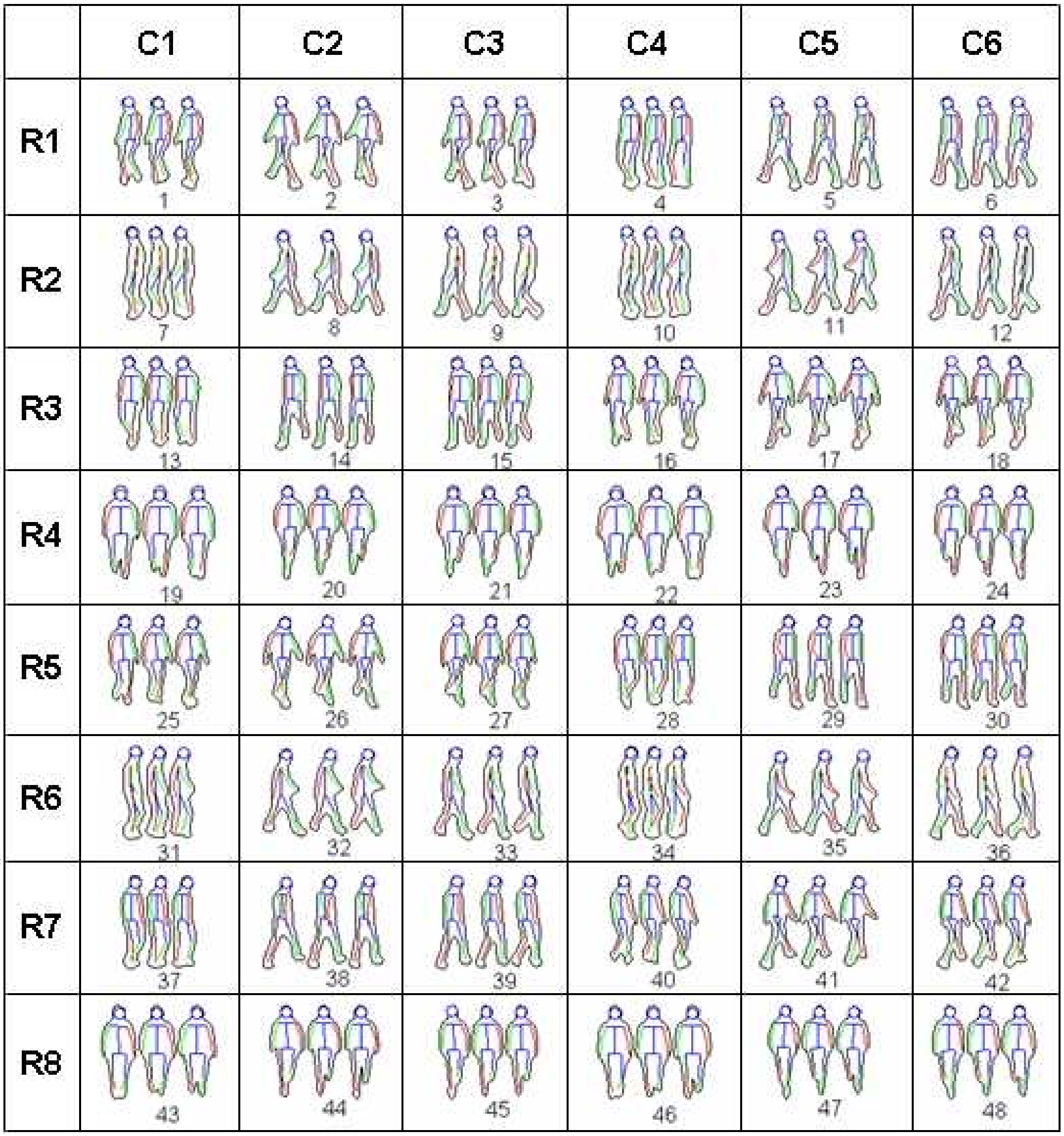


Figure 6.31: Human pose model. Columns show the states of the gait cycle and rows the 8 discretised viewpoints

Table 6.1: Temporal constraints: Cluster transition matrix.

t t-1	1	2	3	4	5	6
1	0.77	0.04	0.19	0	0	0
2	0	0.88	0.12	0	0	0
3	0	0.07	0.8	0.13	0	0
4	0.01	0	0	0.77	0.07	0.16
5	0	0	0	0	0.87	0.13
6	0.14	0	0	0	0.08	0.78

transitions probabilities. In that way, not only the location and direction estimation, but also the uncertainty of the tracking helps to obtain a better PDM fitting.

Once the model has been generated (off-line), it can be applied (on-line) to real sequences. Given an input human blob provided by the Rao-Blackwellised particle filter, Shape and Skeleton estimated in previous frames, and the point of view, the model is fitted for inferring both body shape and posture. This model fitting is applied following the classical PDM methodology [Bowden and Sarhadi, 2000], but introducing the constraints: the prediction of the most probable models from the GMM can be estimated by means of the PTM. It allows a substantial reduction in computational cost since only few models have to be considered.

$$p(x_t, \alpha_t | x_{t-1}, \alpha_{t-1}) = p(C_t^i | C_{t-1}^j) p(R_t^k | R_{1:t-1}^{m_{1:t-1}}) p(x_t, \alpha_t).t \tag{6.20}$$

6.2.2.5 Image tracking

In our case, the state $l_t = [x, y, v^x, v^y, s^x, s^y]$ characterises the image object configuration. (x, y) specifies the translation position of the object in the image plane, (v^x, v^y) the velocity of the target and (s^x, s^y) the width and height scales. In order to estimate the velocities of the target, and predict the new location in next frames, we use a first order dynamic model (constant velocity).

An adequate likelihood function must be applied to track the targets. To weigh each particle, we have combine multiple visual clues assuming independence between them: colour and movement.

$$p(z_t | l_t) = p_{col}(z_t | l_t) \cdot p_{mov}(z_t | l_t). \tag{6.21}$$

Colour is a discriminative cue which differentiates between object and background, but also between objects:

$$p_{col}(z_t | l_t) \propto \sum_{l_t} \exp\left(- \frac{(z_{l_t}^{col} - \mu_{c_F}) \Sigma_{c_F}^{-1} (z_{l_t}^{col} - \mu_{c_F})^\top}{\min_{c_{Bk}} \left((z_{l_t}^{col} - \mu_{c_{Bk}}) \Sigma_{c_{Bk}}^{-1} (z_{l_t}^{col} - \mu_{c_{Bk}})^\top \right)} \right), \tag{6.22}$$

where μ_{c_F},Σ_{c_F} are the components of a colour mixture model of the target and $\mu_{c_{Bk}}$,$\Sigma_{c_{Bk}}$ are the components of a colour mixture model of the background.

Movement can not distinguish between objects, but it improves the tracking eliminating background areas with the same colours than the objects.

$$p_{mov}(z_t | l_t) \propto \sum_{l_t} \exp((z_{l_t}^{col} - M_{l_t}^{Bk})^\alpha) - 1, \tag{6.23}$$

where $M_{l_t}^{Bk}$ is the background model and α is an ad-hoc parameter ($\alpha = 4$).

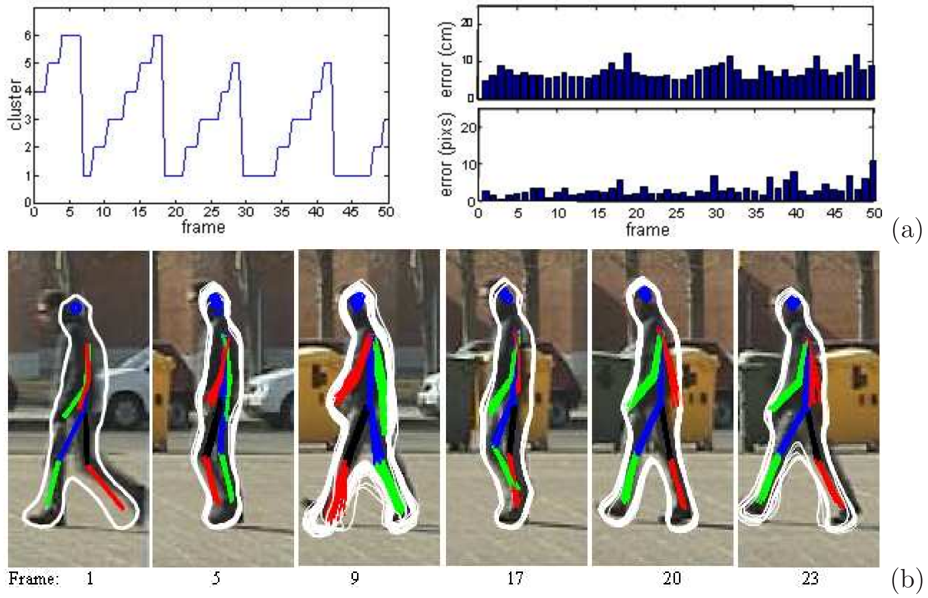


Figure 6.32: (a) Pose error in centimeters and tracking error in pixels (right), and temporal clusters (left). (b) Particles obtained for the Straight line walking sequence.

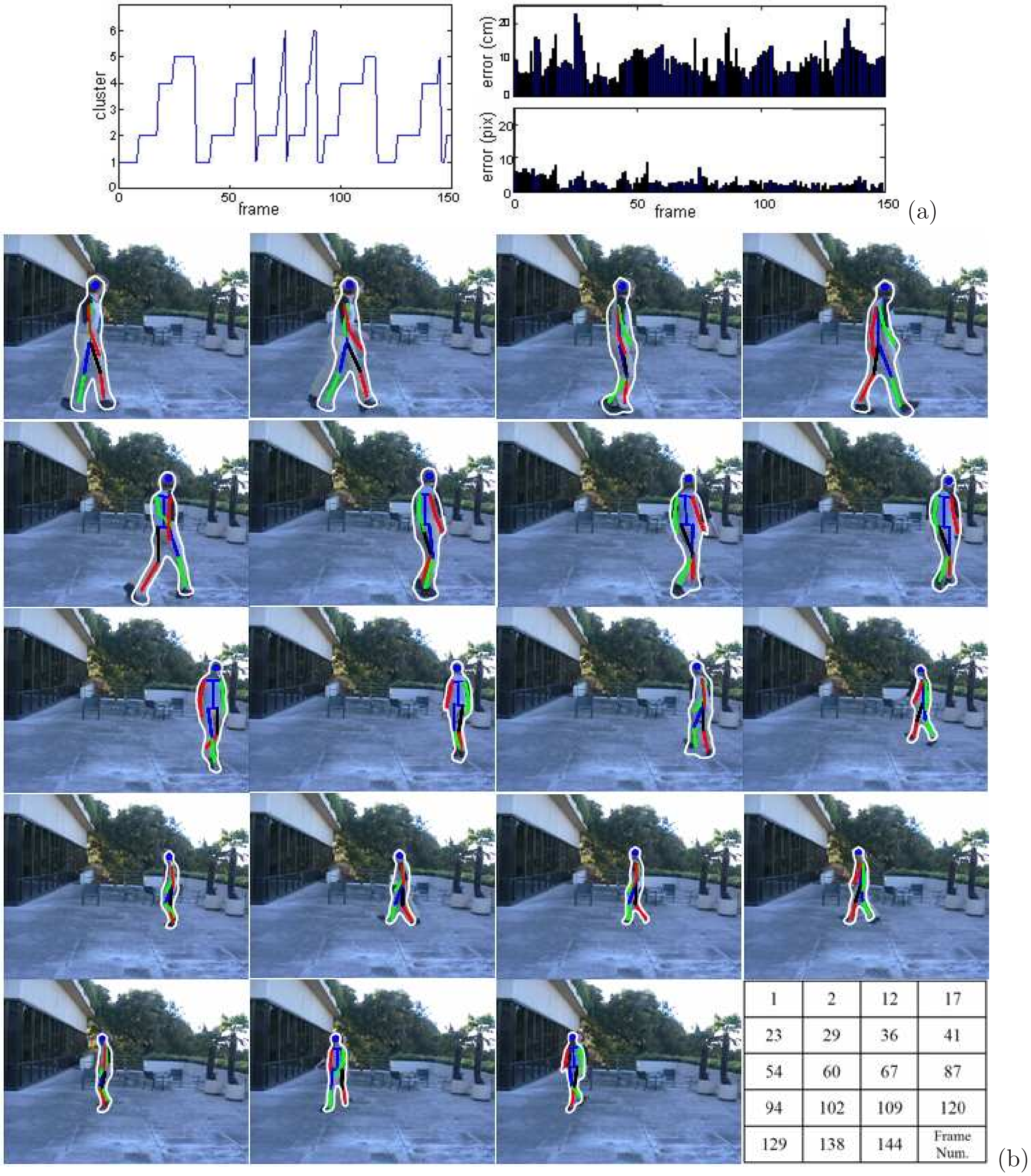
6.2.2.6 Experimental results

The algorithm has been evaluated with different testing sequences that illustrate common situations which may occur in the analysis of pedestrian motion: straight line walking, direction and scale changes, etc. The process starts with a manual initialisation: indicating the location, the angle and the adequate model in the first frame. The automatisation of this stage is out of the scope of this project, pending for future work.

To evaluate the results, we have extracted three different kinds of statistics: the temporal motion continuity of the model (the column of the PTM corresponding to the estimated model), the mean square error (in cm) between skeleton joints obtained in each frame and a ground truth skeleton extracted manually (pose error) and the tracking error of the bounding box (in pixels). The pose error has been normalised with respect to the real target height, thus obtaining an error independent of the scale in the image.

As illustrated in Fig.6.32, results for a pedestrian crossing the scene without any change in the viewpoint are excellent. In the walk-circle sequence (Fig.6.33), results are globally very satisfactory. However, the model fails when the system must apply the back model because of the high sensitivity of this particular model to shadows and fragmented blobs (the difference between states in this view consists in the few pixels which compose the feet) and the viewpoint differences with the training set. However, the system is able to recuperate the dynamic behaviour of the input motion when the image measurement is improved.

Nevertheless, as it is shown in the state of the art, the discriminative approaches that we have presented are only valid for those actions which have been trained. This fact limits their application to real scenarios, where we do not have control at all about the activities. Even though a large database with multiple activities can mitigate the problem, most discriminative approaches are not generalisable and their performance decreases with the number of learned activities. Moreover, a larger amount of activities implies a high computational cost, which can be not admissible in real time application and surveillance. To cope with this problem, in next



1	2	12	17
23	29	36	41
54	60	67	87
94	102	109	120
129	138	144	Frame Num.

Figure 6.33: (a) Pose error in centimeters and tracking error in pixels (right) and temporal clusters (left) (b) Contour and skeleton obtained for the Walk-circle sequence from www.nada.kth.se/hedvig/data.html.

section we explore generative approaches, which do not depend on learned motion.

6.3 Articulated Tracking

As we have mentioned in the introduction, conventional tracking methods based on 2D models cannot deal by themselves with the intrinsic ambiguity of projected 3D postures, self occlusions and distortions introduced by camera perspective. For that reason, this kind of methods usually restricts the type of sequences they can handle and the scenarios where they can operate. Instead, we propose to use some specific knowledge about biomechanics and human gait analysis to deal with these drawbacks.

In this section, we propose a novel framework based on a set of particle filters to track human body parts. It relies on a generative approach based on a 2D model constrained only by human biomechanics. The inclusion of biomechanical knowledge about bipedal motion significantly reduces the complexity of the problem. This reduction of complexity is achieved by the detection of the pivot foot - i.e. the foot which is static during a step - and its trajectory during a whole step. Then using our 2D model, tracking of human body parts is achieved by a set of particle filters [Isard and Blake, 1998a; Deutscher and Reid, 2005], which iteratively refine their solution.

In this work, we concentrate our efforts on tracking a subject legs since the other body parts do not benefit from biomechanics constraints. Our results are evaluated against the HumanEva data set which is becoming the standard for assessing human body tracking algorithms [Sigal and Black, 2006]. Not only does it provides sequences showing various activities seen from different view points, but it also includes ground truth captured using a motion capture system. Therefore, it offers the opportunity to judge the quality of our results and their objective comparison with those of other researchers who processed the same data.

6.3.1 Limb Tracking: General Principle

Human motion is highly multi-modal and this non-gaussianity is amplified in the image plane by the camera perspective. Therefore, a tracking framework capable of working with non-linear distribution is required. Since particle filter has been successfully applied for this purpose [Isard and Blake, 1998a], we apply this algorithm within our tracking framework. The presented approach demonstrates the feasibility of recovering human poses with data from a single uncalibrated camera using a limb tracking system based on a 2D articulated model.

Our scheme is based on a set of particle filters to fit a 2D articulated model on each frame of a video sequence. In addition, we take advantage of a biomechanics constraint inherent in human bipedal motion: during a ‘step’, one leg pivots around a single point. This allows us dealing with much more activities than other techniques which rely on training on a specific activity. Since we are able to detect the position of this point, this constraint is integrated in an asymmetrical 2D model where the two legs are treated differently. Finally, model fitting is performed after different trackers have been applied successively.

Initially, a ‘standard’ particle filter process operates to track lower limb locations until the end of the ‘step’. Due to the high dimensionality of the problem and the ill-conditioned model, it may not be able to produce satisfactorily tracking. In order to refine the tracking of the articulated model, two assistant particle filters are then launched in parallel using information intrinsic to the ‘step’ of interest (see Figure 6.34). The main reason to use two trackers instead of just one is to handle weaknesses inherent to tracking, i.e. the risk of being trapped in a local minimum and the degradation and potential divergence of tracking over time.

To take advantage of the 'pivot' point constraint and trajectory information, we propose to rely on data captured during a full 'step' before completing the tracking task. While a short delay is introduced - typically around 10 frames (i.e. 0.5s in our test dataset) - in a real time system, this allows processing a wide range of human activities involving displacement without loss of accuracy. Although some actions, such as running or jumping, break the 'pivot' constraint during short periods of time and the 'pivot' point can be momentarily occluded, this can be detected and handled without affecting significantly the proposed tracking framework. Thus, if no 'pivot' point is detected, an unconstrained tracking can be applied. In the same way, during dual stance when both feet are in contact with the ground, a dually constrained tracking process can be launched.

Whereas the 'support' leg location could be accurately predicted thanks to the strong constraint imposed by the 'pivot' point, the tracking of the 'swinging' leg is more difficult, and usually only provides a reasonable approximation of the leg position. Therefore, the state vector of the particle filter is partitioned into two consecutive stages. First, the position of the 'support' leg is calculated. Secondly the pose of the 'swinging' leg is processed using the position of the hip of the other leg as an extra constraint.

First, the last position of the 'swinging' leg, given by the 'standard' tracker, is updated to fit the position of the new 'pivot' point. Then, one of the auxiliary processes, called 'forward' tracking, recalculate the pose of the 'swinging' leg adding constraints extracted from the trajectory information. The other tracker, called 'backward' tracker, is launched in parallel. The new position of the pivot point is used as the initial position of the 'backward' tracking process where the temporal axis is inverted. Since the 'forward' tracker fits model from the first frame of the 'step', while the 'backward' tracker starts from the last frames, their predictions can be compared using a stopping criterion to find the best convergence point between both processes in order to guarantee a soft transition (see Figure 6.34).

6.3.2 Biomechanics Constraints for Human Motion Tracking

Most human motion tracking methods rely on constraints such as specific activity, constant velocity, linear or periodic motion which critically impact on their accuracy and/or their generality. Study of human biomechanics, however, reveals that human motion itself provides some explicit constraints. In this section, we show they can be utilised to simplify the task of tracking human body parts. Walking is a very common human activity whose many other motions, such as loitering, balancing and dancing, can be seen as derivatives and where the underlying mechanics of walking can be applied. All these bipedal motions are based on a series of 'steps' defined as one leg 'swinging' around a 'support' leg whose foot, or 'pivot', stays in contact with the ground at any instant [Fryer, 1971].

Therefore, the detection of this pivot point from a video sequence permits a significant reduction of the complexity of the tracking task without important restriction regarding the types of motions which can be processed by the tracker.

Knowledge of the precise position of the pivot foot also allows using different strategies for tracking either the 'support' or the 'swinging' leg, which enhances the power of our 2D model. Moreover, positions of consecutive striking feet provide some information about the subject's trajectory in the image plane which supplies clues regarding the relative camera-subject position.

In addition to the 'pivot' foot constraint, the 'support' leg has another property: upper and lower legs are supposed to be aligned during the pivot motion around the static foot. Therefore, estimate of the locations of the associated knee and hip is refined if they do not form a straight line with the pivot foot. This assumption is strictly true because this leg should support the weight of the whole body and the impact against the floor. When the knee starts to blend

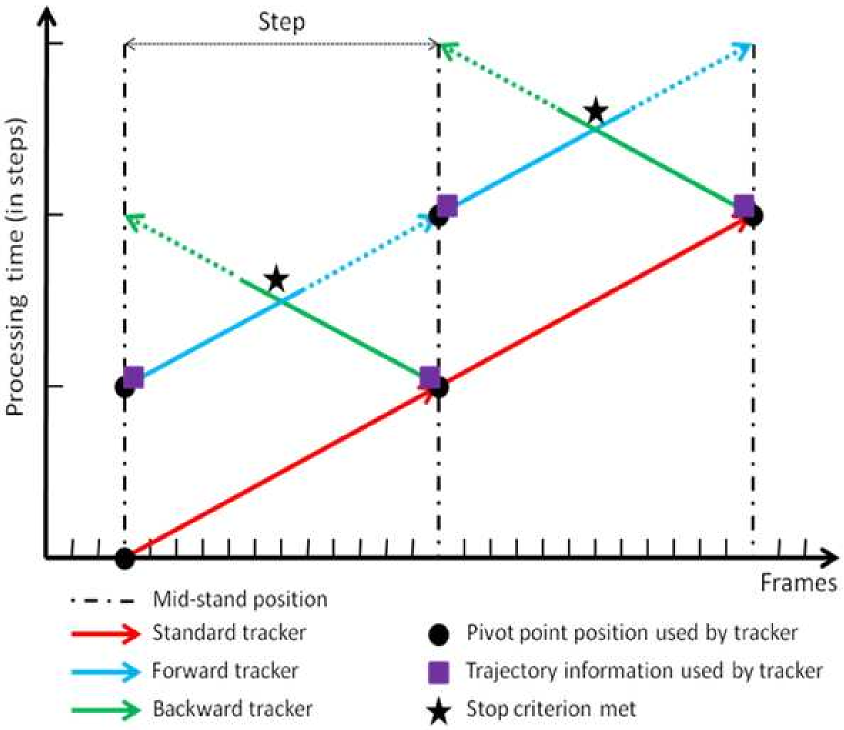

Step
Processing time (in steps)
Frames
Mid-stand position
Standard tracker
Forward tracker
Backward tracker
Pivot point position used by tracker
Trajectory information used by tracker
Stop criterion met

Figure 6.34: Multi tracker framework.

in order to take impulse for the next step, the 'pivot' point moves to the other leg. Only in movements such as jumping or running there is no 'pivot' point for a few milliseconds, but even in those situation, when this pivot point exists the assumption is valid.

6.3.2.1 Detection of pivot foot

In our framework, the static foot is detected using the algorithm proposed in [Bouchrika and Nixon, 2006]. It is based on the biomechanics of gait motion. During the strike phase, the foot of the striking leg stays at the same position for half a gait cycle, whilst the rest of the human body moves. The pivot foot is detected using a low-level feature: corners produced by the Harris corner detector (see Appendix B). Corners associated to the pedestrian of interest are accumulated across several frames (i.e. 20 in our implementation). The region where the leg strikes the ground must have a high density of corners. Although this approach is usually efficient (when an individual walks parallelly to the plane of the camera, the static foot is detected easily), motions towards or away from the camera produce many points seen as static on the body due to the influence of the perspective. We deal with this by apply a double filtering process, i.e. temporal and spatial, to remove outliers and false positive by maintaining both temporal and spatial coherence of the 'pivot' point.

For every pedestrian i, the associated corners are accumulated across several frames using equation (6.24):

$$C_i = \sum_{t=1}^{N} (H(I_t) \wedge L_{i,t}) \tag{6.24}$$

where H is the output of the Harris corner detector, I_t is the original image at frame t, $L_{i,t}$ is i_{th} object at frame t and $\wedge$ is the logical conjunction operator. Although we only consider one pedestrian, as commented in the introduction of this chapter, the pivot point detection algorithm could be extended to deal with multiple people by selecting an appropriate association algorithm.

The region where the leg strikes the ground must have a high density of corners since the static foot is fixed for half a gait cycle. Dense areas of corners are located using a measure for density of proximity. The value of proximity at point p depends on the number of corners within the region R_p and their corresponding distances from p. R_p is assumed to be a square area with centre p, and radius of r that is determined as the ratio of total image points to the total of corners in C_i.

First proximity value dp of corners are computed for all regions R_p in C_i using equation (6.25). This is an iterative process starting from a radius r. The process then iterates to accumulate proximity values of corners for point p.

$$\begin{cases} d_p^r = \frac{N_r}{r} \\ d_p^i = d_p^{i+1} \frac{N_i}{i} \end{cases} \tag{6.25}$$

where d_p^i is the proximity value for rings of radius i away from the centre p, and N_i is the number of corners which are of distance i from the centre, rings are single pixel wide. Afterwards, all the densities for the subregions R_p for all points p are accumulated into a matrix to produce the corner proximity matrix of the frame. Highest values in the matrix generally correspond to the heel strike areas.

6.3.3 Multiple Particle Filter Tracking Based on 2D Articulated Model

6.3.3.1 2D asymmetrical articulated model informed by trajectory information

Our model aims to track simultaneously the global position of the limbs in the image, which defines the coordinates of the subject in the image, as well as the relative position of the different parts of the limbs, which defines the pose of the person. That is, a simultaneous estimation of the limbs in the image and in the pose space is carried out. For that, the tracker state vector is composed of the image coordinates of the hip points and the parameters which model the relative motions and positions, such as angles and lengths in the image plane. In order to introduce the biomechanics constraints, which rely on a relative independence between both legs, both hip points are employed as references and the angles of both legs with respect to the hips are included in the state vector. The state vector of each leg is described by the following equation:

$$X_{leg} = [x_{hip}, y_{hip}, \dot{x}_{hip}, \dot{y}_{hip}, \theta_{thigh}, \theta_{leg}, \dot{\theta}_{thigh}, \dot{\theta}_{leg}, l_{thigh}, l_{leg}, \dot{l}_{thigh}, \dot{l}_{leg}] \tag{6.26}$$

where x and y are the coordinates in pixels, θ is the angle between a limb and the x axis and l is the length of the limb (see figure 6.35).

Once the 'support' leg is estimated, the hip point of the 'swinging' leg is constrained by the distance between the two hips, which is set at a fixed anthropometric value D_0. Moreover, the two hips points are supposed to share the same y coordinate. Both assumptions don't limit neither the ability to generalise to different people nor the type of activity. The hip distance is proportional to the width of the legs, which is set in the initialisation, and therefore different for every tracked person. Regarding the y-coordinate, its almost true if we consider the y-axis as that one which goes along the dorsal spine of the subject. This direction can be determined by calculating the momenta of the human figure, being the y-axis the larger axis of the ellipse that surround the subject. Thus, only the perspective of the camera makes that this assertion is not always truth whatever the activity, but the deviation is negligible if we assume that the camera is far away from the subject and the projection is orthographic and we don't use a zenital view.

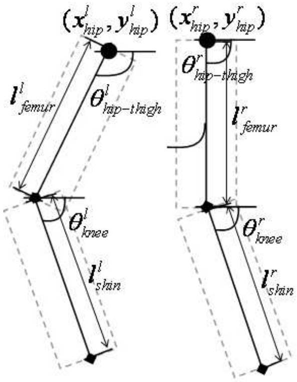


Figure 6.35: 2D articulated model.

Due to its nature, 2D tracking allows a higher flexibility and simplicity of use and initialisation than 3D tracking. However, in 2D it is not possible to introduce traditional constraints, such as motion dynamic or kinematics. Instead we include 3D properties in the 2D world. In the 3D world, the distance between the hips remains constant over time. However, when this fixed distance is projected in the camera plane, its value is changed by two different parameters: the location and the orientation. Whereas the location introduces a factor of scale which is estimated with the global size of the legs, the orientation distorts this distance in a non-linear way which depends on the view point. Techniques such as Hidden Markov Models which model a particular motion dynamic or kinematics constraints cannot be applied because it is not possible to include the camera perspective effect in a non-calibrated environment without loosing the generality.

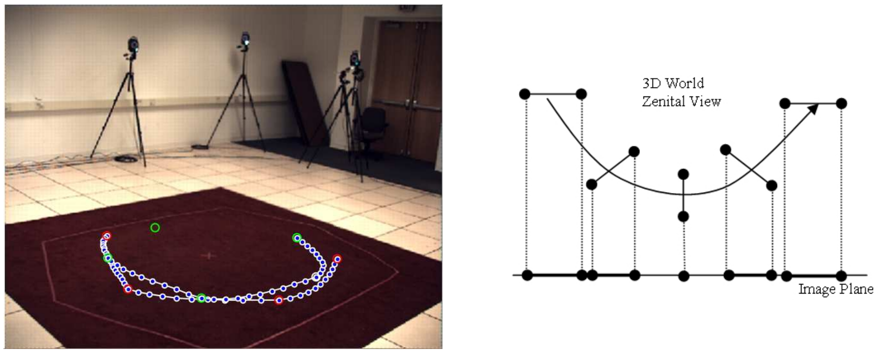
Because of the stochastic nature of our tracking algorithm, the exact value of this distance is not required. Given the poses of the hips at the beginning and the end of a step, values of the hips between these two frames are estimated. In fact, the distance is correlated to the angle of the step trajectory in non-linear manner as shown on Figure 6.36a. We approximate this correlation function using a function which models a S-curve. This function offers a more faithful representation of reality than the sinusoidal function, which in principle also provides a good approximation but gives a worse fitting to the real data of the database than the S-curve.

$$D(\theta) = D_0 \cdot \frac{1 - e^{-\alpha\theta}}{1 + e^{-\alpha\theta}} \tag{6.27}$$

where D_0 is the maximum size of the hip distance with respect to the size of the leg (in our implementation, it is half the value of the sum of the thigh widths), θ is the angle between the trajectory and the x axis in the image plane and α is an empirical factor which controls the speed of the curve descent.

Therefore, we infer hip distances by estimating at each frame the trajectory angle. This is performed by fitting cubic splines to all pivot points (see figure 6.36a,b).

Knowing the new pose of the 'swinging' leg at the end of the step, and therefore the three point which composed it (hip, knee and ankle), the final size of each limb is also known. As



(a) (b)

Figure 6.36: (a) Interpolated trajectory (blue dots) of the pivot points (red and green dots). (b) Correspondences during a turn between the hip distances in a zenithal view and in the image plane.

the size of the leg is included between the size at the beginning and at the end of the step, we can use both values to delimit the uncertainty of the size parameter during the step.

It allow us to introduce the two gait constraints which help both loop and back tracking process to improve the results of the 'swing leg': the size and the hip distance constraints. Since this information is known a posteriori, it can only be applied to the auxiliary tracking process.

6.3.3.2 Multiple particle filter tracking

One of the most challenging problems of 2D tracking is to deal with the perspective effect which amplifies changes in trajectories and, therefore, can create major variations in the target's size. Therefore, the usage of a simple first order model does not allow representing size dynamics adequately. Since our tracking framework is based on a full 'step' where heel strike positions are known, the final position of a step is partially reinitialised. Consequently, information is available to define the trajectory of the target during each step. Moreover, new tracking constraints are derived regarding maximum and minimum apparent limb sizes and distances between the hip points during the step. This last constraint provides a reference point for the 'swing' leg similarly as the pivot point restricts the location of the 'support' leg. These new constraints, which were not initially available when the standard tracker operated, would reduce significantly the complexity of the tracking problem. Furthermore, when using a particle filter based tracker, the probability of divergence increases after each prediction: the closer a frame is to the initialisation frame, the more accurate the estimation is likely to be.

In order to take advantages of these new constraints and tackle this inherent tracker weakness, we propose that once the standard tracker has processed a full 'step', two new trackers should be launched in parallel (see Figure 6.37). These trackers have the same configuration and dynamical models enhanced by the constraints extracted from the output of the standard tracker. Whereas the 'forward' tracker starts from the first frame of the step, the 'backward' tracker begins at the last frame and tracks targets backwards.

Consequently, for each frame two estimates are available. However, since estimate reliability depends on its distance from the initialisation frame, a criterion is designed to decide at which frame the backward tracker is more likely to provide more accurate estimates than the forward tracker. Although the particle filter does not provide an actual estimation for each frame, a weighted mean estimation is extracted combining all the hypotheses. By using this temporal

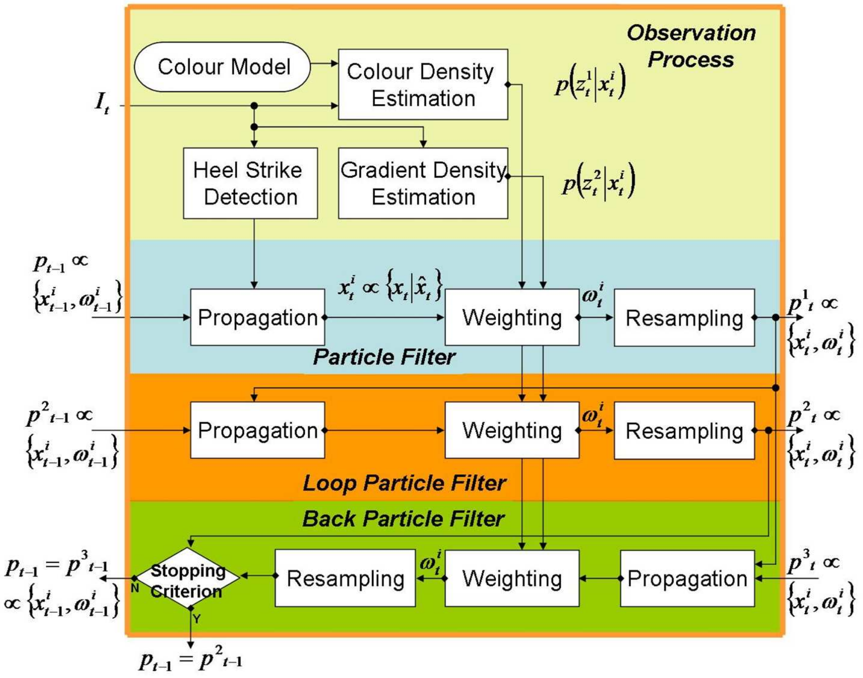

Observation Process
Colour Model
Colour Density Estimation
I_t
$p(z_t^1|x_t^i)$
Heel Strike Detection
Gradient Density Estimation
$p(z_t^2|x_t^i)$
$p_{t-1} \propto \{x_{t-1}^i, \omega_{t-1}^i\}$
Propagation
$x_t^i \propto \{x_t|\hat{x}_t\}$
Weighting
ω_t^i
Resampling
$p^1{}_t \propto \{x_t^i, \omega_t^i\}$
Particle Filter
$p^2{}_{t-1} \propto \{x_{t-1}^i, \omega_{t-1}^i\}$
Propagation
Weighting
ω_t^i
Resampling
$p^2{}_t \propto \{x_t^i, \omega_t^i\}$
Loop Particle Filter
Back Particle Filter
$p_{t-1} = p^3{}_{t-1} \propto \{x_{t-1}^i, \omega_{t-1}^i\}$
Stopping Criterion
N
Y
Resampling
ω_t^i
Weighting
Propagation
$p^3{}_t \propto \{x_t^i, \omega_t^i\}$
$p_{t-1} = p^2{}_{t-1}$

Figure 6.37: Multiple particle filter framework.

estimation, the measurement of its likelihood function is obtained. The reduction of reliability over time is introduced as an exponential decreasing function that multiplies the estimation likelihood (see Figure 6.38). This decreasing function responds to the intuitive idea that the backward tracking is more reliable near the end of the step that the forward one and viceversa. Therefore, comparison between the forward and backward tracking is made, and a stopping criterion is established: when the measurement value (using the colour likelihood, the edge one or both combined) of the backward tracking is lower than the forward one, the reverse tracking process is stopped.

$$f_{reliable}(t) = e^{-(1-t)} \tag{6.28}$$

6.3.3.3 Predictive motion model

Although a simple first order dynamic model is clearly insufficient, this approach is valid over a short period of time. It would be only possible to introduce realistic dynamics in 3D models, but even in those cases their modelling is complex and not much robust and reliable. Therefore we use these linear models for tracking location, size and angular parameters. The responsibility of introducing the non-linearities fall on the biomechanics constraints previously exposed, such as the hip distance $D(\theta)$ or the size constraint in the auxiliary tracker.

6.3.3.4 Likelihood function

An adequate likelihood function must be applied to track the targets. In order to weigh each hypothesis, several visual features are combined: colour and edges (see figure 6.39). Colour is

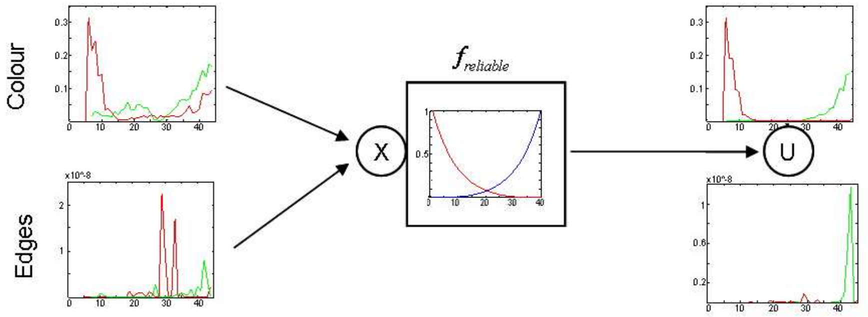


Figure 6.38: Stopping criterion established to stop the auxiliary trackers by combining edge and colour likelihoods.

a discriminative feature which differentiates between object and background, but also between objects. Moreover, it is pose invariant. Edges also provide a good visual feature due to the continuity of the human limbs. Because of their invariance to colour, lighting and pose, they are especially useful to deal with self-occlusions between limbs.

Since we assume these features are independent from each other, we can combine them to obtain the observation probability:

$$p(z_t|x_t) = p(z_t^1|x_t) \cdot p(z_t^2|x_t) \tag{6.29}$$

where z_t^1 and z_t^2 are the colour and edge observations respectively.

Colour features are obtained by sampling each region by a grid and expressing the colour information by RGB values subsampled to 4 bits to filter out noise and small variations. The colour density is measured by comparing the colour feature of each region of the articulated model with its corresponding colour model. It is evaluated by estimating the Bhattacharyya coefficient between their histograms.

$$p(z_t^1|x_t) = \prod_{r \in X_t} (\sum_{h=1}^{H} \sqrt{r(h) \cdot q(h)})^{\alpha_c} \tag{6.30}$$

where r is a body part from the articulated model, H are all the histogram bins, $r(h)$ is the current histogram, $q(h)$ is the reference model and $\alpha_c > 0$ is an empirical factor to strengthen the discriminative power of the feature.

A gradient detector is used to detect edges, and the result is thresholded to eliminate spurious edges. The Canny algorithm is applied for this purpose. The result is smoothed with a Gaussian filter and normalised between 0 and 1. The resulting density image P^e assigns a value to each pixel according to its proximity to an edge using an adaptation of the Euclidean distance transform.

$$p(z_t^2|x_t) = \prod_{r \in X_t} \frac{1}{N} \sum_{i=1}^{N} P^e(I,i)^{\alpha_e} \tag{6.31}$$

where r represents each of the regions which compose the articulated model, N are all the pixels which compose the region and $\alpha_e > 0$ is an empirical factor similar to α_c. Both factors depend on the clutter and the similarity with the background. If we trust more in one of the feature for our particular sequence, a higher value will be assigned in its probability density function. Otherwise, the same value can be assigned to both factors.

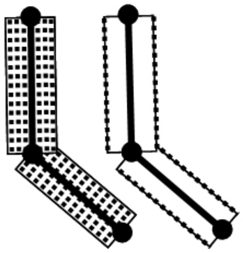


Figure 6.39: Configurations of the pixel map sampling points for the edge and the colour measurements. The sampling points for the edge measurements are located along the contours of the regions which compose the articulated model. For the colour measurements, the sample points are taken from a grid sampling these regions.

6.3.4 Results

The algorithm has been tested over the HumanEva (HE) dataset I and II. These datasets have been produced to make available a benchmark to the scientific community to evaluate and compare different tracking and pose recovery methodologies. As motion capture and video data were collected synchronously, motion capture data provides the groundtruth. Since cameras are calibrated, ground truth data points can be projected on the 2D sequences in order to evaluate quantitatively 2D pose estimates. A standard set of error metrics [Sigal and Black, 2006] is defined that can be used for evaluation of both pose estimations and tracking algorithms. To ensure fair comparison between algorithms that use different numbers of parts, only predicted joints are included in the error metric.

6.3.4.1 Datasets

This dataset contains multiple subjects performing a set of predefined actions with a number of repetitions. Of the order of 100,000 frames of synchronised motion capture and video were collected at 60 Hz.

Since the pose of a human body can be represented using M virtual markers, the state of the body can be written as $X = x_1, x_2, \dots x_M$, where $x_m \in \Re_2$ (2D body model is used) is the position of the marker m in the image. The error between the estimated pose $\hat{X}$ and the ground truth pose X can then be expressed as the average absolute distance between individual markers. To ensure fair comparison between algorithms using different numbers of parts, we added a binary selection variable per-marker $\hat{\Delta} = \hat{\delta}_1, \hat{\delta}_2, \dots \hat{\delta}_M$. Therefore, the final proposed error metric is:

$$D(X, \hat{X}, \hat{\Delta}) = \sum_{m=1}^{M} \frac{\hat{\delta} \| x_m - \hat{x}_m \|}{\sum_{i=1}^{M} \hat{\delta}_i} \tag{6.32}$$

where $\hat{\delta}_m = 1$ if the evaluated algorithm is able to recover marker m, and 0 otherwise.

For a sequence of T frames we can compute the average performance, ν_{seq}, and the standard deviation of the performance, σ_{seq}, using the following equations:

$$\nu_{seq} = \frac{1}{T}\sum_{t=1}^{T} D(X_t, \hat{X}_t, \hat{\Delta}_t) \tag{6.33}$$

$$\sigma_{seq} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[D(X_t, \hat{X}_t, \hat{\Delta}_t) - \nu_{seq}\right]^2} \tag{6.34}$$

6.3.4.2 Experimental results and discussion

The algorithm has been tested with 3 sequences from these datasets: *S2_Walking_1_(C1)*, *S2_Combo_1_(C1)* from HE I and *S2_Combo_1_(C1)* from HE II. Since the latest sequence is especially long, we divided it in two parts: part 1, which is at the beginning of the sequence and corresponds to walking, and part 2, which is at the end and shows some balancing, see Table 6.2 for details. These sequences were chosen to include a variety of movements (walking a complete circle and balancing) seen from different points of view and happening mainly outside the camera plane (see Figures 6.41 and 6.42). Although experiments were performed with a number of particles in the Particle Filter ranging from 200 to 500, their number did not affect tracking accuracy once a minimum number is exceeded.

The pivot point detector can produce erroneous locations: the average error is 20.4 pixels. Therefore, tracking can be affected negatively by this initial process. To analyse independently the tracking algorithm, results are also provided where manual annotation was used to define pivot points (see Table 6.2). The mean error is around 15 pixels. It increases to 18 pixel when tracking is combined with automatic pivot point detection.

Figure 6.40 depicts the error of our tracking framework compared with a version using only the standard tracker. Not only does our system perform significantly better, but this chart also highlights one of the strength of our proposition: tracking is able to recover from serious divergence because of the partial reinitialisation provided by detection of pivot points and trajectory constraints. Although the tracker diverges around frame 200, where limbs reach their apparent maximum size and are self-occluded, legs are accurately labelled on frames 219 and 242 (see Figures 6.40 and 6.41). It is solved automatically due to the reinitialising mechanism that the pivot point provides. Regarding the error increase around frame 200, this is due to a bad estimation of the leg size that produce a consequent fitting error. An external bounding-box tracker which fix the size of the person could be a possible solution for this error. We propose its implementation as future work.

Table 6.2: Numerical results with manual and automatic pivot point detection

		Manual pivot point		Automatic pivot point	
Sequence (C1 camera)	Frames	Absolute Mean Error [in pixels]	Standard Deviation [in pixels]	Absolute Mean Error [in pixels]	Standard Deviation [in pixels]
S2_Walking1, HE I	[6, 418]	17.1	8.7	25.9	10.9
S2_Combo1, HE I	[1661, 2054]	9.3	5.8	9.1	6.2
S2_Combo1, HE II	[1, 307]	25.1	12.4	25.8	12.3
S2_Combo1, HE II	[747, 1202]	9.7	2.3	10.0	3.1
Total	1570	**14.6**	**9.9**	**17.6**	**14.7**

Table 6.3: Comparison with state of the art

Algorithm	Dataset	Pixel error	Constraints	Training	Initialised
Manual pivot	HE I	13.2	Bipedal motion	No	Yes
	HE II	15.9	Bipedal motion	No	Yes
Automatic pivot	HE I	17.5	Bipedal motion	No	Yes
	HE II	17.7	Bipedal motion	No	Yes
Lee et al.	HE I	5-7*	Activity specific & cyclic	Yes	No
Howe	HE I	12.5	Activity specific	Yes	No
	HE II	18.5	Activity specific	Yes	No
Pope et al.	HE I	10-14*	View and activity specific	Yes	No
	HE II	17-20*	View and activity specific	Yes	No
Husz et al.	HE I	33	Single calibrated camera	Yes	Yes
	HE I	14.8	Multiple calibrated cameras	Yes	Yes
	HE II	19	Multiple calibrated cameras	Yes	Yes

* Pixel error estimated from 3D error

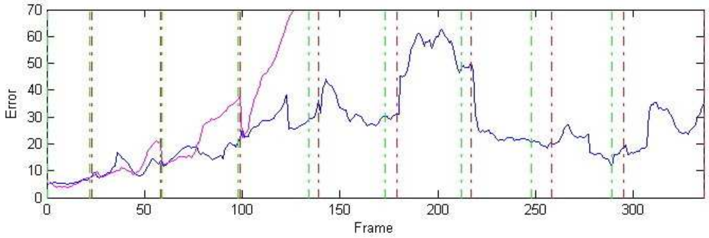


Figure 6.40: Tracking error for each frame of the part 1 of S2_Combo_1_(C1) (HE II) sequence. Green and read lines are respectively the manual and automatic detection of the begining/end of a 'step'. Magenta line is the error using only the standard tracker. Blue line shows the error using the whole framework.

Table 6.3 shows how our results compare with other techniques [Lee and Elgammal, 2006; Husz et al., 2007; Poppe, 2007; Howe, 2007a] used to recover either 2D or 3D poses from the HumanEva data sets. When authors only provided mean errors for 3D poses, they were converted in pixels using approximate relationships between pixel and object lengths for each of the HumanEva datasets.

Most methods perform similarly on the HumanEva datasets, i.e. a pixel error in the 12-15 and 17-20 ranges for respectively HE I and HE II. The only exception is Lee and Elgammal's [Lee and Elgammal, 2006] work which relies on a manifold whose topology is learned using a training set. Their technique performs extremely well: joint mean accuracy is 31 mm, i.e. 5 to 7 pixels. The main drawback of their approach is it relies on a walking scenario or more generally on cyclic activities that have to be learnt explicitly. Therefore, it remains to be investigated how their approach can be extended to deal with open and non-repetitive actions.

The hierarchical particle filter proposed by Husz et al. [Husz et al., 2007] relies on a motion model based on action primitives which predicts the next pose in a stochastic manner. Although their tracker performs similarly to ours when 2 or more camera sequences are available, its performances degrade significantly when processing a single sequence. Both Howe [Howe, 2007a; Howe, 2007b] and Poppe et al. [Poppe, 2007] present example-based approaches to pose

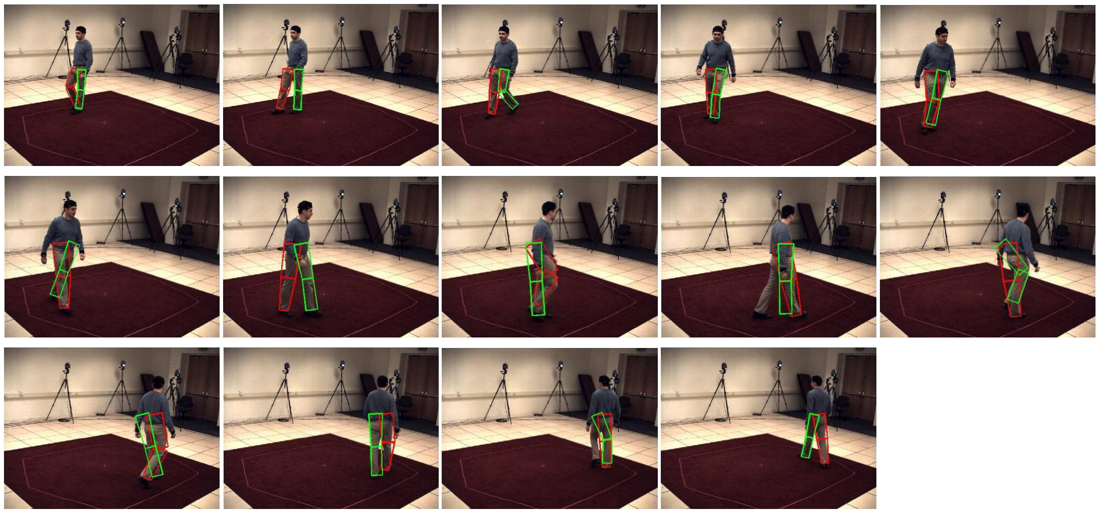


Figure 6.41: Results for *S2_Combo_1_(C1)* (HumanEva II) sequence. Frames: 1, 10, 29, 45, 81, 95, 125, 154, 175, 194, 219, 242, 272, 290.

recovery. They use very different image descriptors, respectively silhouettes and histograms of oriented gradients, but their results appear to be quite similar. The main drawback of these methods is they are action specific and therefore they may not be able to track individuals which display either unexpected motions or a combination of motions. It is also important to notice that like many 3D algorithms, Poppe's assume that the location of the silhouette in the image is provided by an auxiliary image tracker. Therefore, the results they present do not take into account the lack of accuracy such a tracker would produce. One strength of our methodology is that all processing steps are fully integrated. Finally, since our framework is based on a generative approach, no training phase is required. Thus, our system can recover human poses of unusual movements as shown in Figure 6.42 and 6.44.

We have shown that a 2D model constrained by human biomechanics can be as efficient at tracking 3D motions as 3D models. Not only does the use of a 2D model reduces the computation complexity of tracking human body parts, but also simplifies the tracker initialisation. Whereas techniques have been proposed for automatic recovery of postures in 2D, the equivalent 3D task is far from a solved problem and usually relies on a set of calibrated cameras.

As drawbacks, our system has been designed for bipedal motion and the arms has not been taken into account. In addition, one of the strengths of the method is the capacity of reinitialising at each step, which reduces the probability of divergence. However, if the pivot point is occluded by a static object or another subject for a long period of time, then the method can suffer a decrease in its performance. In this situations a more advanced reinitialised method could be necessary [Kuo et al., 2008].

6.4 Conclusions

In this chapter we have explored the possibilities than generative and discriminative approaches offer us. Due to the thesis's goal, only 2D approaches have been tested, more suitable for video surveillance purposes. However, all the proposed methodologies allow the extraction of the human pose independently of the viewpoint or the subject location, which has involved an

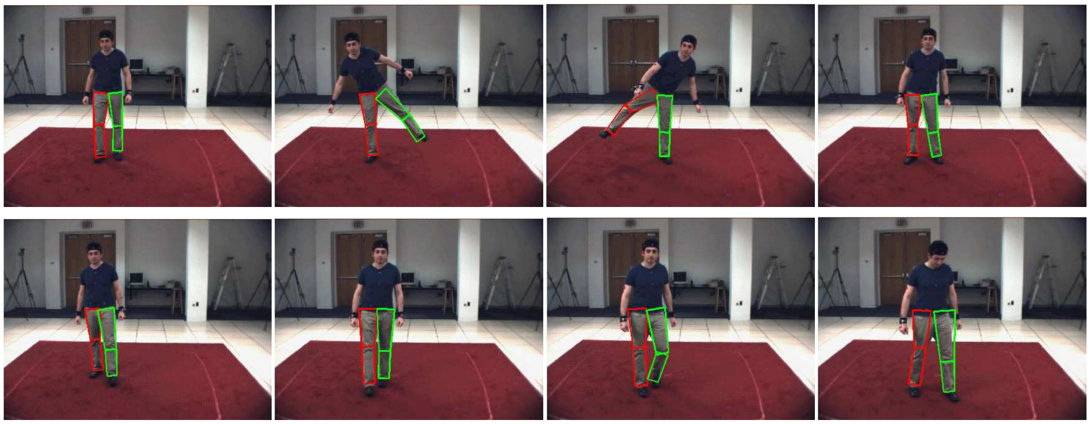


Figure 6.42: Results for *S2_Combo_1_(C1)* (HumanEva I) sequence. Frames: 1661, 1730, 1853, 1920, 1967, 2021, 2045, 2073

extra challenge but is a requirement to be useful in real applications.

Initially, we have introduced a tracking and pose recovery framework based on learned PDM models. In spite of their robustness, their inability to generalise new poses from the learned model has been in evidence. To solve this drawback, a generative approach, which is not limited by a training pose dataset, has been tested. Although their flexibility to track new movements has been proved, we should remark the difficulty that implies this kind of approaches, which requires the introduction of constraints, and morphological and biomechanical information in order to obtain a successful result. The fact of employing 2D models makes more difficult the application of human gait knowledge and does not permits the introduction of the classical biomechanical restrictions.

In the first section, we have developed a statistical model for human silhouette and the corresponding 3D skeletal structure. This model can be used to determinate the pose and structure of the human body from a monocular view. The problem of non-linear principal component analysis is partially resolved by applying a different PDM depending of pose estimation; frontal, lateral, diagonal, estimated by a linear discriminant analysis. Additionally, the fitting is carried out by selecting the closest allowable shape from the training set by means of a nearest neighbor classifier. However, to cope with this problem we consider the necessity of using non-linear statistical models as proposed in [Bowden and Sarhadi, 2000].

To obtain binary regions we do not base the final result to the appropriate selection of a threshold, but we automatically update dynamically this parameter in relation to the human silhouette we are iteratively matching. The performance of the model is improved by introducing concepts related to human gait analysis which take into account temporal dynamic to track human body. The incorporation of temporal constraints on the model increases reliability and robustness. A truly 3D skeletal structure model allows us to predict hidden human body parts in 2D images. Experimental results show the goodness of the present method. However, to improve tracking of a person in complex images where small body movements can cause huge discontinuities in the feature points, we have considered a more complex human motion analysis and also to use a particle filter for tracking.

These improvements have been proposed in the next section. We have proposed a system for view-independent monocular human body tracking. The present approach is based on Rao-Blackwellised particle filter as global framework to integrate stochastic and deterministic tracking. By combining both methods, a more efficient algorithm is achieved with a lower

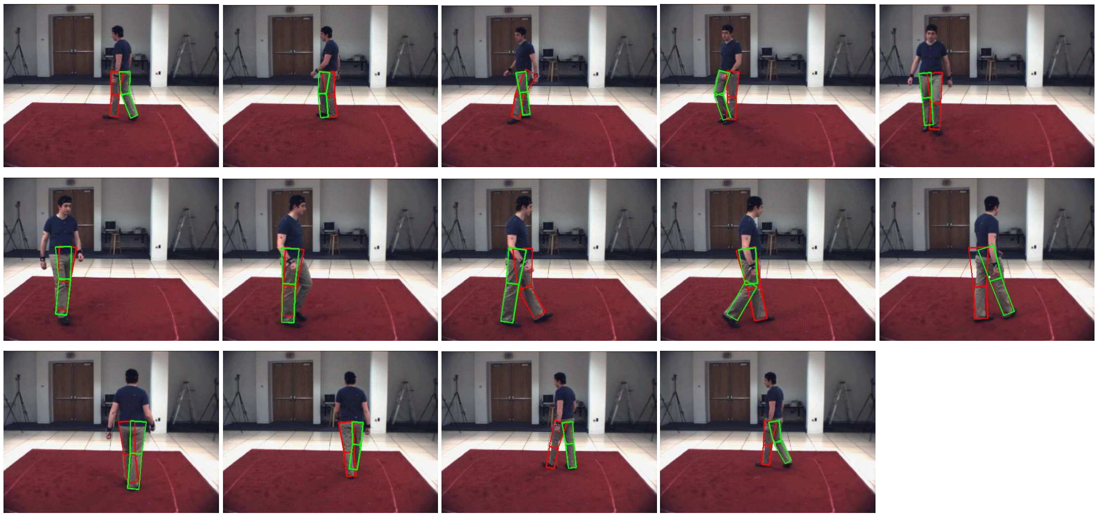


Figure 6.43: Results for *S2_Walking_1_(C1)* (HumanEva I) sequence. Frames: 6, 28, 75, 107, 152, 198, 230, 241, 256, 285, 320, 370, 407, 423.

computational cost.

We have developed an algorithm capable of solving the location in the image using PF, and using this information to estimate analytically the body pose. The human body model applied has been constructed as a PDM with temporal and spatial constraints. The system has been evaluated on real sequences of pedestrians. Experimental results have been reported with satisfactory outcomes.

As future work, we plan to extend the model to situations with multiple people and occlusions, where multimodality can really demonstrate its utility. Furthermore, we will include a more complete set of actions and motions into the analytical model. In addition, we will deal with the automatic initialisation.

After exploring the discriminative approaches, a generative proposal has been studied. The second part of the chapter introduces a novel framework based on a set of particle filters to track human body parts from a single camera. The results presented here demonstrate the feasibility of recovering human pose using a 2D limb tracking system on the basis of articulated models constrained only by human biomechanics. Not only does the use of a 2D model reduces the computation complexity of tracking human body parts, but also simplifies the tracker initialisation. The presented approach has been successfully applied to walking and balancing sequences which include changes of view in the 3D world. Moreover, its accuracy is comparable to other systems relying on constraints incompatible with most video surveillance applications. Therefore, our framework yields potential for tracking human body segments in those applications where motions are generally bipedal.

In future work, we want to tackle the tracker initialisation. We will build on the already existing methods [Ramanan et al., 2007] which have been proposed to detect automatically limbs from an individual either from still or sequence images. In particular, we will focus on bottom up strategies which have the advantage of not relying on a training stage. We will also extend our framework so that tracking can be performed when short 'pivot' point occlusions and temporary loss of foot contact, e.g. in running, occur. In addition, we plan to take advantage of the combination of discriminative and generative approach, which has been shown as a promising research line [Sigal et al., 2007].

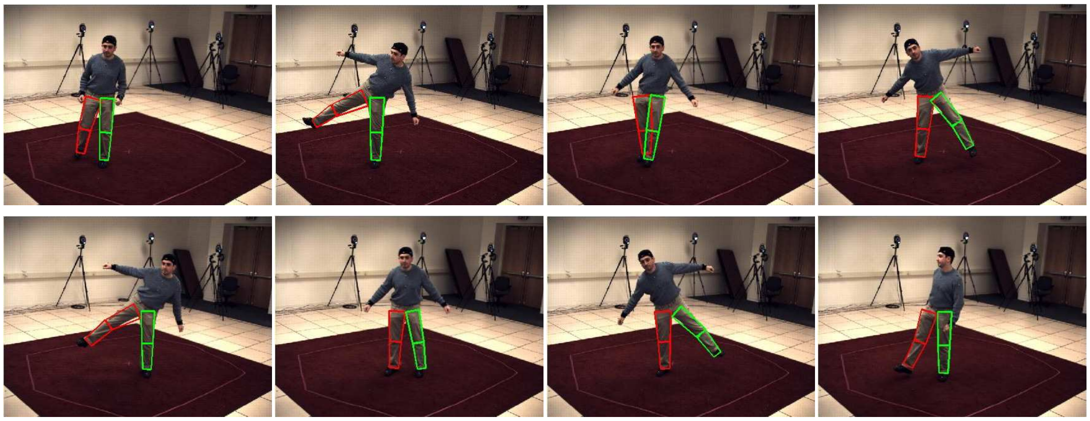


Figure 6.44: Results for *S2_Combo_1_(C1)* (HumanEva II) sequence. Frames: 748, 807, 866, 915, 1004, 1051, 1131, 1220.

7

Conclusions, Discussion and Future Work

"Write therefore the things you see, what they are and signify and what is to take place hereafter."

-Revelation 1, 19-

In this chapter the main conclusions of our work will be presented. Furthermore, a comprehensive discussion of the most important results will be carried out. Finally, new ideas arised along this work will be raised for future possible researches.

7.1 Summary of Work and Result Discussion

The objectives can be summarised in two lines: know the previous techniques for each one of the task involved in the problem and propose improvements in every one. Obviously the work is broad and ambitious but we have exploited the help and knowledge of the scientific team that composes the research group where this thesis has been developed.

In punctual domain, we have developed a system capable of tracking people successfully, in spite of bad measurements or poor quality sensors, thanks to the combination of a static object detector, a height estimator and a multicamera conjugation algorithm. The system has been designed for surveillance applications and even the camera calibration has been simplified as much as possible to fulfil this requirement. The algorithms demonstrated their performance obtaining the best result in the Performance Evaluation of Tracking and Surveillance (PETS 2006) competition [Ferryman et al., 2006].

Regarding the feature extraction field, we have made an exploratory work, with special emphasis on motion and colour extraction. In addition, we have checked the suitability of features such as corners or gradients for human tracking. Colour modelling has been the basic clue to track the targets due to the generality and invariance that these features provide. Different parametric and non-parametric methods have been used to model the appearance of the target, being both useful for different applications. Finally, a robust colour update technique has been presented, which is able to adapt itself to both fast and slow changing illumination conditions. This approach is valid for parametric and non-parametric modelling and it bases its robustness on a feedback mechanism with the tracking algorithm.

An efficient colour tracking algorithm, based on particle filter, has been proposed to speed up the computation of the conventional version of this multi-hypothesis algorithm. The inclusion of techniques such as partitioned or importance sampling reduces the number of samples since they discard those hypotheses with low probability. On the one hand, the use of a probability density image as importance function permits to place the hypotheses in those locations with high probability, improving the a priori information and avoiding bad estimations due to bad dynamical models or unexpected movements. On the other hand, the partitioning of the state vector reduces drastically the dimensionality of the problem, obtaining better results with a lower number of particles.

In addition, the inclusion of the integral image in the evaluation procedure minimises significatively the computational time of evaluating each particle. In this sense, we should remark the modification that we have introduced in order to evaluate rotated masks or arbitrary shapes, solving thus the main drawback that the integral image had. The usage of a laplacian mask helps to a more accurate estimation of the target, specially in the scale space.

An incursion in multi-target tracking has been done. Identical targets have been assumed, given a specific scenario on sport analysis, which implies a higher degree of complexity. In this context, the efficiency of the tracking framework has been maximised thanks to the use of a unique colour model for several targets. Because of our philosophy of efficient and robust applications, independent targets have been used to track the targets and employing auxiliary techniques to deal with the coalescence and help the tracking. Thus, multiple hypothesis tracking, auction algorithm or an extra Kalman filter onto the real world have been implemented to feed back information from all the target trackers and solve ambiguities.

However, the high complexity of multi-target tracking has demanded multi-sensor systems. A multi-layered particle filter has been proposed as an efficient way to insert these interaction functions in a joint framework, as well as to conjugate multiple sensor information. Moreover, this technique has been proved useful to remove distracters for one-object tracking problems under extreme conditions. A football tracking application has been presented, where most of the previous improvements have been combined.

Finally, articulated models have been employed to track not only the global motion of the target but also the relative motion of the limbs. In this domain, 2d models have been proposed since they are much more adequate for surveillance purposes, being able to work in monocular sequences, have a lighter computational load and require a simple initialistaion. For that, the main drawback that has relegated these techniques, i.e the viewpoint dependence, has been tackle in depth. Morphologic and biomechanical information, introduced as part of the model or by means of constraints, allows the achievement of this goal.

The two possible methodologies, discriminative and generative approaches, have been tested and compared. A discriminative approach based on an active shape-skeleton model was trained using walking sequences. The inclusion of gait information into the PCA feature spaces enables a coherent sequentiality. It is worthy to remark the improvement of the motion detector by means of a selective threshold based on morphological information. This discriminative model has been integrated in a Rao-blackwellised tracking framework in order to combine the global and the relative tracking, and improving the final results with respect to the isolated tracking algorithms.

To remove the activity dependence that the discriminative approaches introduce, a generative approach has been presented. This proposal does not have any restriction regarding the activity or the viewpoint. On the contrary that the previous proposal, the biomechanical information is not learned during a training phase. Instead, it is introduce by means of hard and soft constraints to the allowed movements. A more general but less robust tracking was obtained.

7.2 Objective Fulfilment

In this section we remember the objectives that we proposed in Chapter 1 and analyse the degree of fulfilment. We can consider that all the objective have been fulfilled, in a higher or lower degree, even though it is obvious that all of them are open topics which are still under research by the scientific community.

Exploratory work As we mentioned in the introduction, this thesis has a clear exploratory purpose. Therefore, the first objective was to obtain a solid background in human tracking and analyse the weak points in order to introduce novelty in those areas. We think we have fulfilled this objective and we have acquired a clear vision of the tracking field. Thanks to the learned knowledge, we are able to decide the most suitable tracking algorithm for each specific problem, on the basis of their strengths and drawbacks.

Tracking domains Besides the acquisition of a global vision, we have analysed the tracking problem and proposed an organisation based on three levels of understanding. This organisation allows us to adapt the system to the particular conditions of the scenario and provide the needed tools to solve the particular problem. For each level we have introduced some novelty by means of several proposals, algorithms and specific applications, such a left luggage detection algorithm or a sport analysis application.

Real scenarios We have stressed the importance of developing algorithms capable of working under real conditions. To achieve this purpose, all the proposals that we have made have the clear intention of increasing the efficiency and the robustness regarding the conventional techniques. This premise has been implemented over all the domains, always finding the solution which permitted to solve the problem in a more efficient and robust way. Thus, a multi-sensor system has been implemented to deal with the problem of bad quality or poor measurements. In addition, an efficient particle filter has been proposed to reduce the number of hypotheses and their computational load, and finally 2D human pose recovery approximations have been presented to extend their application to video surveillance and real scenarios. Another clear example is the application developed for the ASTRO project, where several efficient techniques have been combined to obtain a successful result.

Feature extraction An exhaustive study on feature extraction has been done, with special emphasis on colour modelling due to the versatility that it provides. Colour feature extraction is variable and adjustable depending on the quality of the scenario. Since the main purpose was to help the tracking algorithm and adapt to different environment, we think a good job has been done.

Human modelling The human morphology has been modelled using a set of different representation, from a simple point or a set of connected regions up to a complex planar articulated model. Special emphasis has been done in Chapter 6, where three human model as well as several human and biomechanical constraint have been implemented. We have checked the indispensable role that they play for obtaining a successful track. Therefore, we think this goal has been fulfilled.

7.3 Conclusions and Justification

Several conclusion can be inferred of our work:

Modelling Human modelling allows a high range of variability, from fine and detailed articulated models to coarse approximations like a bounding box. The accuracy of the model will be defined by the application at which the model is aimed and the quality of the available measurements. The chosen model has a non-negligible importance and contributes significatively to the success of the system. Selecting the suitable tracking domain is therefore an election which should be taken carefully. As example, we can see how the fact of using a laplacian mask instead of a conventional rectangular mask provides a much more robust and accurate tracking system, specially in the scale space.

Unified frameworks Unified frameworks allow a simultaneous estimation of several parameters using different methodologies but with the same objective: the global optimisation of the whole parameter space. In this manner, stochastic and deterministic techniques can work collaboratively to improve the joint result. In addition, these frameworks allow the insertion of feedback techniques to re-estimate the previous results by including external help, such as interaction functions.

Multi-target tracking The coalescence inherent to a multi-target tracking problem increases exponentially the complexity of human tracking. The difficulty that this problem implies makes necessary the use of as much information as possible. Thus, multi-sensor systems help to solve these situations as long as they are independent. A way to simplify the problem consists in assuming independent tracking but with a posterior feedback of the interaction between targets. In spite of all these techniques, extremely complex situations exist where it is not possible to guarantee a correct solution.

Features A good feature extraction algorithm is a cornerstone to achieve a successful tracking, in a manner that a theoretical perfect extractor could simplify the tracking up to a point that it could be unnecessary. Colour and motion have been proved as the most versatile features in all the tracking domains. Gradients are useful for limb tracking, specially in presence of self-occlusions. Finally, corners are not the best choice for tracking deformable targets as human being, but they are a useful tool for detecting static areas an introducing gait constraints.

Parametric vs non parametric On the one hand, parametric models are less time-consuming, but they require a more complex initialisation process, where some parameters must be tuned. On the other hand, non-parametric models are extremely simple to be initialised and updated, but it can introduce more noise in the model. As conclusion, we can say that both methodologies are equally valid, depending the election on the specific application. If we have some a priori knowledge about the model or the target, parametric models are more suitable, like for instance in sport application. However, if the target can take any arbitrary distribution, non parametric models are more flexible, like in surveillance applications.

Discriminative vs generative Although both approaches are valid, we should know their differences and drawbacks. Whereas discriminative approaches are more robust, they are activity dependant so they will fail if the training process does not contain enough examples. In addition, their performances usually decrease with the number of activities that the system is able to recover. On the contrary, generative approaches will be able to recover any pose as long as it is physically possible. However they are more prone to divergence. Both methodologies employ human information to recover the pose successfully: generative approaches parameterise this information whereas generative ones infer it from a training set of examples. The knowledge about their differences and advantages will allow us to combine them to improve the results [Sigal et al., 2007].

Human constraints Human constraints are an essential tool not only to reduce the dimensionality of the problem but also to ensure coherence and sequentiality in the human pose recovery. Given the inherent ambiguity that this problem entails, morphological an biomechanical restrictions are basic to obtain a successful tracking of the human pose.

7.4 Contribution and Publication

As part of the scientific method and scientific process, the discovered improvements should be published to be available for the scientific community, which can correct or take advantage of the acquired experience. Then, papers are considered the way to evaluate the novelty and reliability of a scientific work. In order to remark the novelty introduced by this work, the papers are detailed next

7.4.1 Refereed Journals:

Punctual domain tracking:

- C. Medrano, E. Herrero, J. Martínez, C. Orrite, ”Mean field approach for tracking similar objects” In Computer Vision and Image Understanding (Submitted 2007)

Region domain tracking:

- J. Martínez, J.E. Herrero, J. R. Gómez, C. Orrite, C. Medrano, M. A. Montañés, ”Multi-Camera Sport Player Tracking with Bayesian Estimation of Measurements” in Optical Engineering (Submitted 2008)
- J. Martínez, C. Orrite, J.E. Herrero, ”Fast and Efficient Particle Filter for Feature-Based Tracking” in Image and Vision Computing (Submitted 2008)
- J. Martínez, C. Orrite, C. Medrano, ”Rao-Blackwellised Particle Filter for Colour-Based Tracking” in Pattern Recognition Letters (Submitted 2008)

Articulated tracking:

- G. Rogez, C. Orrite, J. Martínez, ”A Spatio-Temporal 2D-Models Framework for Human Pose Recovery” in Pattern Recognition, Vol 41 (9), pp 2926-2944. (2008).

7.4.2 Conferences:

Punctual domain tracking:

- J. Martínez, J.E. Herrero, J.R. Gómez, C. Orrite, ”Automatic left luggage detection and tracking using multi-camera UKF”, in IEEE PETS 2006, New York, USA, 2006.

Region domain tracking:

- J. Martínez, C. Orrite J. E. Herrero, ”An Efficient Particle Filter for Color-Based Tracking in Complex Scenes” in IEEE AVSS’07, London, UK, 2007.

- J.J. Gracia, C. Orrite, J. Martínez, J. Herrero, "Multimodal Appearance Distribution Tracking" , IEEE PETS 2005, Breckenridge, Colorado, 2005.
- J. R. Gómez, J. E. Herrero, M. Montañés, J. Martínez, C. Orrite "Automatic detection and classification of football players", IASTED International Conference on Signal and Image Processing (SIP 2007), Honolulu, USA, August 2007.
- C. Medrano, R. Igual, J. Martínez, C. Orrite "Multi-target tracking with occlusion management in a mean field framework", Eighth International Workshop on Visual Surveillance (VS 2008), Marseille, France, October 2008.

Articulated tracking:

- C. Orrite, J.Martínez, J. E. Herrero, G. Rogez, "2D Silhouette and 3D Skeletal Models for Human Detection and Tracking", in 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge (UK), 2004.
- J. Martínez, C. Orrite, G. Rogez, "Rao-Blackwellized Particle Filter for Human Appearance and Position Tracking", in LNCS-Lecture Notes in Computer Science 4477, Girona, Spain, 2007.
- J. Martínez, J.C Nebel, D. Makris, C. Orrite, "Tracking Human Body Parts Using Particle Filters Constrained by Human Biomechanics", in BMVC'08, Leeds, UK, 2008.
- G. Rogez , C. Orrite, J. Martínez, "Human Figure Segmentation Using Independent Component Analysis" in LNCS-Lecture Notes in Computer Science 3523, 2005.
- G. Rogez, J.J. Guerrero, J. Martínez, C. Orrite, "Viewpoint independent human motion analysis in man-made environments", BMVC'06, Edimburgh, UK, 2006.
- G. Rogez, C. Orrite, J. Martínez, J.E. Herrero, "Probabilistic spatio-temporal 2d-model for pedestrian motion analysis in monocular sequences", LNCS-Lecture Notes in Computer Science 4069, 2006.
- G. Rogez, J. Martínez, C. Orrite, "Dealing with Non-linearity in Shape Modelling of Articulated Objects", in LNCS-Lecture Notes in Computer Science 4477, 2007.
- G. Rogez, I. Rius, J. Martínez, C. Orrite, "Exploiting Spatio-Temporal Constraints for Robust 2D Pose Tracking", 2nd Workshop on Human Motion Understanding, Modeling, Capture and Animation, Rio de Janeiro, Brasil, 2007.

7.4.3 Projects:

In addition, the techniques and methodologies developed have been used in a large set of projects:

R&D Projects

- DGAGE: System of third generation for access control and security by advanced computer vision techniques

  Reference: DGA2002

  Sponsor: Government of Aragón

  Starting Date: 2003-01-01

Ending Date: 2004-12-31

- BIOSECUR-FACE: Human detection, tracking and authentication by facial biometrics and gait analysis

  Reference: TIC2003-08382-C05-05

  Sponsor: Spanish Ministry of Science and Technology

  Starting Date: 2003-12-01

  Ending Date: 2006-11-31

- I-EYE2: Third-Generation Surveillance System at Intelligent Environments by Computer Vision Techniques

  Reference: Profit 390000-2004-30

  Sponsor: Spanish Ministry of Education and Science

  Starting Date: 2004-01-01

  Ending Date: 2004-12-31

- BIOSECURE: European Network of Excellence on Biometrics for Secure Authentication

  Reference: IST-2002-507534

  Sponsor: European Union

  Starting Date: 2004-06-01

  Ending Date: 2007-07-31

- CIPGAL Learning system for parking control with a license plate recognition OCR, damage inspection in cars, and managing the number of available slots

  Reference: CIT-390000-2005-18

  Sponsor: Spanish Ministry of Education and Science

  Starting Date: 2005-01-01

  Ending Date: 2005-12-31

- MON AMI: Mainstreaming on Ambient Intelligence

  Reference: IST-5-0535147

  Sponsor: European Union

  Starting Date: 2006-07-14

  Ending Date: 2009-12-31

- HARMRES: Human Activity Recognition and Modelling in Real Scenarios

  Reference: TIN-2006-11044

  Sponsor: Spanish Ministry of Education and Science

  Starting Date: 2007-01-01

  Ending Date: 2009-01-01

Industrial Projects

- Integration of Traffic Tracking Models on the OVS Platform by Visual Tools

  Sponsor: Visual Tools S.A.

  Starting Date: 2003-07-01

  Ending Date: 2003-12-31

- ASTRO05: Automatic System for Tactical Review and Optimization

  Reference: DGA 0438-6 2005

  Sponsor: Government of Aragón

  Starting Date: 2005-07-01

  Ending Date: 2005-12-31

- ASTRO06: Collaboration project (DGA, UZ and Real Zaragoza) for the improvement of the sport performance based on image analysis

  Reference: DGA 2005/0388

  Sponsor: Government of Aragón

  Starting Date: 2006-01-01

  Ending Date: 2007-12-31

- Vision system for improving the trunk cut

  Reference: 2008/0013

  Sponsor: Maderas Bielsa, Sociedad corporativa

  Starting Date: 2007-12-01

  Ending Date: 2008-5-31

7.4.4 Application Patents:

Finally, an application patent has been obtained as fruit of our work:

Sistema y Procedimiento de Observación Ampliada (Enhanced Observation System)

Patent Number: P200703312

Publication Date: 2007-12-03

Inventor: José Elías Herrero Jaraba, Jesús Martínez del Rincón, Miguel Ángel Montañés Laborda, Jorge Raúl Gómez Sánchez, Ignacio Julve Castro, Carlos Orrite Uruñuela, Francisco Javier Martínez Contreras.

Applicant: University of Zaragoza

Country: Spain

7.5 Future Work

Finally, we want to mention different proposals and ideas that were generated in the course of this doctoral thesis and were beyond the scope of our investigations but could conduct to new research lines. We summarise the points where we see potential for improvements and optimisations:

Initialisation Although some detection concepts have been introduced in this work, it has not been discussed properly assuming manual initialisation in most of the cases. We should distinguish between two different kinds of initialisations: not only the location of the target in the first frame must be provided but also the model that characterises it. Regarding the first issue, it is worthy to note the complexity increase of the initialisation process over the understanding level: from a simple point in punctual domain up to a volumetric articulated model in the most complex level. The second issue also involves several levels of difficulty which, although they are related with the domain, also depend on the observation process. Thus, a simple histogram can be obtained automatically by extracting pixels corresponding to the target, once the location in the first frame is known. On the contrary, a parametric feature model can require a parameter tuning, such as the number of clusters in a Gaussian mixture.

With the purpose of a simple, and in a near future, automatic initilisation, we have simplified the initialisation as much as possible. In punctual domain, its simplicity allows us to implement a completely automatic initialisation, whereas region domain requires a manual process to annotate the bounding box of the region as well as the label. Even in the case of articulated models, 2D models have been applied to make easy the starting process, specially in comparison with 3D models.

Multi target tracking As we have mentioned before, this is probably the hottest topic in human tracking. In Chapter 5 we have presented several proposals to model coalescence and interaction between targets. However, we realised that a lot of work should be done yet to complete this vast task. Figure 5.21 is a clear example of the long way that we have to cover in order to solve real-life scenarios. A deeper study of JPDA, MCMC and Mean field techniques could give us a starting point.

A special emphasis should be done to deal with multi-target in the pose recovery domain. In this domain, not only the interaction between target must be modelled, but also how the occlusion of part of the subject affects to the model fitting and the pose recovery has to be studied. Some concepts to address this problem has been introduced, such as the use of multiple cameras, the application of constraints or the strength that the reinitialisation mechanisms provides, but is obvious than a big effort should be done to obtain useful results for real applications.

Integration In this thesis, we have tried to solve tracking problems for every tracking domain by applying specific solutions under different conditions. Therefore, the logical next step is the integration of all the proposals in a unique framework. An unified framework or a means of automatically selecting the tracking domain, which have not been provided in this thesis, should be able to decide the most suitable tracking domain at each time step and for each target [Gammeter et al., 2008]. In this manner, the system should be capable of changing algorithms and techniques to be applied as the environmental conditions change.

Since the information that each domain provides is incremental, the system could always ensure a minimum level of estimation, but it could employ a feedback between domains to improve the individual results of all of them. The result would be similar to the framework proposed in Chapter 6 to combine shape and location information.

Multi sensor A brief incursion in multi-sensor tracking application has been presented in this thesis (Appendix D). Despite the fact that the results were not completely satisfactory, interesting conclusions were obtained. By selecting adequate sensors, we can mitigate the high ambiguity that cameras contain as sensors. The proposed framework allows the integration of complementary sensor information and provides a method to model their corresponding uncertainties to calculate an accurate estimation. We think that this research field could be a cornerstone in the future of video surveillance just as the human biology, the brain working and the profuse research made in robotics endorse.

Generative approaches The generative articulated model should be extended to model the whole human body including the arms. This would make possible the extraction of the human pose as first step to apply an activity recognition algorithm. Moreover, new human constraints must be introduced to obtain a successful recovery of the human pose without loss of generality, as we have tried in this thesis. We consider the capability of recovering the pose independently of the viewpoint as a requirement and the morphological and biomechanical constraints as the way to achieve it.

Human gait This thesis has been focused on human tracking in the image or in the real world space (by means of a plan representation), that is, physical parameters with spatial/temporal meaning have been tracked and used to build the state vector. Specific modelling and observation processes have been described and adapted to those spaces. However, most of the proposed tracking methodologies are extensive to other parameter spaces, such as the pose space.

The modelling of these spaces, in order to introduce gait constraints, are a basic pillar of the most recent 3D pose recovery methodologies, such as GPLVM or other discriminative approaches. Human tracking in those spaces is a crucial tool to introduce the temporal coherence of the human gait, and to adapt the learned models to the real scenarios. We think that the extension of our work to those feature spaces is an interesting field of research with promising results.

A

Gradients

Gradient extraction is not a new topic, it has been studied exhaustively in the literature. Its importance resides in the fact that gradients summarises a big percentage of the image information like shape or appearance of the target in a more tractable characteristic space.

Gradient detection is based on detecting abrupt changes in the image pixel values due to four main causes:

- Changes in the depth of the objects or surfaces.
- Discontinuity in the surface orientation.
- Changes in the surface reflectance.
- Illumination changes such as shadows or light sources.

Well-known methods like Laplacian, Sobel and Canny are based on punctual operators i.e. they detect the contour using only the information of these pixels and their nearest neighbours. Nevertheless, many times the extracted contours do not fit with the real shape of the object, mainly due to the background clutter or low resolution images. Instead, the resulting contour appears fragmented because of the similarity between the object and the scene.

A.1 Gabor Filters

Gabor filters try to find those points belonging to segments or contours. This is made by weighting each pixel with a value that measures its membership to a contour using a predefined filter. A 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave:

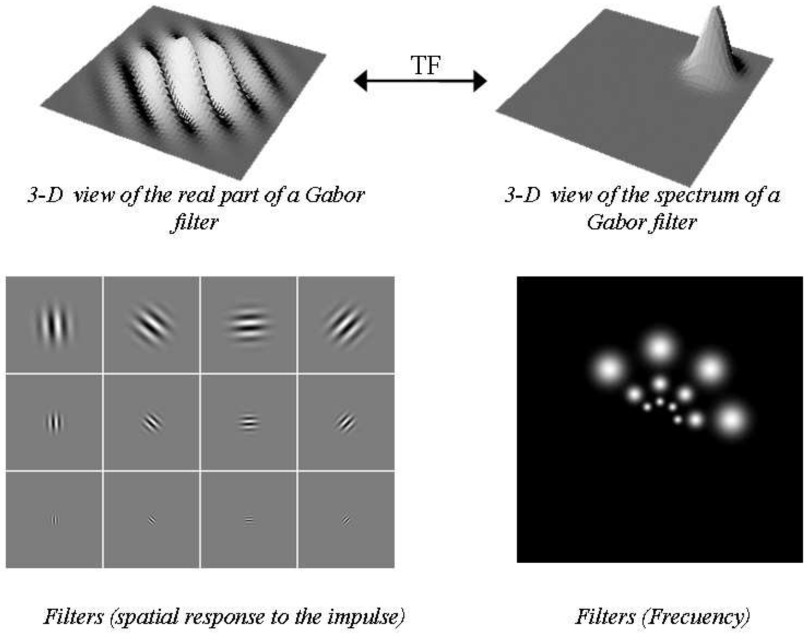
$$\Phi_{u,v}(z) = \frac{\|k_{u,v}\|}{\sigma^2} e^{\frac{-\|k_{u,v}\|^2 \cdot \|z\|^2}{2\sigma^2}} \left[e^{ik_{u,v} \cdot z} - e^{\frac{-\sigma^2}{2}} \right] \tag{A.1}$$

where $z = (x, y)$, and $k_{u,v} = k_v e^{i\phi_u}$ gives the frequency with $k_v = k_m ax/f^v$, and $\phi_u = u\pi/8$ gives the orientation with $\phi_u \in [0, \pi)$.

This filtering process can be made using a convolution in space or a filtering in the frequency domain.

If we apply the 2D Fourier transform over a image, we obtain its spectrum in frequency, where the middle of the image shows the low-frequencies and the border the high frequencies. The presence of lines in this spectrum implies the predominance of lines with a perpendicular orientation in the image. Hence, it is possible to obtain segments of a specific length and orientation in the image by multiplying Gaussian filters centred on the perpendicular line to that one with the desired orientation. The width of the filter depends on the resolution or size of the segments to be extracted.

In order to extract all the segments of different lengths, a Gabor pyramid is applied (see Figure A.1), which is a set of selective filters that provide the gradients of the image for several directions and resolutions. The results obtained for a sequence of images with 3 levels of resolutions and 4 orientations are in Figure A.2 and A.3.



3-D view of the real part of a Gabor filter

3-D view of the spectrum of a Gabor filter

Filters (spatial response to the impulse)

Filters (Frecuency)

Figure A.1: Bank of Gabor filters. (Image source: http://www.neuroinformatik.ruhr-uni-bochum.de/

Results (Figure A.3) permit us to extract some conclusions. Abrupt contours between regions of high contrast are easily detected even using slightly different scales or orientations to the correct ones. On the contrary, partially occluded or camouflaged contours, which should be reconstructed by applying a lower scale, are not linked anyway if the gap has considerable dimensions. The cause is the loss of resolution due to the spatial decimated that the frequency filtering implies. Furthermore, the lateral tails of the filter that appear in space can produce maxima not only in the contour but also inside the object like, for instance, because of creases in the clothes. This secondary contours, although interesting in some applications, usually introduce noise in the main extraction, that is, the external contours of the limbs.

A.2 Oriented Gradients

In order to avoid the drawbacks of the previous method, we build a bank of elliptical filters in concordance with the response that we want to obtain. Since the filtering in this case is a spatial convolution, its application to a larger scale does not imply a loss of resolution but a

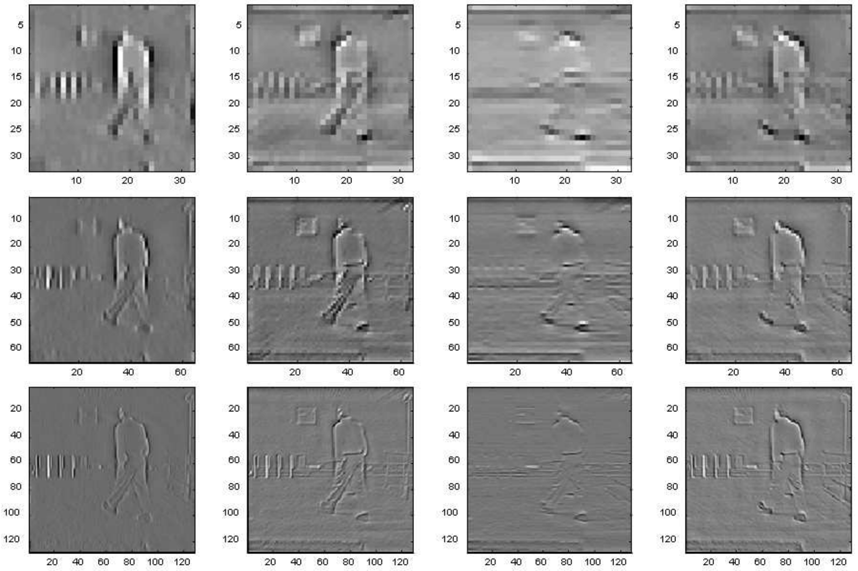


Figure A.2: Results of applying Gabor filter pyramid

search of a contour of a larger length.

Filters are generated as a Gaussian in one dimension and its derivation in the other one. Afterwards, the filter is multiplied by a rotation matrix with the chosen angle. Only angles between 0 and 180 degrees must be sampled since, for instance, the response for an angle of 270^o is the same than the response for 90^o but with a negative sign.

$$H(x,y) = G(x) \cdot \partial G(y) = k \cdot y \cdot e^{\frac{-x^2}{2\sigma_1^2}} \cdot e^{\frac{-y^2}{2\sigma_2^2}} \tag{A.2}$$

where $k = \frac{-1}{2\pi\sigma_1\sigma_2^3}$ is the normalisation constant.

Once the gradients have been computed, the contours are obtained by applying consecutive maximum suppression, a technique employed commonly in the Canny operator. Each contour candidate is evaluated if it is a local maximum in the direction of maximum variation, which it is specified by the gradient. In other words, we calculate the value in the perpendicular to the

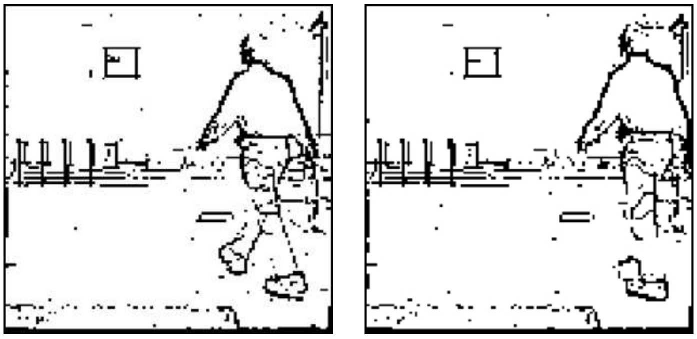
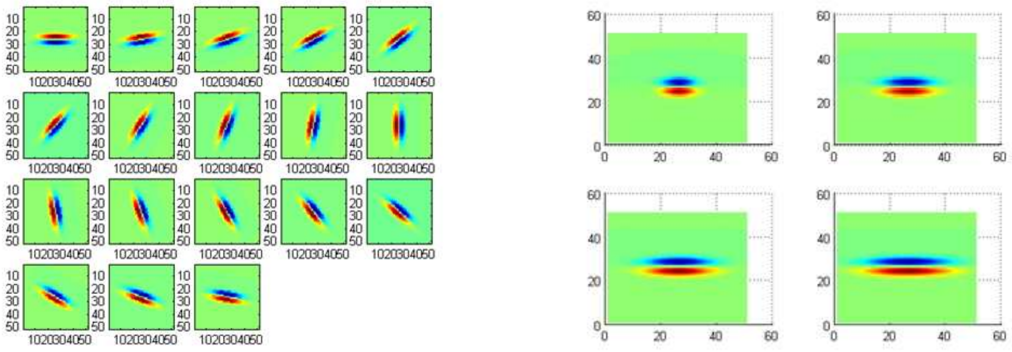


Figure A.3: Thresholded results of Gabor filtering



a) b)

Figure A.4: Bank of oriented filters for: a) different orientations, b) different scales.

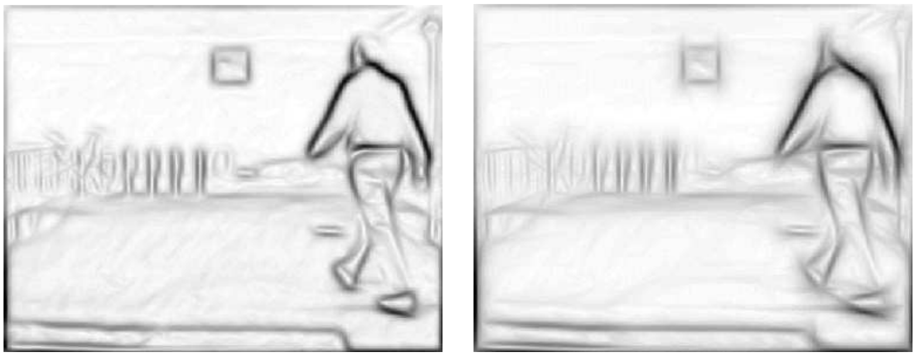


Figure A.5: Oriented filtering results for two different scales.

contour, interpolating the values of the nearest pixels as it is shown in Figure A.6.

Depending on the quadrant, which is fixed by the gradient orientation, the previous and posterior points are calculated interpolating different pixels. If the values G_1 and G_2 are smaller than the pixel value, the point is considered local maximum in the 3x3 neighbourhood and it is labelled as contour. Results for the test image is shown in Figure A.7.

The effect of the different scales can be seen in Figure A.8. The lower scales show small contours but they fragment those ones that are not easily distinguishable. On the other hand, larger scales recompose the contour at the expense of removing the small ones.

A.3 Conclusions

Two method for gradient extraction based on the convolution of Gaussian filters has been tested. Both methods allow extracting contours with invariance to scale, location and rotation. Better results have been obtained using spatial filtering instead of frequency filtering. The decomposition of gradients into a pyramid of filter gives us advantages to split the contours into scales and angles for a posterior fitting or identification of the object.

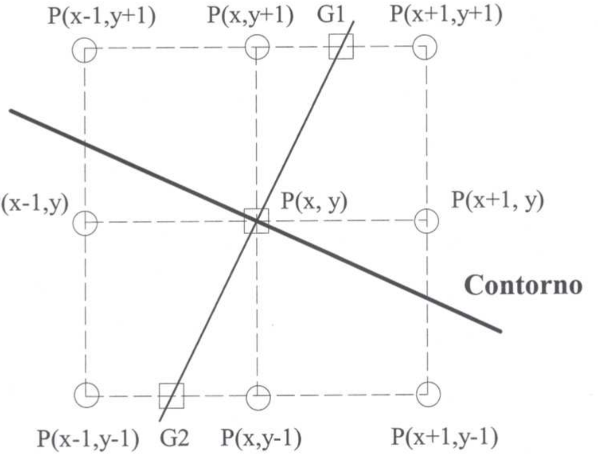


Figure A.6: Non-maximum suppression algorithm.

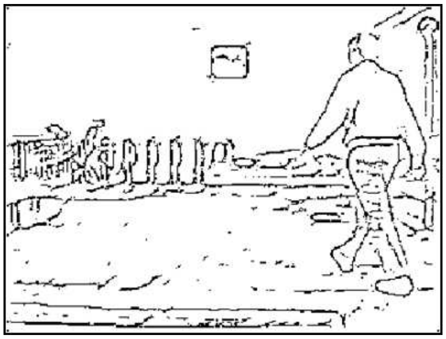


Figure A.7: Extracted contours using the local maximum search.

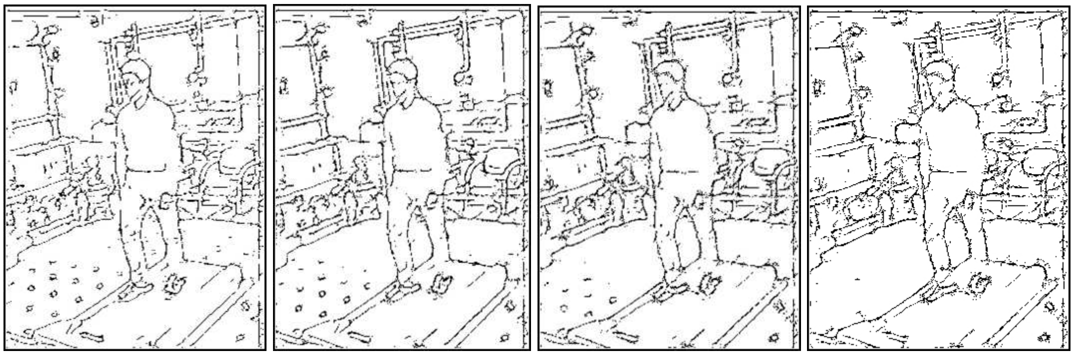


Figure A.8: Extracted contours using four different scales.

B

Corners

Corner extraction has the aim of obtaining invariant and distinctive characteristics which allow establishing correspondences between different viewpoints of the same object in a scene. These keypoints must be so distinctive as possible, such as corners or intersections to characterise the exclusive appearance of the object. In addition, they should be invariant to scale and rotation, and if it is feasible, to affine distortions, 3D viewpoint changes, noise, illumination changes and clutter.

B.1 Lowe Detector

The detector proposed by David Lowe [Lowe, 2004] is divided into 4 basic steps which extract the features of the keypoints:

- **Scale-space extrema detection:** It seeks maxima and minima in the image and in the scale space. A difference of Gaussians is applied to generate that space.
- **Key point localisation:** The points generated in the previous stage are tested to check their stability.
- **Orientation assignment:** One or more orientations are assigned to each point, selecting the maxima in a histogram composed of the gradient directions.
- **Point descriptor generation:** A descriptor is created for each point. It consists in a vector which contains the gradients of the surrounding pixels normalised in scale and rotation with regard to the extracted value in the previous steps.

Thanks to this methodology, the system is invariant to scale and rotation as well as to soft illumination changes and shape distortions.

B.1.1 Scale-space Extrema Detection

In this phase, the possible candidates which can be detected in the different views of the same object must be identified. An efficient extraction of maxima and minima is obtained by generating the scale space using a convolution between the image and a difference of Gaussian function D

$$D(x, y, \sigma) = (G(x, y, k \cdot \sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k \cdot \sigma) - L(x, y, \sigma) \tag{B.1}$$

where k is the difference between two adjacent scales. Since a octave is divided into an integer number of intervals s (see Figure B.1), this parameter is $k = 2^(1/s)$. A octave is calculated multiplying the σ value by 2. The efficiency of the system can be increased sub-sampling the image in each stage instead of doubling the σ value.

A difference of Gaussians is used because it is a good approximation to the laplacian of the Gaussian, and therefore, invariant to scale. [Lowe, 2004; Lindeberg, 1998].

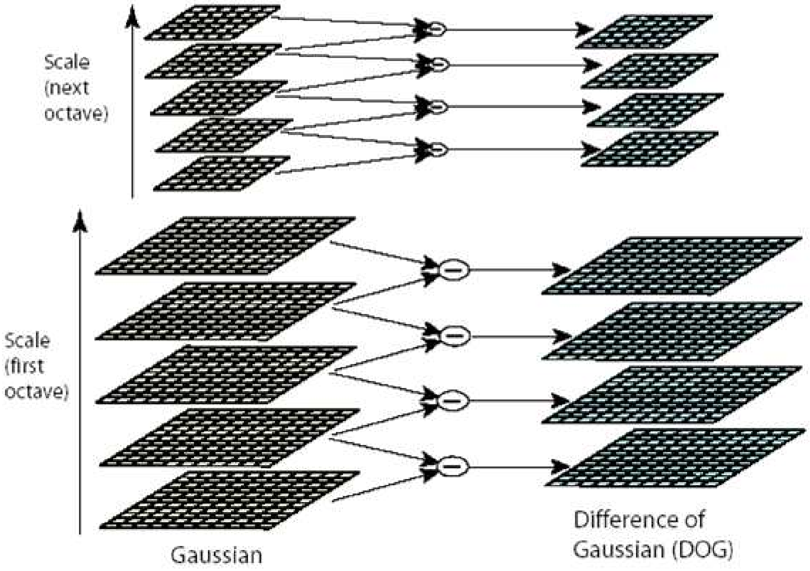


Figure B.1: Scale-space. (Image source: [Lowe, 2004])

The local extrema detection is made comparing each point with the eight neighbours in the current scale and with the nine ones in the scales above and below. That point will be considered a keypoint if it is lower or higher that all of them.

In order to avoid discarding spatial frequencies due to the smoothness of the Gaussian, we can generate a first octave duplicating the size of the original image and interpolating the new pixel values.

B.1.2 Key Point Localisation

Once the candidate point has been detected, we must adjust accurately the location, scale and radius of curvature, which allows us to remove the low contrast points (sensitive to noise) or those located along the edges.

Applying a 3D quadratic function to the local point, we can determine the maximum interpolated location and increase the stability and correspondence. This approximation is based on applying a Taylor expansion to the scale space.

$$D(x) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{B.2}$$

The extrema location is obtained by taking the derivative of this function with regard to x and setting it to zero, giving

$$\hat{x} = \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \qquad D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \tag{B.3}$$

We only consider valid extrema those that fulfil $|D(\hat{x})| < 0.03 \cdot pixel\_value$, removing thus the low contrast points.

Nevertheless, to guarantee the stability of the points, removing the low contrast points is not enough. Points along the contours are unstable in presence of noise but they give strong response levels. These points have a big curvature in the direction perpendicular to the edge and a small one along it. The value of those curvatures is given by the eigenvalues of the Hessian Matrix H, calculated in the scale and location of the keypoint.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{B.4}$$

Being α and β the largest and smallest eigenvalues respectively, we can compute the trace and determinant using their sum and product

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta Det(H) = D_{xx} \cdot D_{yy} - D_{xy}^2 = \alpha \cdot \beta \tag{B.5}$$

If we define r as the relation between both eigenvalues $\alpha = r \cdot \beta$, it is possible to define a relationship which will be minimum when both values are equal and the more it rises the more different they are. In this manner, a candidate is consider keypoint if

$$\frac{Tr^2(H)}{Det(H)} < \frac{(\alpha + \beta)^2}{\alpha \cdot \beta} = \frac{(r \cdot \beta + \beta)^2}{r \cdot \beta^2} = \frac{(r+1)^2}{r} \tag{B.6}$$

Figure B.2 shows how the inconsistent points are removed through the different stages.

B.1.3 Orientation Assignment

Once the scale and location have been determined, we should look for the local orientation of each keypoint in order to achieve rotation invariance. First of all, the adequate scale is chosen, that is, for each point we work with the smoothed image L in that scale. We calculate magnitude and orientation for every pixel

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{B.7}$$

$$\theta(x,y) = \tan^{-1} \frac{(L(x+1,y) - L(x-1,y))}{(L(x,y+1) - L(x,y-1))} \tag{B.8}$$

Using a resolution of 10 degrees, we generate a histogram of 36 bins with the orientations of the pixels surrounding the point. The value of each pixel is included in the histogram weighted by the magnitude of the gradient and by a circular window with a variance 1.5 times the scale. The final orientation assigned to the keypoint is the biggest peak as well as all the peaks whose value is larger than the 80% of the maximum. In those cases when more than one peak is chosen, the keypoint is duplicated with the same parameters but different orientation.

B.1.4 Keypoint Descriptor

The location, scale and orientation parameters configure a bidimensional coordinate system which we can use to describe the region surrounding the characteristic point. Thus, a descriptor for each point is obtained. This descriptor provides highly distinctive characteristics and invariance to the rest of parameter such as illumination conditions or 3D viewpoint.

Although sampling intensity around the point could be employed as normalised correlation measurement, this option is quite sensitive to 3D rotation or non-rigid deformation. Instead, we calculate the module and phase of the gradient in the corresponding scale. We rotate those

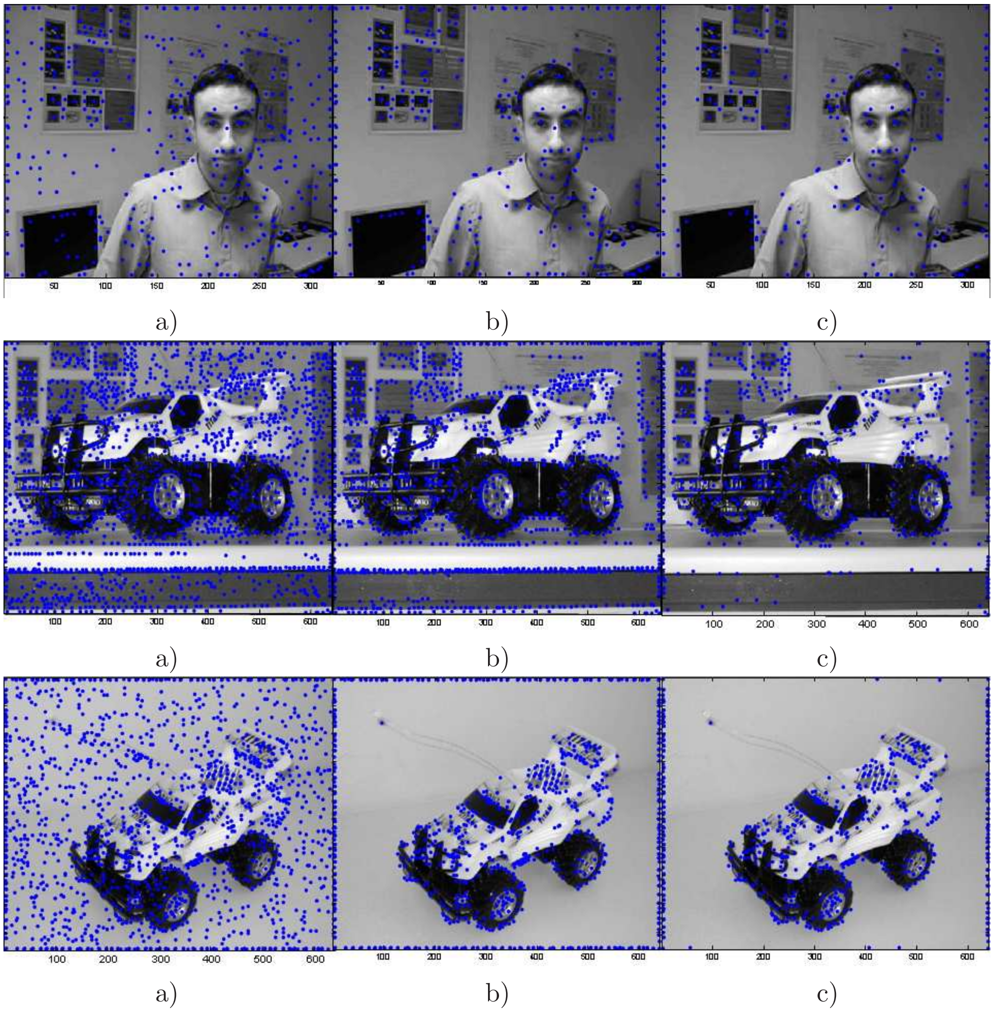


Figure B.2: Keypoints detected. a) Maxima obtained using the multi-scale b) Surviving points removing the low contrast ones. c) Surviving points removing the points on the edges.

gradients to compensate the orientation of the keypoint and we weigh it using a Gaussian function whose sigma is 1.5 times the width of the descriptor. This gaussian function is useful to avoid sudden changes in the descriptor due to small movements of the window. Then, we divide the descriptor in 16 regions containing 4x4 samples and we generate a histogram of 8 bins (45^o resolution) for each one. The descriptor (Figure B.3) thus calculated is a vector of 128 bytes which contains the values of the orientation histograms.

B.2 Harris Detector

Harris detector is a well-known and simple corner extractor. Its simplicity and coherence have converted it in one of the most important characteristic point extractor.

The first step to compute the algorithm is the calculation of the autocorrelation matrix H

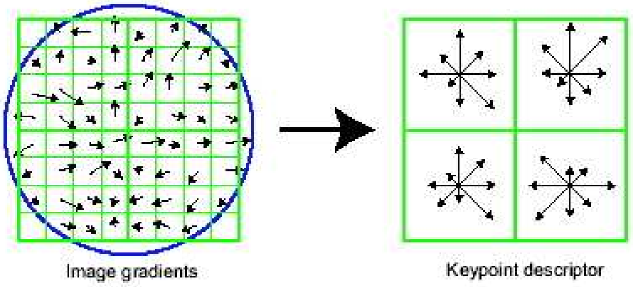


Figure B.3: Keypoint descriptor generation. (Image source: [Lowe, 2004])

for every pixel.

$$H = \begin{bmatrix} I_x^2(x,y) & I_x(x,y) \cdot I_y(x,y) \\ I_y(x,y) \cdot I_x(x,y) & I_y^2(x,y) \end{bmatrix} \tag{B.9}$$

where $I_x(x,y) = \frac{\partial(I(x,y)*G(x,y,\sigma))}{\partial x}$ being $G(x,y,\sigma)$ a Gaussian of variance σ.

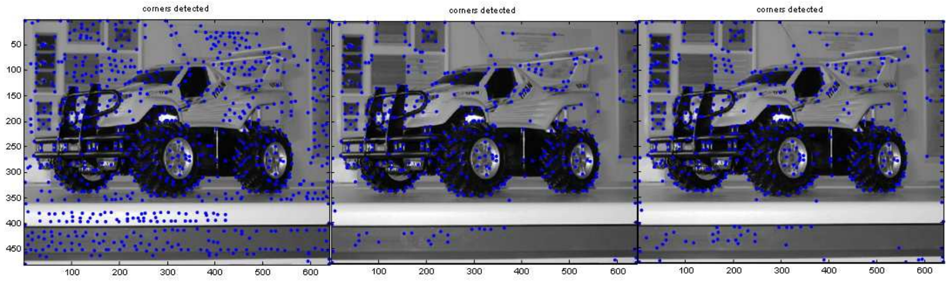
By calculating the eigenvalues of the matrix and selecting the maxima, we can find the keypoints. However, we can avoid this expensive calculation using the trace and the determinant in a similar way that the Lowe detector does.

$$M(x,y) = Det(H) - k \cdot Tr^2(H). \tag{B.10}$$

The local maxima are those pixels whose M-value is larger than the eight-neighbours and larger that a fix threshold, which is useful to remove unstable or low contrast points.

Another variation of this method is to substitute the matrix M by a different one (see Figure B.4) whose main advantage is not to require a parameter k since the calculation of that parameter is quite ad-hoc. For this reason we have selected this second version.

$$M(x,y) = Det(H)/Tr^2(H). \tag{B.11}$$



a) b) c)

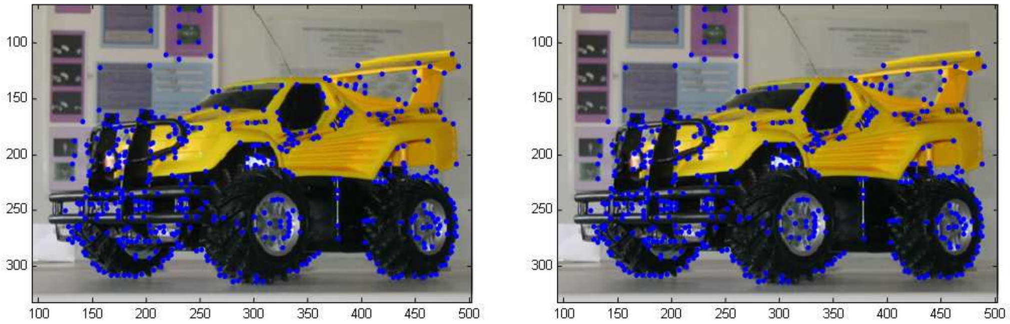
Figure B.4: Keypoints obtained by Harris. a) First approach to the calculation of M and a threshold equal to 1000. b) First approach to the calculation of M and a threshold equal to 100000. c) Second approach to the calculation of M and a threshold equal to 500.

The third approach to the Harris detector is called Tomasi-Kanade variation (Figure B.5) and is based on the calculation of the matrix H over a window around each pixel

$$H = \left[\begin{array}{cc} \sum_{(x_k,y_k)\in W} I_x^2(x_k,y_k) & \sum_{(x_k,y_k)\in W} I_x(x,y)\cdot I_y(x_k,y_k) \\ \sum_{(x_k,y_k)\in W} I_y(x,y)\cdot I_x(x_k,y_k) & \sum_{(x_k,y_k)\in W} I_y^2(x_k,y_k) \end{array} \right] \tag{B.12}$$

Once the keypoints have been detected, an orientation is assigned. For that, we calculate the larger eigenvector and apply the following equation

$$\theta = \arctan \frac{\lambda_1 - I_x^2}{I_x \cdot I_y} \qquad \text{with} \qquad \lambda_1 = \frac{I_x^2 + I_y^2 + \sqrt{(I_x^2 + I_y^2)^2 - 4(I_x^2 \cdot I_y^2 - (I_x \cdot I_y)^2)}}{2} \tag{B.13}$$



a) b)

Figure B.5: Keypoints obtained by a) traditional Harris. b) Tomasi-kanade.

Since Tomasi-Kanade does not provide a significant difference or improvement but implies a bigger computational load, we choose the traditional algorithm.

B.2.1 Colour Harris Detector

Previous algorithms work with intensity i.e. gray images. However, colour is a crucial characteristic, specially in human tracking or face detection. Montesinos et al. [Montesinos et al., 1998] propose a simple extension of Harris detector for colour images in the RGB space, although it is also valid in other colour spaces like HSV or YCbCr.

This modification only needs to change the matrix H to include three channels

$$H = \left[\begin{array}{cc} G(x,y,\sigma) * (R_x^2 + G_x^2 + B_x^2) & G(x,y,\sigma) * (R_x \cdot R_y + G_x \cdot G_y + B_x \cdot B_y) \\ G(x,y,\sigma) * (R_x \cdot R_y + G_x \cdot G_y + B_x \cdot B_y) & G(x,y,\sigma) * (R_y^2 + G_y^2 + B_y^2) \end{array} \right] \tag{B.14}$$

Therefore, the orientation is defined as:

$$\theta = \arctan \frac{\lambda_1 - (R_x^2 + G_x^2 + B_x^2)}{(R_x \cdot R_y + G_x \cdot G_y + B_x \cdot B_y)} \tag{B.15}$$

Figure B.6 shows the comparison between black&white and colour approaches and the improvement that this last approximation implies. Note that many corners (rectangles in the posters, aileron, etc..) only appear in the colour images.

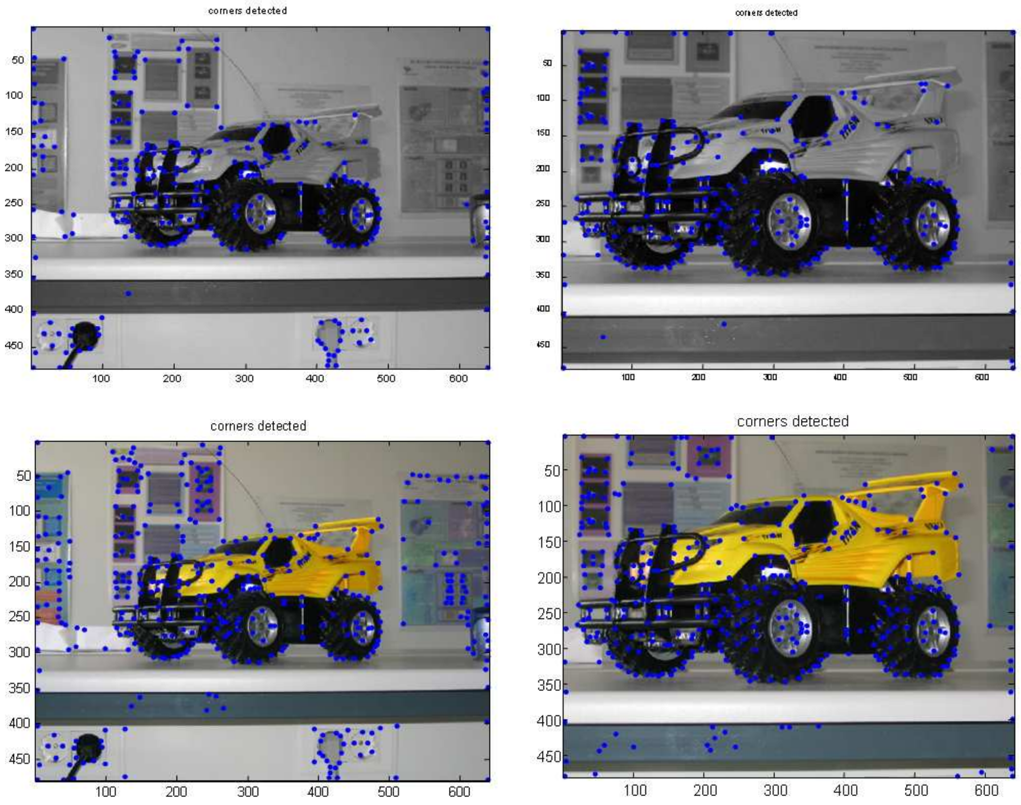


Figure B.6: Comparison for grey level and colour Harris detector, for two different scales. Parameters: $\sigma = 3$ and Low-contrast threshold $= 1000$

B.2.2 Multi-scale Harris Detector

Harris corner detector is very sensitive to scale changes, because of that it is not useful to establish correspondences between images of different sizes. By generating a scale space similar to the Lowe method we can obtain scale invariance [Mikolajczyk and Schmid, 2001]. After generating the smoothed images for each scale, a conventional Harris detector is applied for each one, but only those larger than the neighbours in the scales above and below are selected.

Results for a initial variance $\sigma = 1.6$ and a interval between sigmas $k = 2^{1/4}$ and a space between scales of 6 images are shown in the Figure B.7.

A small step between scales is necessary to obtain good results but it increases the computational cost. Furthermore, it produces duplicity of keypoints slightly moved. We can solve this problem increasing the step between sigmas (some keypoints are lost) or using a lower range of scales (scale invariance is lost) as it can be seen in Figure B.8. Another solution is to implement a Harris detector invariant to affine transformations [Mikolajczyk and Schmid, 2002; Darrell, 2005].

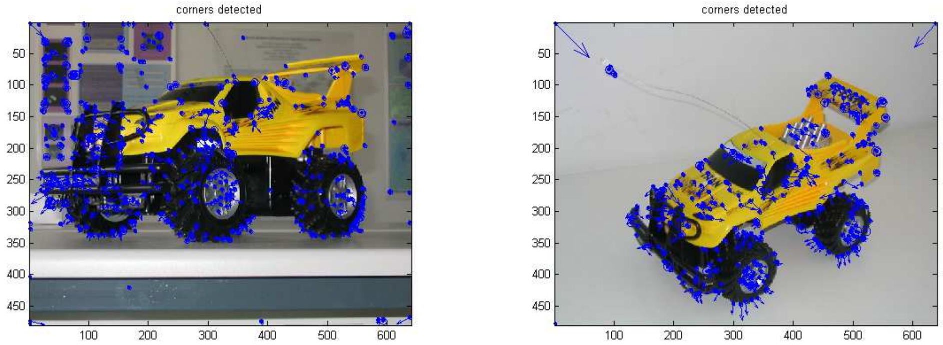


Figure B.7: Keypoints obtained. Arrows show angle and module of the corner and the radius of the ellipse show the scale.

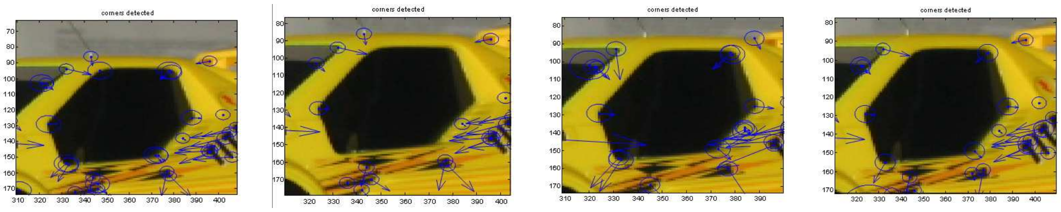


Figure B.8: Scale-space parameter evaluation. a) $k = 2^{1/4}$ and 6 scales, b) $k = 2^{1/4}$ and 2 scales, c) $k = 2^{1/3}$ and 6 scales, d) $k = 2^{1/2}$ and 6 scales

B.3 Conclusions

Harris and Lowe detector have shown similar results. Although Lowe detector gives an integrated framework to extract keypoints and identify them over time and resolution, Harris can be modified, as we have seen, to incorporate the advantages that Lowe detector has (such as multi-scale), being a little bit more coherent over time, and placing keypoints nearer the real locations.

Nevertheless, noone is consistent when big changes of the viewpoints appear. Finally, their application to human tracking do not produce good results. This is due to the fact that human are intrinsically deformable objects without corners. Even their application to characteristic region with characteristic points like faces are not consistent enough over time: changes in the expression of the face change drastically the location and number of keypoints.

C

GMM Techniques and Validation Indexes

In this appendix, we present two conventional GMM methodologies and a set of indicators or indexes which can be used to tackle one of the most delicate topics in clustering: the number of clusters required to model the characteristic space correctly. Obviously an extensive literature exists related to this topic, and it could entailed a thesis by itself. Nevertheless, we have decided to apply two general approaches although better results could be obtained using other specific or complex techniques more adequate for our particular situation and targets. In this way, no assumptions are made and we keep the independence of the method regarding the target and the shape of its characteristic space.

C.1 Fuzzy C-means

Fuzzy c-means (FCM) is a popular clustering methodology based on the fuzzy logic principles, that is, an element can belong to one or more different clusters with a certain degree of confidence.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a data set of n elements in an s-dimensional Euclidean space R^s and let c be a positive integer larger than one. A partition of X into c clusters can be presented using mutually disjoint sets $X_1, X2, \ldots, X_c$ such as $X = X_1 \cup X_2 \cup \ldots \cup X_c$. The membership of an element to each set is given by $\mu = \{\mu_1, \mu_2, \ldots, \mu_c\}$. μ_i is the membership function such that $\mu_i(x) = 1$ if $x \in X_i$ and $\mu_i(x) = 0$ if $x \notin X_i$. Since we have applied a fuzzy clustering algorithm, an element can be part of several clusters, so $\mu_i(x) \in [0, 1]$ and $\sum_{i=1}^{c} \mu_i(x) = 1$.

FCM is an iterative clustering algorithm based on minimising the function J_{FCM}

$$J_{FCM}(\mu, a) = \sum_{i=1}^{c} \sum_{j=1}^{n} \mu_{ij}^{m} \|x_j - a_i\|^2 \tag{C.1}$$

where $a = \{a_1, a_2, \ldots, a_c\}$ is the set of c cluster centres and $m > 1$ is a parameter called fuzzifier which adjusts the clustering performance of FCM [Yang and Wu, 2006].

$$\mu_{ij} = \left(\sum_{k=1}^{c} \frac{\|x_j - a_i\|^{\frac{2}{m-1}}}{\|x_j - a_k\|^{\frac{2}{m-1}}} \right)^{-1}, \qquad a_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{n} \cdot x_j}{\sum_{j=1}^{n} \mu_{ij}^{n}} \tag{C.2}$$

being $i = 1, \ldots, c$ and $j = 1, \ldots, n$.

As output, FCM gives a set of Gaussians defined by their means a, variances σ and weights w. However, like in conventional clustering, it is necessary to pre-assume the number c of clusters algorithms, which is in general un-known and difficult to estimate.

The problem for finding an optimal number of clusters c is usually called cluster validity. Most CVIs are defined by combining the evaluation criteria of compactness and separability. The first one measures the closeness of cluster elements. The second one indicates how distinct two clusters are. It is usually computed between two clusters. Thus, the aim of CVIs is to obtain small intra cluster distances and large inter-cluster distances.

C.1.1 PC & PE

The first proposed cluster validity functions are the partition coefficient PC and partition entropy PE [Yang and Wu, 2006]. Both indexes are defined as follows:

a)

$$PC(c) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{2} \tag{C.3}$$

where $1/c \leq PC(c) \leq 1$. The optimal cluster number c^* is found by solving $max\{PC(c)\}_{2\leq c\leq n-1}$

b)

$$PE(c) = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{2}\log_2\mu_{ij} \tag{C.4}$$

where $0 \leq PE(c) \leq \log_2 c$. The optimal cluster number c^* is found by solving $min\{PE(c)\}_{2\leq c\leq n-1}$

However, these indexes have the disadvantage of lack of connection with the geometrical structure of the data due to the fact that they use only membership functions.

C.1.2 XB & FS

Other validity indexes which take into account explicitly the geometrical properties are the XB and FS [Yang and Wu, 2006].

c)

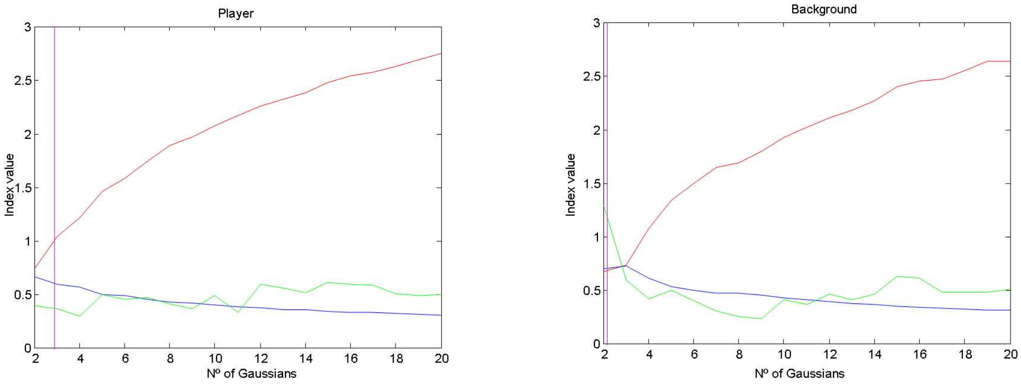
$$FS(c) = \sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}\|x_j - a_i\|^2 - \sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}\|a_i - \bar{a}\|^2 \tag{C.5}$$

with $\bar{a} = \frac{1}{n}\sum_{i=1}^{c} a_i$ where the first term measures the compactness and the second one the separation between clusters. The optimal cluster number c^* is found by solving $min\{FS(c)\}_{2\leq c\leq n-1}$

d)

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}\|x_j - a_i\|^2}{n\cdot\min\|a_i - a_j\|^2{}_{i,j}} \tag{C.6}$$

where the numerator is a compactness measurement and the denominator is a separation measurement. The optimal cluster number c^* is found by solving $max\{XB(c)\}_{2\leq c\leq n-1}$



a) b)

Figure C.1: Optimum number of clusters for placer and background. Green=XB, Magenta=FS, Blue=PC, Red=PE. a) Foreground, b) Background

The best way to obtain the optimum number c consists in combining indexes in pairs (PE-PC, XB-FS). The optimum c^* will be the point in which both curves cross. Both pairs of indexes have been tested for football sequences.

The segmentation images obtained are shown in Figures C.2 and C.3 using the optimum number of clusters depicted in Figure C.1.

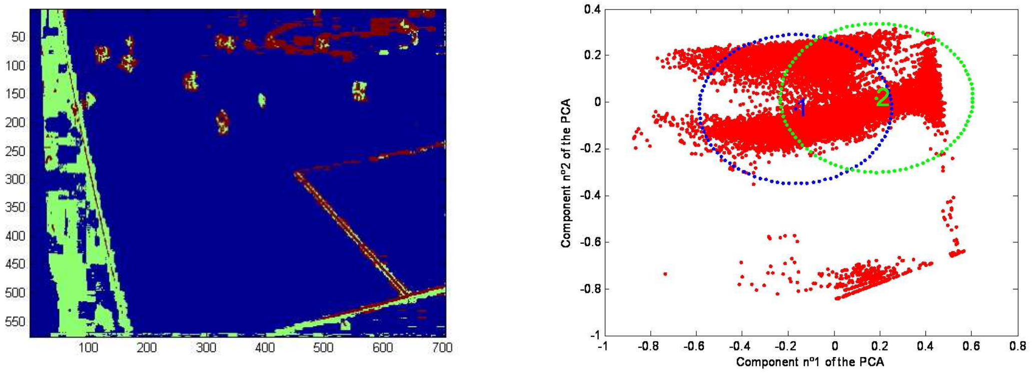


Figure C.2: FCM using PE and PC validation functions. a) Clustering, b) Gaussians

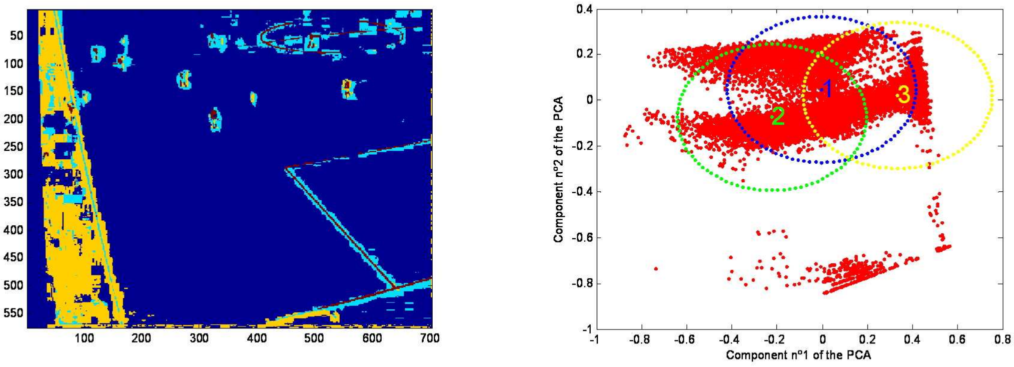
C.1.3 SD and v_{sv}

Other tested indexes are the couple SD ($Scat\&Distance$)

$$Scat(c) = \frac{1}{c}\sum_{i=1}^{c}\frac{\|\sigma(a_i)\|}{\|\sigma(\bar{x})\|} \text{ with } \sigma(a_i) = \frac{1}{n}\sum_{k=1}^{n}(x_k - a_i)^2$$

$$Dis(c) = \frac{max\{\delta(c_i,c_j)\}_{i\neq j}}{min\{\delta(c_i,c_j)\}_{i\neq j}}\sum_{i=1}^{c}\left(\sum_{j=1}^{c}\delta(c_i,c_j)\right)^{-1} \text{ where } \delta \text{ is the Mahalanobis distance.}$$

e) $SD(c) = Dist(c_{max}) \cdot Scat(c) + Dist(c)$



a) b)

Figure C.3: FCM using FS and XB validation functions. a) Clustering, b) Gaussians

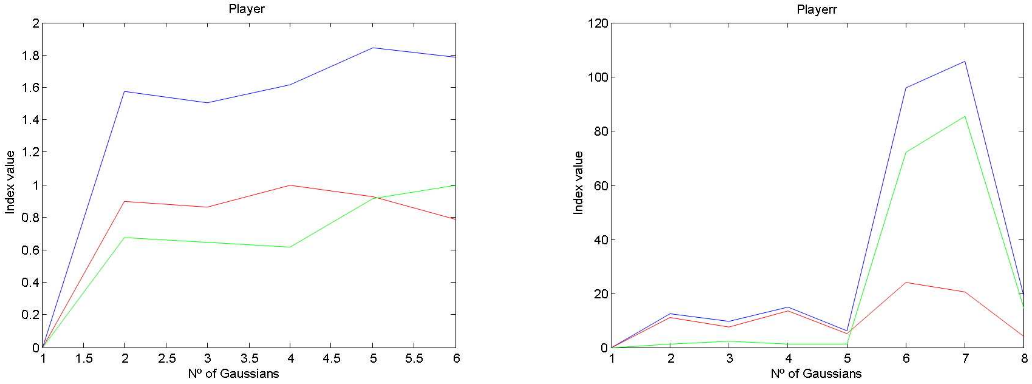
and v_{sv} [Kim and Ramakrishna, 2005].

$v_u(c) = \frac{1}{c}\sum_{i=1}^{c}\left(\frac{1}{n_i}\sum_{x\in X_i}\delta(x,c_i)\right)^{-1}$.

$v_o(c) = \frac{c}{min\{\delta(c_i,c_j)\}_{i\neq j}}$.

f) $v_sv(c) = v'_u(c) + v'_o(c)$ where $v'_u(c), v'_o(c)$ are min-max normalised versions of $v_u(c), v_o(c)$.

Results are shown in Figures C.4 and C.5.



a) b)

Figure C.4: Optimum number of clusters for players a) v_{sv}(blue), v_u(red), v_o(g): 3 Gaussians b) SD(blue), $Dis(length(Dis)) * Scat$ (red), Dis(green): 5 Gaussians

C.2 SOM Clustering

The previous results show non coherent results since each pair of indexes gives us a different number of clusters. This is due to the fact that colour spaces are not easy to cluster because of

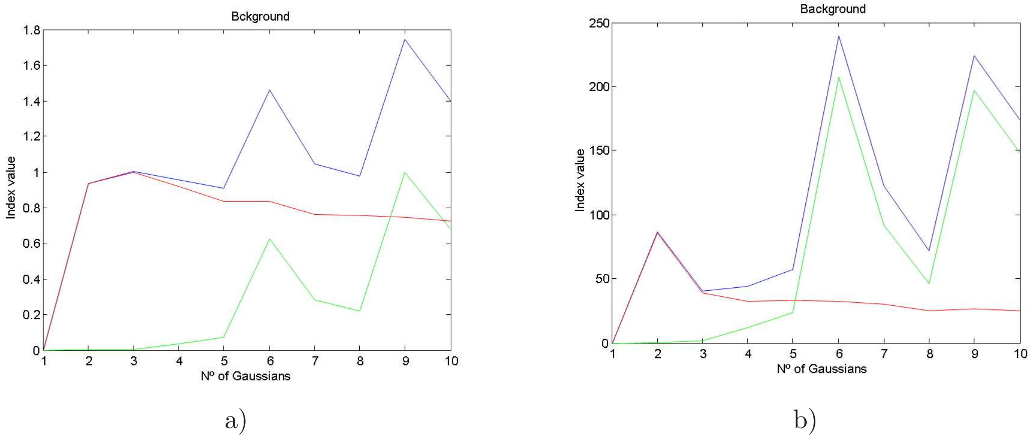


Figure C.5: Optimum number of clusters for background a) v_{sv}(blue), v_u(red), v_o(g): 5 Gaussians b) SD(blue), $Dis(length(Dis)) * Scat$ (red), Dis(green): 3 Gaussians

its continuity and noisy samples. Another trial which can be implemented is to cluster using a Self-Organizing Map combined with a CVI [S. Wu, 2004].

The basic SOM consists of a set of neurons with neighbourhood relations between them. These relations are learned during the training, and after that the SOM is able to divide the input space into regions with the nearest feature vectors. Furthermore, SOM presents the property of topology preservation, which is suitable for visualization and clustering purposes.

The basic SOM is an iterative algorithm. Each neuron i has a d-dimensional feature vector w_i. During the training, a data vector $x(t)$ is chosen randomly at each time step t. Those neuron with the closest vector wi to $x(t)$ is the wining neuron and is denoted by c.

$$c = argmin\|x(t) - w_i\|_i, \qquad i \in [1, 2, \dots M] \tag{C.7}$$

The neighbourhood kernel function $h_{ic}(t)$ is a decreasing function on time and on distance to c, which affects to the neighbouring nodes and supplies the topology preservation to the map.

$$h_{ic}(t) = \exp - \frac{\|Pos_i - Pos_c\|^2}{2\sigma^2(t)}, \qquad i \in N_c \tag{C.8}$$

where Pos_i are the coordinates of neuron I on the output grid and $\sigma(t)$ is the kernel with which defines the neighbourhood.

Finally, the weight update rule can be written as

$$w_i(t+1) = \begin{cases} w_i(t) + \varepsilon(t) \cdot h_{ic}(t)\,(x(t) - w_i(t)) & \forall i \in N_c, \\ w_i(t) & otherwise. \end{cases} \tag{C.9}$$

where $\varepsilon(t)$ is the learning rate. Both $\varepsilon(t)$ and $\sigma(t)$ decrease over time.

As clustering validity index, we have used the composing density between and with clusters CD_{bw} [S. Wu, 2004]. Its calculation can be written as follows:

$$Intra\_den(c) = \frac{1}{c} \sum_{i=1}^{c} \sum_{j=1}^{r_i} \sum_{l=1}^{n_i} f(x_l, v_{ij}) \tag{C.10}$$

where x_l belongs to i_{th} cluster, v_{ij} is the j_{th} representation point of the i_{th} cluster and $f(x_l, v_{ij})$ is defined by

$$f(x_l, v_{ij} = \begin{cases} 1 & \|x_l - v_{ij}\| \leq stdev \\ 0 & otherwise. \end{cases} \tag{C.11}$$

with $stdev = \sqrt{\sum_{i=1}^{c} \|stdev(i)\|^2/c}$ and $stdev(i) = \sqrt{\sum_{k=1}^{n_i}(x_k - a_k)|^2/(n_i - 1)}$.

$$Inter\_den(c) = \frac{1}{c}\sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \frac{\|min\|v_{ik} - vil\|_{kl,k\neq l} - min\|v_{jk} - vjl\|_{kl,k\neq l}\|}{\|stdev(i)\| + \|stdev(j)\|} \tag{C.12}$$

$$\times \sum_{r=1}^{n_i+n_j} f\left(x_r, \frac{min\|v_{ik} - v_{il}\|_{kl,k\neq l} + min\|v_{jk} - v_{jl}\|_{kl,k\neq l}}{2}\right) \tag{C.13}$$

with

$$f(x_l, u_{ij} = \begin{cases} 1 & \|x_l - u_{ij}\| \leq \left(\|stdev(i)\| + \|stdev(j)\|\right)/2 \\ 0 & otherwise. \end{cases} \tag{C.14}$$

However, the intercluster distance is significantly higher than inter-cluster one. So,

$$Sep(c) = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \frac{\|min\|v_{ik} - vil\|_{kl,k\neq l} - min\|v_{jk} - vjl\|_{kl,k\neq l}\|}{1 + Inter\_den(c)} \tag{C.15}$$

and the validation index is

$$CD_{bw}(c) = Intra\_den(c) \cdot Sep(c) \tag{C.16}$$

Results obtained with this clustering algorithm and its CVI can be seen in Figure C.6, where the optimum number of cluster obtained was 5 clusters for foreground.

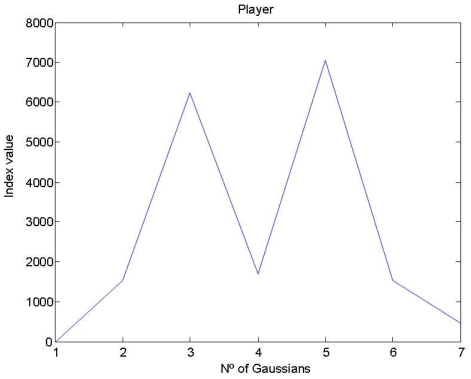


Figure C.6: Clustering using a SOM

D

Multi-sensor Tracking System

Multi-sensorial systems are common in other fields of knowledge, like robotics, but more unusual in computer vision, where they are only used when the difficulty of the problem makes impossible to solve it in another way. However, the usage of intrusive sensors can be useful in specific application or fields, in which the importance of robustness or accuracy takes priority over the cost or limitations that they implies.

In this section, we focus on the integration of vision sensors (colour cameras) and ultrasound sensors for tracking in indoor environments.

D.1 State of the Art

There is an extensive literature about sensor conjugation, specially in robotics. However, the philosophy of these works differs from our objectives: while robotics tries to auto-locate the robot and map the environment, our research concentrates on object and human tracking with external sensors. An example of the first case is [Wallner et al., 1995], where a 3D scene reconstruction is made using sonar sensors and stereo vision. Both sensors have complementary error in depth and angular resolution.

More interesting are those papers which employ a tracking algorithm to track an object over time and conjugate the observation from different sensors. Two different strategies are followed: the first one [Brooks and Williams, 2003; Amit S. Chhetri and Papandreou-Suppapppola, 2003; Doucet et al., 2002; Chhetri et al., 2004] consists in selecting the most adequate sensor for each time step; the second one [Kogut and Trivedi, 2002; Michalowski and Simmons, 2006; Girija et al., 2000; Perez et al., 2004; Punska, 1999; Crowley and Demazeau, 1993] combines all the observation simultaneously. On the one hand, the first alternative allows a longer life of the sensors, a bandwidth reduction and a lower complexity. On the other hand, the second approach provides better results by introducing complementary information, which increases the accuracy and robustness. Our proposal can be framed in this second group.

Another possible division of the papers can be done attending to the scheduling algorithm. Kalman filter (and its variants) and particle filter stand out. The first one [Brooks and Williams, 2003; Kogut and Trivedi, 2002; Michalowski and Simmons, 2006; Girija et al., 2000; Varshney, 2008; Crowley and Demazeau, 1993] allows dicretising the observations coming from the different sensors, dealing with them in a more independent way. On the contrary, particle filter [Amit S. Chhetri and Papandreou-Suppapppola, 2003; Doucet et al., 2002; Chhetri et al., 2004; Perez et al., 2004; Punska, 1999] is more suitable for combining measurements from the same kind of sensor, or sensors whose outputs can be expressed in the same format. In both cases, a shared reference is required to integrate the results.

Many different types of sensor have been combined: laser [Brooks and Williams, 2003; Michalowski and Simmons, 2006], inertial sensor [Girija et al., 2000], radar [Amit S. Chhetri and Papandreou-Suppapppola, 2003; Girija et al., 2000; Chhetri et al., 2004], GPS [Girija et al., 2000], infrared [Amit S. Chhetri and Papandreou-Suppapppola, 2003; Chhetri et al., 2004] or stereophonic sound [Perez et al., 2004].

A good survey can be found in [Wagner, 2007], where an exhaustive analysis of mobile or static sensors is introduced. Different multi-sensorial fusion methodologies are explained, being grouped into complementary, competitive or cooperative sensor networks. Finally, error models are generated to characterise the accuracy of the tracker and obtain a better conjugation.

As conclusion, we can assert that the more complementary the sensors are, the better result is obtained. In addition, temporal or spatial redundance helps to improve the results for each time step.

D.2 Proposal

Ultrasound sensors provide discrete data without a measurement of their reliability. Spurious data and rebounds appears occasionally. The accuracy of the observation is the same in all the area covered by the sensors. Moreover, the observation has a label with the identity of the target. That is, the association problem is solved for these sensors, being unnecessary a matching algorithm. A simple scheduling algorithm, like Kalman, would be enough for tracking a target observed with these sensors. However, the frequency of observation is quite poor: around 1 or 2 per second.

On the contrary, cameras provide a much richer information but, at the same time, with more noise and more difficulty to be understood. Using a calibration, three different kinds of measurements can be obtained: motion, colour and height difference (see Chapter 5, Section 5.4.2.2).

Our proposal consists in a tracking system which combines ultrasound sensors and cameras on a shared reference. For that, we have a plan of the building where the sensors are installed. Once all the observation have been projected, they are dicretised and an error is assigned to each one in order to fuse them correctly. The fusion algorithm is based on MCUKF.

D.2.1 Vision System

We have applied the same algorithm that we employed in ASTRO project (Chapter 5, Section 5.4.2). It is able to work with multiple cameras and multiple targets and the chosen methodology allows a simple integration of multiple sensor, even belonging to different types, on a common reference. The general scheme of the vision part is shown in Figure D.1. Observation results are depicted in Figure D.2.

A previous calibration has been done, in order to establish a correspondence between the image and the reference, as it can be seen in Figure D.3.

D.2.1.1 Example

The vision algorithm has been tested in the indoor environment, to check its suitability under these conditions. Monocular sequences have been recorded with a framerate of 24fps. Acceptable results have been obtained (see Figures D.4, D.5 D.6 and D.7) in spite of the strong

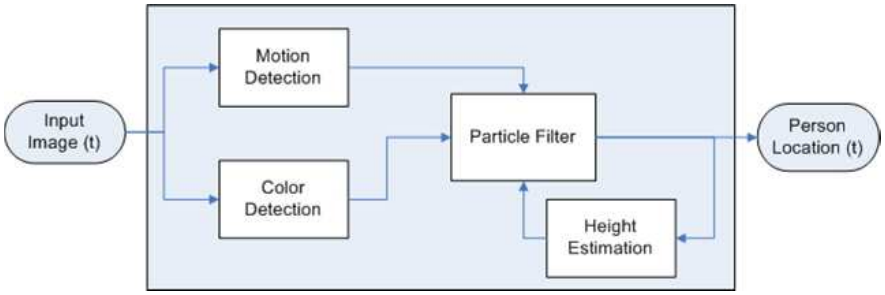


Figure D.1: General scheme of the vision system.

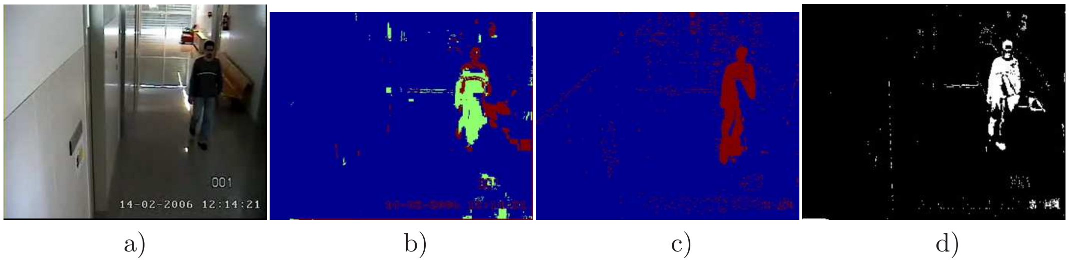


Figure D.2: Camera measurements: a) Current frame, b)Colour PDI, c)Motion image d)Motion image without shadows.

shadows and reflections. However, the perspective effect produces errors in the position of the person on the plan. To reduce this effect, additional cameras or sensors are necessary.

D.2.2 Ultrasound System

Ultrasound system is composed of 6 sensors which define a covering area of 6x3 meters. Sensors receive a signal from a emitter card with which the mobile object is equipped. Emitter cards have a different label in order to track more than one object simultaneously, although it is also possible that a unique target can be equipped with more than one card to reduce the error.

The tracking system returns the label that we are tracking, the time step and the X,Y,Z coordinates, besides the distance to each one of the sensors (see Table D.2 and Figure D.8).

Since the card is held by the subject in the experimental sequences, that is, slightly away from the centre of gravity, the estimated error of the card can be assumed around 10 centimeters.

As main drawback, the capture system is only able to catch one observation per second. This fact limits the utility of the system, by itself, because the target can be lost depending on the speed of the target.

Table D.1: Results Indoor monocular sequences.

	Duration [sec]	Length [frames]	Mean Error [meters]
Seq. 1	10.45	250	0.58
Seq. 2	6.67	160	0.49

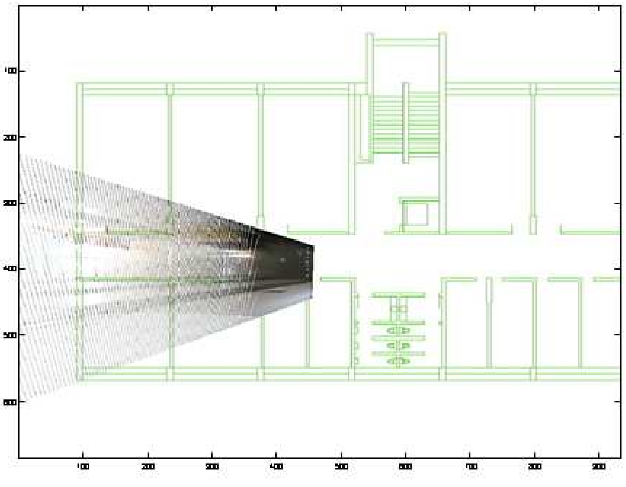


Figure D.3: Projection of the image on the plan.

Table D.2: Ultrasound measurements

Time	Label	X	Y	Z	Distance to the 1st sensor	. . .
04:58:12.867	T20	2,188129	-0,1670129	0,5149519	90,35079	...
04:58:13.117	Ber5	2,315609	0,3080424	0,582642	43,49703	...
04:58:14.749	T20	2,228202	0,04642275	0,5016984	90,67402	...
04:58:14.990	Ber5	2,251367	0,313927	0,6588765	43,23935	...
04:58:15.901	T20	2,234647	0,05643392	0,4958158	90,72294	...
04:58:16.141	Ber5	2,02399	0,4177306	0,6359055	1,069992	...
04:58:17.133	T20	2,233124	0,04100054	0,493125	33,96745	...
04:58:17.373	Ber5	2,118911	0,5050535	0,8658688	3,379262	...
04:58:18.305	T20	2,23312	0,04911823	0,5056254	90,74005	...
04:58:18.565	Ber5	1,883703	1,012415	1,203147	21,86105	...

D.2.3 Integrated System

Once vision and ultrasound measurements have been obtained and projected on the plan, a global estimation is applied using a MCUKF per target on the shared reference. To combine measurements from different sensors, the error or confidence of each one must be modelled. While the error of the vision system is estimated thanks to the dispersion of the particles and the homographic error, the ultrasound sensor error has been obtained with a statistical study.

On the one hand, the accuracy of the ultrasound is between 5 and 10 centimeters, much higher than the vision system. Therefore, we assign a covariance of 13 cm to every ultrasound measurement. On the other hand, there is a considerable number of false positive due to rebounds. However, these distracters are easily discarded thanks to the association algorithm.

Due to the fact that the refresh rate of both system is completely different, being the vision system's framerate higher than the ultrasound one, the whole application should be able to run only using the vision capture system. The ultrasound system will be used, therefore, to reduce the uncertainty every few seconds, correct identity swapping or recover lost trackers.

Other advantages inherent to the combination of sensors are provided. Ultrasound sensors allow an automatic initialisation of the vision system and the automatic extraction of its appearance model. In addition, they also help the system to transfer the subject location from the line of vision of one camera to another. Although the initialisation of a new tracker in a camera is a relatively easy task, the decision about the target going out of the camera is more

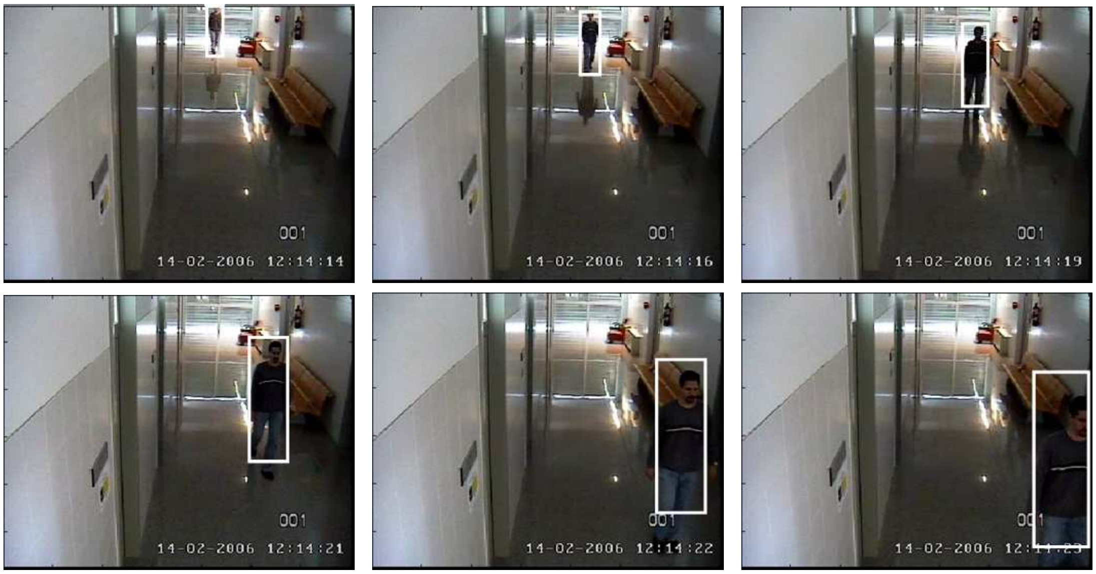


Figure D.4: Sequence 1 qualitative results

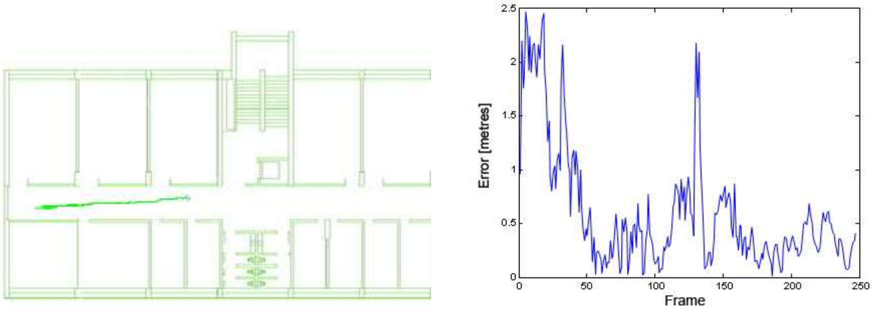


Figure D.5: Sequence 1 quantitative results

complex. Clutter around the border of the image can produce failures and send distracters to the plan.

Since different sensors are provided by different equipments, a synchronisation is required. It has been made with a precision of miliseconds by syncrhronising the internal clocks of the equipments via wifi.

D.3 Results

Preliminary results have been obtained in a sequence with only one person walking. The vision system is composed of two analogical camera and a frame grabber with a rate of 15 fps. The ultrasound system gives one observation per second approximatively, that is, a measurement every 10-15 frames. The available overlap area is a small zone of the indoor environment.

Results are depicted in Figures D.9 and D.10. The third column shows the locations and error of the vision system (blue ellipses) and the ultrasound sensors (black circle), as well as the

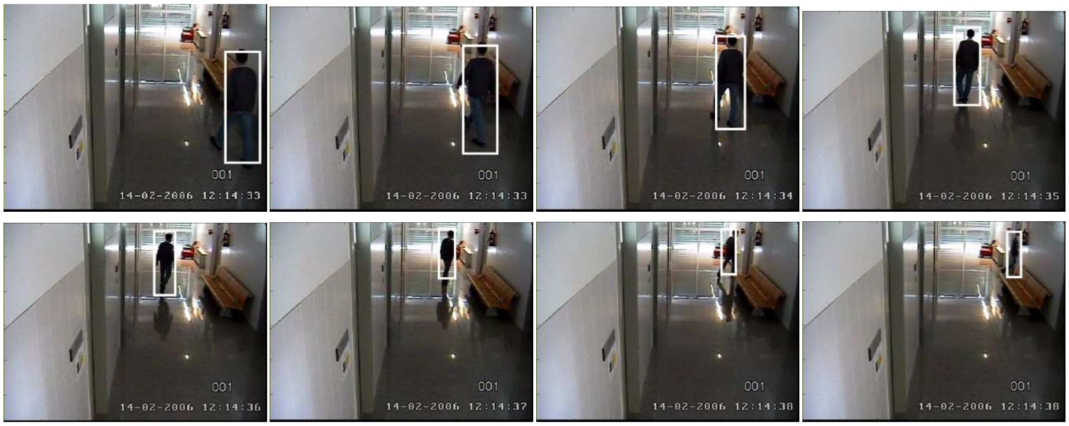


Figure D.6: Sequence 2 qualitative results

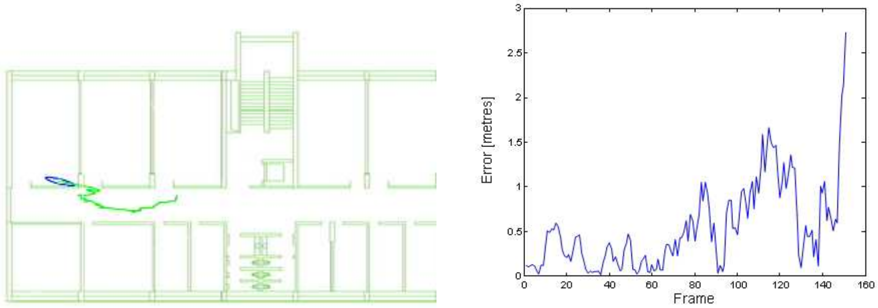


Figure D.7: Sequence 2 quantitative results

global estimation. The comparison between the system with and without ultrasound sensor is shown in Figure D.11.

D.4 Conclusions

The first conclusion that we can extract is the little influence of the ultrasound sensor in the combined system. Due to the poor refresh rate of this kind of sensors, the accuracy is only improved when extra measurement are received. If the vision system fails, the additional sensors are not able to correct large divergences. This is due to the fact that far observations from the ultrasound system are discarded by the association algorithm in order to avoid failures when rebounds and spurious appear. To obtain better results, a higher refresh rate should be achieved. Otherwise, the use of additional sensors is almost anecdotal.

Nevertheless, the usage of this kind of sensor is useful to recover from label swapping after occlusion. In spite of the fact that the additional ultrasound system does not improve the tracking during the occlusion, it will be able to detect the change of identities once the occlusion has finished. This advantage has a drawback: an intrusive card containing the label of the target must be used.

Moreover, we think that the proposal is valid for multi-sensor integration when the refresh rate of the sensors is more similar. The usage of a different kind of sensors, or the improvement

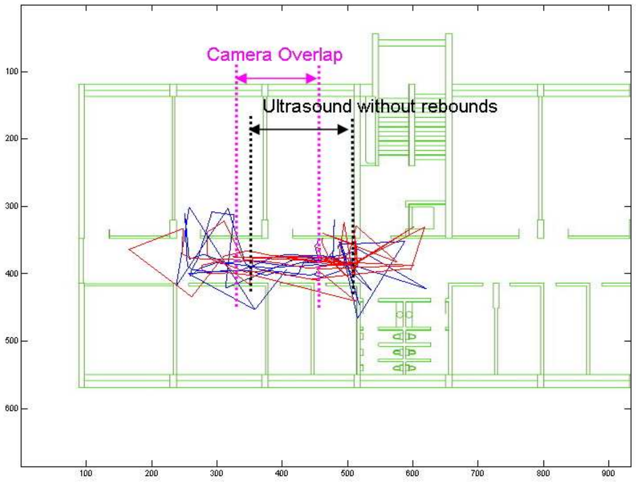


Figure D.8: Ultrasound trajectories and overlap areas with the vision system.

of the ultrasound capture system, could give us a robust and accurate application.

Regarding the vision system, the indoor environment where the algorithm has been tested presents bad conditions for a successful tracking: bad illumination, reflectance, shadows, poor colour images, etc. However the tracking algorithm is able to track a target under these condition, although complex interactions or occlusion could produce a failure.

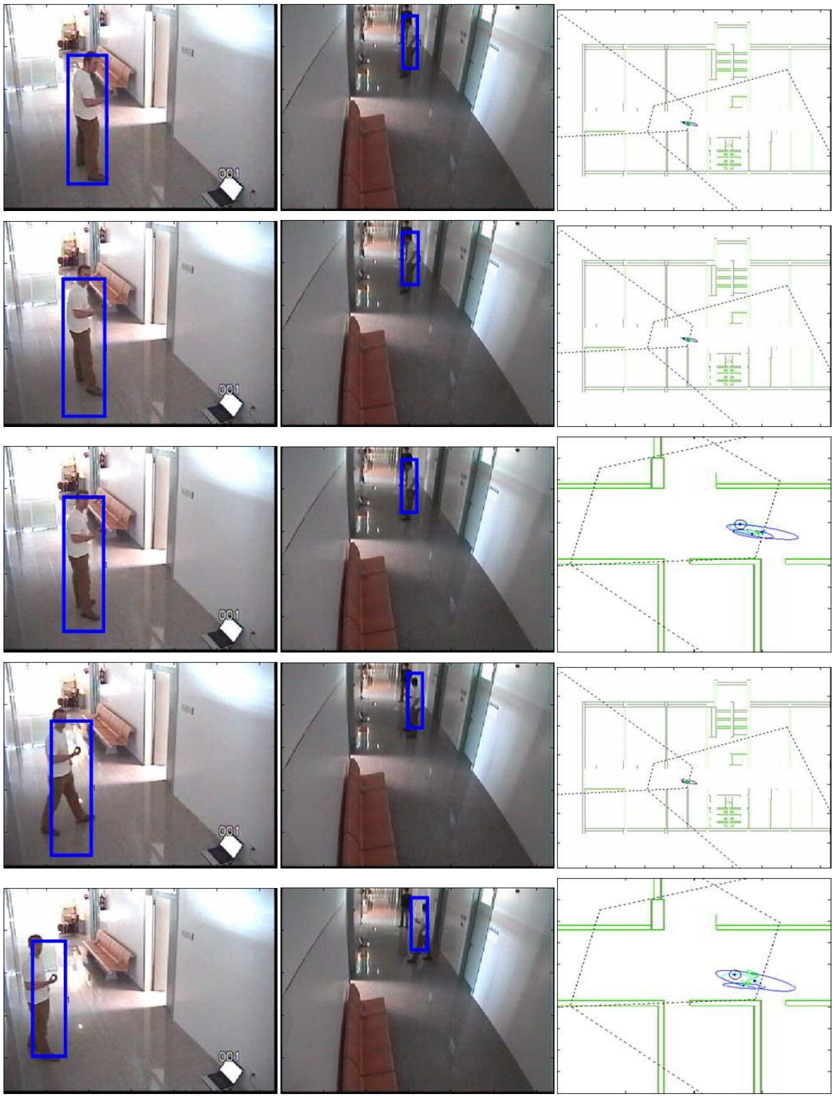


Figure D.9: Results obtained using the multi-sensor system.

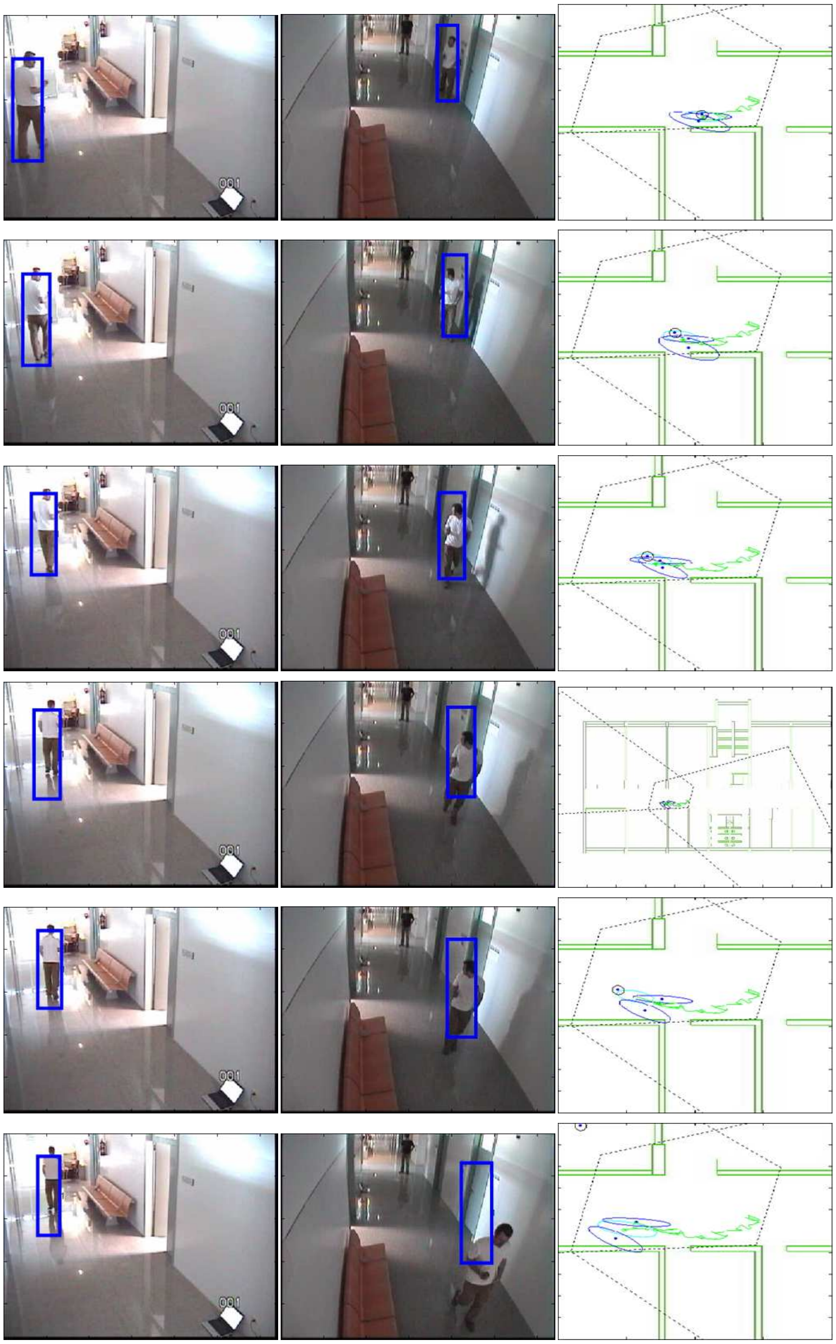


Figure D.10: Results obtained using the multi-sensor system.

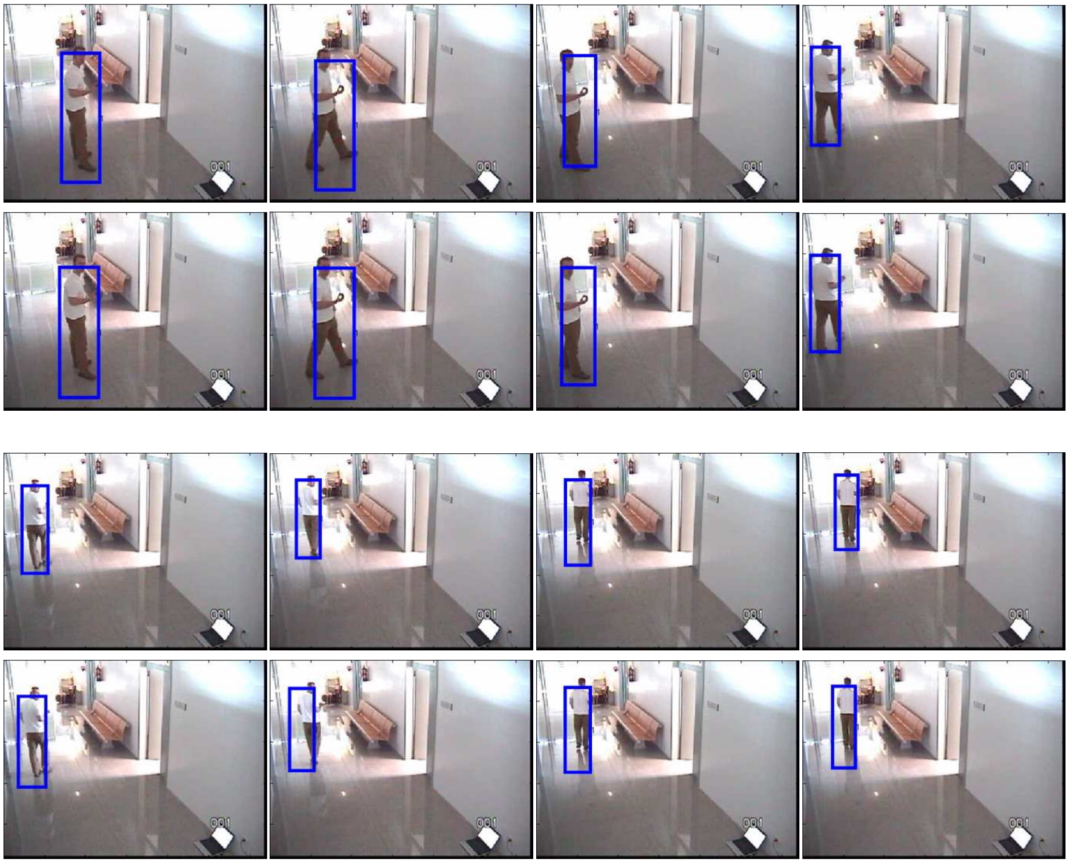


Figure D.11: Comparison between the vision system (Second row) and integrated system (First row).

E

ASTRO Data Association Algorithm

The goal of this stage consists in selecting the measurements for each tracker. This is the only stage that can not be computed sequentially or independently for each tracker. The proposed algorithm is based on the theory of data association presented in [Bar-Shalom and Li, 1993]. In this manner, we have chosen a modified version of the nearest neighbour algorithm [Zhang et al., 1996]. The main difference with respect to the original algorithm consists in a simple approach which limits the number of combinations in order to reduce the computing time, although obtaining a sub-optimal result.

In our algorithm, two conditions must be fulfilled for assigning the measurements: not more than one measurement of each camera can be assigned to each tracker, and a measurement can not be assigned to two different trackers. In other words, an algorithm to solve the conflicts must be applied. A set of possible measurements is assigned to each tracker using as criterion the Mahalanobis distance between trackers and measurements. Mahalanobis distance is a metric calculation based on correlations between variables, which allows us to obtain the similarity between these variables, taking into account the distribution of samples. The equation of Mahalanobis distance is as follows:

$$D = \sqrt{(x-\bar{x})^T C^{-1} (x-\bar{x})} \tag{E.1}$$

where $\bar{x}$ is the mean and C the covariance matrix.

With an appropriate threshold, it is possible to know if there is any measurement suitable to a tracker for each prediction. We consider all the measurements which are placed in a zone corresponding to the 95% around the mean, that is, all the measurements whose distances will be lower than 5.99 (chi-square test).

Later, a matrix of possibilities Π is composed. If each tracker i has a number m_i of possible measurements $M_1^1, M_2^1, \ldots, M_{m_i}^1$, the first row of the matrix is initialised with the measurements of the first tracker.

$$\mathbf{\Pi}_1 = \left[M_1^1 \; M_2^1 \ldots M_{m_1}^1 \right] \tag{E.2}$$

In the second iteration, the matrix Π_1 is replicated m_2 times, and a new row with the measurements assigned to the second tracker is appended to obtain all possible combinations:

$$\begin{aligned} \mathbf{\Pi}_2 = \left[\begin{array}{cccc} M_1^1 \; M_2^1 \ldots M_{m_1}^1 & M_1^1 \; M_2^1 \ldots M_{m_1}^1 & \ldots & M_1^1 \; M_2^1 \ldots M_{m_1}^1 \\ M_1^2 \; M_1^2 \ldots M_1^2 & M_2^2 \; M_2^2 \ldots M_2^2 & \ldots & M_{m_2}^2 \; M_{M_2}^2 \ldots M_{m_2}^2 \end{array} \right] = \\ = \left[\begin{array}{cccc} \mathbf{\Pi}_1 & \mathbf{\Pi}_1 & \ldots & \mathbf{\Pi}_1 \\ \mathbf{M}_1^2 & \mathbf{M}_2^2 & \ldots & \mathbf{M}_{m_2}^2 \end{array} \right] \end{aligned} \tag{E.3}$$

Before starting each iteration, non-compatible combinations must be erased, because two trackers can not catch the same measurement. Thus, the matrix Π'_2 is obtained. The process continues in the same way until finishing all the trackers.

$$\mathbf{\Pi}_i = \begin{bmatrix} \mathbf{\Pi'}_{i-1} & \mathbf{\Pi'}_{i-1} & \dots & \mathbf{\Pi'}_{i-1} \\ \mathbf{M}_1^i & \mathbf{M}_2^i & \dots & \mathbf{M}_{m_i}^i \end{bmatrix} \tag{E.4}$$

After applying the algorithm, the resulting matrix Π has a number of rows equal to the number of trackers, and a number of columns equal to the number of valid measurement-tracker combinations. Thus, the combination whose sum of distances will be minimum is selected. A graphic example is depicted in Figure E.1.

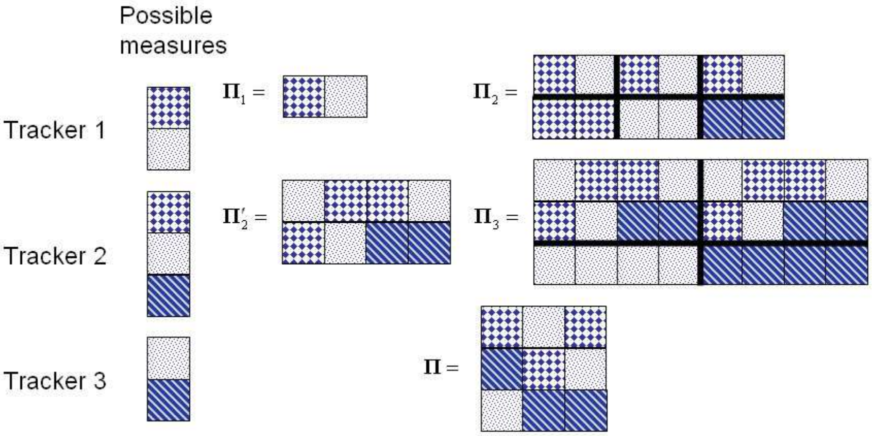


Figure E.1: Diagram with a simple example of how different combinations are generated and impossible combinations are filtered.

However, there are some important points that we must take into account:

- We can choose between two strategies for doing the combination filter: wait for the last iteration, or apply the filter on each iteration. The first option is easier, but the second is more efficient.
- If it is possible, the matching should be calculated independently for each camera, in order to reduce the number of options.
- The possibility of not having a measurement must always be considered for each tracker. If an incorrect measurement is assigned, it will affect and damage the other trackers.

E.1 Problems of a Multisensor Multitarget Matching

There are several problems using multisensor matching. Most of these problems only happen if there has been a detection or segmentation error previously, but solutions must be designed to avoid these errors causing a bad tracking. We have considered the following errors:

- One tracker associated to measurements from different players.
- Several trackers associated to measurements from the same player.
- Large covariances.

- Unknown number of possible combinations.

The objective of these corrections is to ensure that each tracker is tracking a player. It is not possible to ensure that in a determined moment each tracker will be tracking one and only one player. But the system must be capable of noticing any important error and be able to restore the desired situation in a relatively brief period of time.

E.1.1 Problem with one tracker associated to measurements from different players

This problem only happens when each sensor just covers a part of the tracking field, being possible that two or more players are nearby but not all are seen from a certain camera. In this case, when a tracker has a large covariance (for example, when it has lost the player it was tracking) it is possible that it takes both measurements as if they have come from the same player.

If the covariance is large, then it is also possible that these measurements from different cameras are distant, indicating that they come probably from different players. The detection of the problem consists then in establishing a threshold that represents the maximum distance between two measurements if they are provided by the same player.

The solution applied is different if the tracker captures two measurements or more. In case the tracker has two measurements assigned, the solution is to unassign the furthest measurement from the tracker. After that, it is advisable to check if the measurement may correspond to another tracked player.

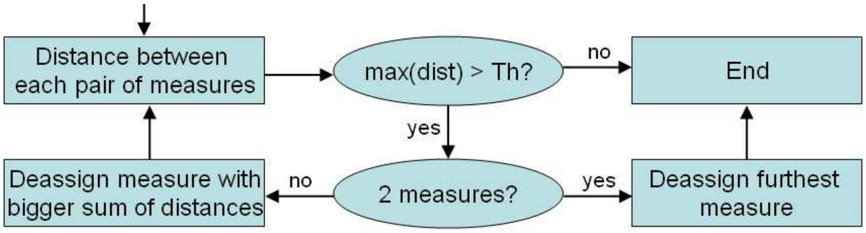

Distance between each pair of measures
max(dist) > Th?
no
End
yes
Deassign measure with bigger sum of distances
no
2 measures?
yes
Deassign furthest measure

Figure E.2: Diagram used for discarding measurements when they are too far away. Each unassigned measurement is reassigned if possible.

In the case of three or more measurements, an iterative process is performed until the distance between each pair of trajectories is below the threshold. First, the measurement with the highest global distance to the other measurements is unassigned. Then, this measurement is assigned to other tracker if possible. Finally, all distances between each pair of measurements are evaluated. This process is repeated until the maximum distance is below the threshold, as seen in Figure E.2.

E.1.2 Problem with several trackers associated to measurements from the same player

It is also possible that different trackers are associated to different measurements from the same player (Figure E.3.a), effectively following these trackers the same player while there are other available players. Normally, this problem only happens after a segmentation error or under unexpected circumstances. Since it is inevitable to make mistakes in previous stages, it is necessary to develop a system that allows us to correct them.

To be sure that these two or more measurements come from the same player, three conditions must be satisfied. The first condition is that any pair of measurements of the supposed bad trackers can not come from the same camera. The second condition relates to the distance between each pair of measurements which must always be below a certain threshold. The third and final condition is that the trackers must also have a distance below another threshold. To be coherent, this threshold must have the same value as the one used in the solution of the previous problem.

The solution for this problem is to merge all measurements into only one tracker when the three conditions are satisfied, leaving the other trackers without measurement, and establishing what we have called *Exclusion Zones*.

Each Exclusion Zones consists in an area that is established surrounding a measurement, valid for only a tracker and only during a determined number of frames. When a measurement is inside an active Exclusion Zone, it is not visible to the tracker that has created this Exclusion Zone (Figure E.3.b).

Exclusion Zones are indispensable, since when all measurements are assigned to the same tracker, the other trackers make their covariances grow to try to catch other measurements but they will recapture the same measurements if the Exclusion zone is not established. It is convenient that the number of frames that the exclusion zone is active through will be enough to allow the tracker to find another measurement, and that its size will be enough to ensure that the tracked player does not cross through its boundaries while the exclusion zone is active (Figure E.3.c).

E.1.3 Problem with large covariances

Another possible problematic situation is generated by trackers with large covariances. When a tracker looses all measurements, its covariance starts to grow, in order to be able to catch an available measurement. But, since Mahalanobis distance is used to measure the distance among trackers and measurements, it is possible that, if the covariance becomes very large, the tracker captures a measurement that corresponds to a nearer tracker with small covariance (Figure E.3.c).

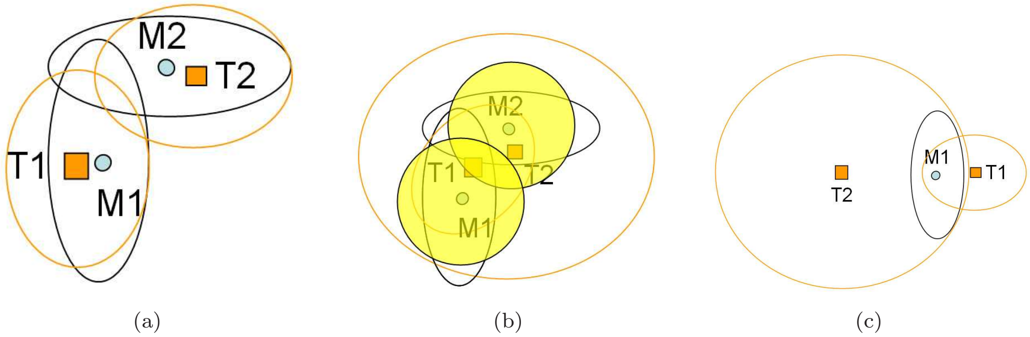


Figure E.3: (a)Tracker T1 is tracking measurement M1, from camera 1, while tracker T2 is tracking measurement M2, from camera 2. But both measurements come from the same player. (b)Both measurements are assigned to tracker T1, while 2 Exclusion zones, one around each measurement, are created for tracker T2, preventing it to catch these measurements while its covariance grows. (c)Being far from measurement M1, tracker T2 gets it because the Mahalanobis distance is lower than for T1, because T2 has a large covariance.

It is possible to limit the maximum covariance area, but this affects the way the trackers recapture lost measurements. Another option is to give priorities when assigning the measurements. One or more thresholds can be set, to divide the trackers in several categories. Trackers with the smallest covariance will choose their measurements first, and the other trackers will only be able to choose among those measurements that the first group of trackers has not caught. It is necessary that any tracker that has not lost its measurements is always included in the group of trackers with the smallest covariance to avoid assignment problems.

E.1.4 Problem with the number of possible combinations

The algorithm previously exposed has the disadvantage that the number of combinations between measurements and trackers is unknown, and it can be very high if each tracker has many different available measurements. It is necessary to limit the number of combinations in order to reduce the number of maximum combinations that the algorithm can take into account.

The easiest method to limit the total number of combinations is to limit individually the number of possible measurements that a tracker can have, discarding further measurements. For example, a threshold of $m = 3$ is a good agreement between speed and efficiency in our case. It is high enough to consider all important options, but not so high that processing time will be too long.

However, there is a better way to limit the number of maximum combinations, consisting in deleting worst partial combinations each iteration, if they exceed our desired limit.

F

Error Analysis

A quality study of system errors has been carried out over a set of test data chosen in one of the cameras installed in the stadium. These data have been classified into three zones in the image, each one with a different mean distance from the camera position. In this way, the perspective effects can be analyse through errors in the overall process. This can be seen in Fig. F.1.

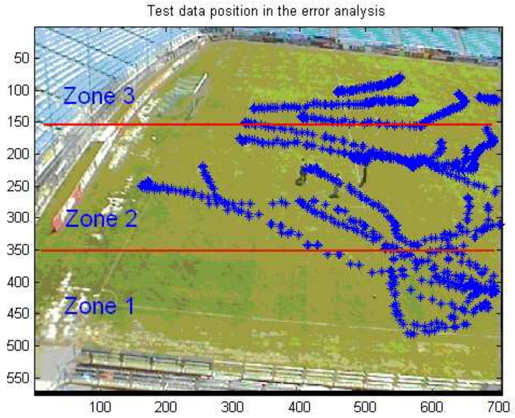


Figure F.1: Perspective zones in the Camera 1.

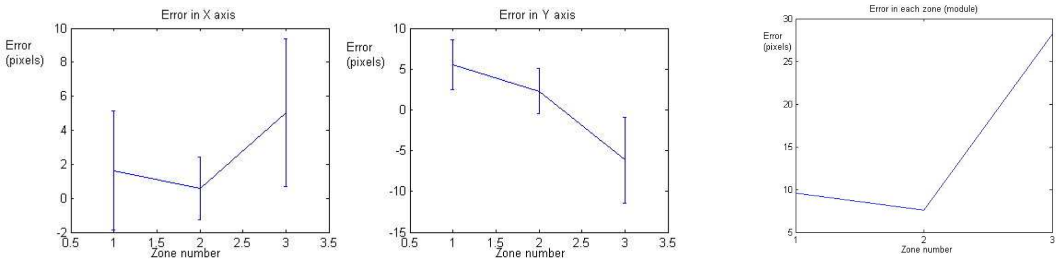
There are two procedures which must be taken into account: measurement extraction (involving Particle filter) and homography. These errors involve the right working of the UKF and its correspondence algorithm.

Simplifying expressions, the time evolution of the UKF is governed by the next equations:

$$\begin{aligned} x_{k+1} &= F \cdot x_k + \omega_E \\ y_k &= H \cdot x_k + \omega_M \end{aligned} \tag{F.1}$$

where $\omega_E \simeq N(0, \sigma_E)$ and $\omega_M \simeq N(0, \sigma_M)$ are the state and measurement errors respectively. It is not possible to know a priori the first one because it depends on the target trajectory. On the other hand, the measurement error can be obtained by analysing the measurement process.

Both errors work together in order to produce the final error in the UKF algorithm, expressed as predicted and estimated covariance (P). Mathematically, this can be expressed



a) b)

Figure F.2: Perspective error in the image. a) Error in both axis, b) Error module.

as $P \approx \omega_E + \omega_M$. The first term can be reduced by adding more complex dynamic models [Senior, 2002] or incorporating different models to take into account different player behaviors [Farmer et al., 2002]. The last term can be covered by studying the measure process and providing better algorithms. In this last case, an alternative proposal has been proposed in this paper. An image tracking system has been incorporated to improve the measurement process reducing thus the measurement noise. Nevertheless, the quality of measurement depends on more parameters like perspective and distortion errors. In our case, the perspective error plays an important role because the players are located at very sparse distances from the camera. This fact is one of the most important reasons to understand the necessity of using several cameras (one camera will always have the best perspective of a player).

Assuming that errors are gaussian with zero mean, two “gaussian” measurements can produce a better observation. This is another reason to use several cameras. However, the hypothesis of null mean is not true. Our measurements possess a positive mean whose value depends on the target position. This characteristic can be observed in Figs. F.2.a and F.2.b. The error in both axis is shown in the first one, and the error module in the last figure. Note that error is lower in the centre of the image, and it rises in the external areas.

Once the problematic of the measurement error has been treated, the homography error can be analysed. This error appears when measurements in the image are converted to plane coordinates. Obviously, this error depends on the player position with respect to the camera, but it appears due to the homography matrix construction, to be more exact, the points selected to build this matrix.

We have followed the methodology cited in [Criminisi et al., 1999] in order to compute the covariance of the estimated homography. All points used to build this matrix are considered, and the implicit error is modelled as a isotropic homogeneous gaussian. Therefore, these errors are defined as $\sigma_x = \sigma_y = \sigma$ for image points and $\Sigma_x = \Sigma_y = \Sigma$ for plane points.

Likewise, the homography covariance is defined as $\Lambda_h = J \cdot S \cdot J$, where $J = -\sum_{i=2}^{9} \frac{u_k \cdot u_k^T}{\lambda_k}$, being u_k the eigenvector of number k of matrix $A^T \cdot A$ and λ_k the corresponding eigenvalue. Finally, S is defined as follows:

$$S = \sum_{i=1}^{n} (a_{2i-1}^T a_{2i-1} f_i^o + a_{2i}^T a_{2i} f_i^e + a_{2i-1}^T a_{2i} f_i^{oe} + a_{2i}^T a_{2i-1} f_i^{eo}) \tag{F.2}$$

with a_i the row i of matrix A. The rest of parameters can be defined writing H in its vectorial form: $H = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_8)$

Therefore:

$$\begin{aligned} f_i^o &= \sigma^2[h_1^2 + h_2^2 - 2X_i(h_1h_7 + h_2h_8)] + 2\Sigma^2(x_ih_7h_9 \\ &+x_iy_ih_7h_8 + y_ih_8h_9) + (\Sigma^2X_i^2 + x_i2\Sigma^2)h_7^2 + (\sigma^2X_i^2 + y_i^2\Sigma^2)h_8^2 + \Sigma^2h_9^2 \end{aligned} \tag{F.3}$$

$$\begin{aligned} f_i^e &= \sigma^2[h_4^2 + h_5^2 - 2Y_i(h_4h_7 + h_5h_8)] + 2\Sigma^2(x_ih_7h_9 \\ &+x_iy_ih_7h_8 + y_ih_8h_9) + (\Sigma^2Y_i^2 + x_i2\Sigma^2)h_7^2 + (\sigma^2Y_i^2 + y_i^2\Sigma^2)h_8^2 + \Sigma^2h_9^2 \end{aligned} \tag{F.4}$$

$$f_i^{oe} = f_i^{eo} = \sigma^2[(h_1 - X_ih_7)(h_4 - Y_ih_7) + (h_2 - X_ih_8)(h_5 - Y_ih_8) \tag{F.5}$$

being (X_i, Y_i) the coordinates on ground and (x_i, y_i) the coordinates in the image.

The typical formula for homographic computing is $x' = Hx$, which is converted to $X = Bh$, where B is the next 3×9 matrix:

$$B = \begin{pmatrix} x^T & 0^T & 0^T \\ 0^T & x^T & 0^T \\ 0^T & 0^T & x^T \end{pmatrix} \tag{F.6}$$

The 3×3 matrix Λ_x is the covariance of the point X in homogeneous world coordinates. The conversion to a 2×2 matrix $(\Lambda_x^{2\times2})$in non-homogeneous coordinates is carried out as follows:

$$\Lambda_x^{2\times2} = \nabla f \Lambda_x \nabla f^T \tag{F.7}$$

where $X = (X, Y, W)^T$ (world coordinates) and ∇f is defined as:

$$\nabla f = 1/W^2 \begin{pmatrix} W & 0 & -X \\ 0 & W & -Y \end{pmatrix} \tag{F.8}$$

Assuming only noise in the homography computing and accurate data in the point x, the covariance of the corresponding world point will be:

$$\Lambda_x = B\Lambda_h B^T \tag{F.9}$$

Therefore, these equations applied to our test data give us the results shown in Fig. F.3.

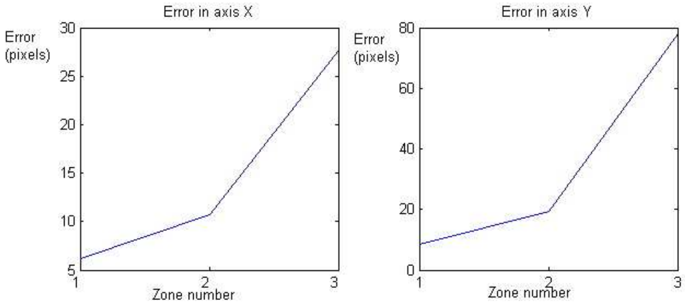


Figure F.3: Perspective error on the plan.

As conclusion, although errors in the image show a valley at the centre of the image, these errors converted to world coordinates increase with the distance from the camera position. The units in world coordinates are pixels, which can be directly converted to metres. This conversion gives values close to those shown in Table 5.5.

List of Tables

List of Figures

List of Algorithms

Bibliography

[Aggarwal, 1987] Aggarwal, J. K. (1987). On the computation of motion from a sequence of monocular or stereo images—an overview. In *Proceedings of the NATO advanced research workshop on Machine intelligence and knowledge engineering for robotic applications*, pages 83–103.

[Aggarwal and Cai, 1999] Aggarwal, J. K. and Cai, Q. (1999). Human motion analysis: a review. *Comput. Vis. Image Underst.*, 73(3):428–440.

[Akita, 1984] Akita, K. (1984). Image sequence analysis of real world human motion. *Pattern Recognition*, 17(1):73–83.

[Althoff et al., 2005] Althoff, K., Degerman, J., and Gustavsson, T. (2005). Combined segmentation and tracking of neural stem-cells. *Lecture Notes in Computer Science*, 3540:282–291.

[Amit S. Chhetri and Papandreou-Suppapppola, 2003] Amit S. Chhetri, D. M. and Papandreou-Suppapppola, A. (2003). Scheduling multiple sensors using particle filters in target tracking. In *IEEE Workshop on Statistical Signal Processing*, pages 549– 552.

[Assfalg et al., 2003] Assfalg, J., Bertini, M., Colombo, C., Bimbo, A. D., and Nunziati, W. (2003). Semantic annotation of soccer videos: Automatic highlights identification. *Computer Vision and Image Understanding*, 92(2-3):285–305.

[Ba and Odobez, 2005] Ba, S. and Odobez, J. (2005). A rao-blackwellized mixed state particle filter for head pose tracking in meetings. In *MMMP*.

[Bar-Shalom and Li, 1993] Bar-Shalom, Y. and Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software.* Artech Houes.

[Baumberg, 1995] Baumberg, A. (1995). *Learning Deformable Models for Tracking Human Motion.* PhD thesis, University of Leeds.

[Baumberg and Hogg, 1994] Baumberg, A. and Hogg, D. (1994). Learning flexible models from image sequences. In *ECCV*, volume 1, pages 299–308.

[Bergman and Doucet, 2000] Bergman, N. and Doucet, A. (2000). Markov chain monte carlo data association for target tracking. In *ICASSP '00: Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference*, pages II705–II708, Washington, DC, USA. IEEE Computer Society.

[Bernardo et al., 1996] Bernardo, E. D., Goncalves, L., and Perona, P. (1996). Monocular tracking of the human arm in 3d: Real-time implementation and experiments. In *ICPR '96:*

Proceedings of the International Conference on Pattern Recognition (ICPR '96), volume 3, pages 622–626, Washington, DC, USA. IEEE Computer Society.

[Bertsekas and Castanon, 1989] Bertsekas, D. P. and Castanon, D. A. (1989). The auction algorithm for the transportation problem. *Ann. Oper. Res.*, 20(1-4):67–96.

[B.Han et al., 2005] B.Han, Yang, C., Duraiswami, R., and Davis, L. (2005). Bayesian filtering and integral image for visual tracking. In *WIAMIS*.

[Bharatkumar et al., 1994] Bharatkumar, A. G., Daigle, K. E., Pandy, M. G., Cai, Q., and Aggarwal, J. K. (1994). Lower limb kinematics of human walking with the medial axis transformation. In *Proc. of IEEE Computer SocietyWorkshop on Motion of Non-Rigid and Articulated Objects*, page 70–76.

[Black and Ellis, 2002] Black, J. and Ellis, T. (2002). Multi camera image measurement and correspondence. *Measurement - Journal of the International Measurement Confederation*, 35(1):61–71.

[Blackman and Popoli, 1999] Blackman, S. and Popoli, R. (1999). *Design and analysis of modern tracking systems*. Artech Houes.

[Blake and Isard, 1998] Blake, A. and Isard, M. (1998). *Active Contours: he Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual tracking of Shapes in Motion*. Springer.

[Bohyung Han, 2004] Bohyung Han, Dorin Comaniciu, Y. Z. L. D. (2004). Incremental density approximation and kernel-based bayesian filtering for object tracking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA.

[Bouchrika and Nixon, 2006] Bouchrika, I. and Nixon, M. S. (2006). People detection and recognition using gait for automated visual surveillance. In *IEE International Symposium on Imaging for Crime Detection and Prevention*, pages 576–581, London, UK.

[Boult et al., 1998] Boult, T., Erkin, A., Lewis, P., Michaels, R., Power, C., Qian, C., and Yin, W. (1998). Frame-rate multi-body tracking for surveillance. In *Proc. of the DARPA IUW*.

[Bowden and Sarhadi, 2000] Bowden, R. and Sarhadi, T. M. M. (2000). Non-linear statistical models for the 3d reconstruction of human pose and motion from monocular image sequences. *Image and Vision Computing*, 18(9):729–737.

[Brand, 2002] Brand, M. (2002). Charting a manifold. In *Advances in Neural Information Processing Systems (NIPS)*, pages 961–968.

[Bray et al., 2007] Bray, M., Koller-Meier, E., and Gool, L. V. (2007). Smart particle filtering for high-dimensional tracking. *Comput. Vis. Image Underst.*, 106(1):116–129.

[Brooks and Williams, 2003] Brooks, A. and Williams, S. (2003). Tracking people with networks of heterogeneous sensors.

[C. Stauffer, 1999] C. Stauffer, E. G. (1999). Adaptive background mixture models for real-time tracking. In *CVPR*, volume II, pages 246–252, olorado Springs, USA.

[Cai and Aggarwal, 1996] Cai, Q. and Aggarwal, J. K. (1996). Tracking human motion using multiple cameras. In *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96)*, volume 3, page 68, Washington, DC, USA. IEEE Computer Society.

[Cai and Aggarwal, 1999] Cai, Q. and Aggarwal, J. K. (1999). Tracking human motion in structured environments using a distributed-camera system. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1241–1247.

[Cai et al., 1995] Cai, Q., Mitiche, A., and Aggarwal, J. (1995). Tracking human motion in an indoor environment. In *Proceedings of Internacional Conference On Image Processing*, volume 1, pages 215–218, Washington D.C., USA.

[Casella and Robert, 1996] Casella, G. and Robert, C. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.

[Chen et al., 2001] Chen, Y., Rui, Y., and Huang, T. S. (2001). Jpdaf based hmm or real-time contour tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 1, page 543.

[Chhetri et al., 2004] Chhetri, A. S., Morrell, D., and Papandreou-Suppappola, A. (2004). The use of particle filtering with the unscented trasform to schedule sensors multiple steps ahead. In *IEEE International Conference on coustics, Speech, and Signal Processing*, pages 301–4.

[Cho et al., 2002] Cho, K.-D., Tae, H.-S., and Chien, S.-I. (2002). Improvement of low gray scale linearity using multi-luminance-level subfield method in plasma display panel. *IEEE Transactions on Consumer Electronics*, 48(3):377–381.

[Choi and Seo, 2004a] Choi, K. and Seo, Y. (2004a). Probabilistic tracking of soccer players and ball. *Statistical Methods in Video Processing*, pages 50–60.

[Choi and Seo, 2004b] Choi, K. and Seo, Y. (2004b). Probabilistic tracking of the soccer ball. In *ECCV Workshop SMVP*, pages 50–60.

[Cipolla and Pentland, 1998] Cipolla, R. and Pentland, A. (1998). *Computer Vision for Human-Machine Interaction.* Cambridge University Press.

[Cohen, 2008] Cohen, C. (February 2008). The gesture recognition home page. http://www.cybernet.com/∼ccohen/.

[Collins, 2003] Collins, R. (2003). Mean-shift blob tracking through scale space. In *Computer Vision and Pattern Recognition*, volume 2, pages 234–240.

[Collins and et al, 2000] Collins, R. and et al (2000). A system for video surveillance and monitoring. vsam final report. Technical report, VSAM.

[Comaniciu and Meer, 1999] Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. In *ICCV '99: Proceedings of the International Conference on Computer Vision*, volume 2, page 1197, Washington, DC, USA. IEEE Computer Society.

[Comaniciu et al., 2000] Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00)*, volume 2, pages 142–151, Hilton Head Island, South Carolina.

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

[Cootes and Taylor, 1997] Cootes, T. and Taylor, C. (1997). A mixture model for representing shape variation. In *Proceedings of British Machine Vision Conference (BMVA)*, pages 110–119.

[Criminisi et al., 1999] Criminisi, A., Reid, I., and Zisserman, A. (1999). A plane measuring device. 17(8):625–634.

[Criminisi et al., 2000] Criminisi, A., Reid, I., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2):123–148.

[Crowley and Demazeau, 1993] Crowley, J. L. and Demazeau, Y. (1993). Principles and techniques for sensor data fusion. *Signal Process.*, 32(1-2):5–27.

[Darrell, 2005] Darrell, T. (2005). 6.891 computer vision and applications. lecture 7: Features and geometry.

[de la Escalera Hueso, 2001] de la Escalera Hueso, A. (2001). *Visión por Computador. Fundamentos y Métodos*. Prentice Hall.

[Deutscher et al., 2000] Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, volume 2, pages 126–133, Hilton Head Island, SC, USA. IEEE Computer Society.

[Deutscher and Reid, 2005] Deutscher, J. and Reid, I. D. (2005). Articulated body motion capture by stochastic search. *Int. Jour. of Computer Vision*, 61(2):185–205.

[Dong et al., 2007] Dong, L., Parameswaran, V., Ramesh, V., and Zoghlami, I. (2007). Fast crowd segmentation using shape indexing. In *ICCV07*, pages 1–8.

[Doucet et al., 2000] Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000). Rao-blackwellised particle filtering for dynamic bayesian networks. In *UAI*.

[Doucet et al., 2002] Doucet, A., Vo, B.-N., Andrieu, C., and Davy, M. (2002). Particle filtering for multi-target tracking and sensor management. In *International Conference on Information Fusion*, pages 474–481.

[Dowdall et al., 2007] Dowdall, J., Pavlidis, I. T., and Tsiamyrtzis, P. (2007). Coalitional tracking. *Comput. Vis. Image Underst.*, 106(2-3):205–219.

[Duong, 2001] Duong, T. (2001). Notes of a seminar given by the author (tarn duong) on 24 may 2001. In *Weatherburn Lecture Series*, http://www.maths.uwa.edu.au/∼duongt/seminars/intro2kde/. Department of Mathematics and Statistics, at the University of Western Australia.

[Ekin et al., 2003] Ekin, A., Tekalp, A. M., and Mehrotra, R. (2003). Automatic soccer video analysis and summarization. *IEEE Trans. on Image Processing*, 12(7):796–807.

[Elgammal et al., 2003] Elgammal, A., Duraiswami, R., and Davis, L. S. (2003). Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(11):1499–1504.

[Elgammal et al., 2002] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002). Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, volume 90, pages 1151–1163.

[Elgammal and Lee, 2004] Elgammal, A. and Lee, C.-S. (2004). Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pages 681–688.

[Farmer et al., 2002] Farmer, M., Hsu, R., and Jain, A. (2002). Interacting multiple model (imm) kalman filters for robust high speed human motion tracking. volume II.

[Fei-Fei et al., 2005] Fei-Fei, L., Fergus, R., and Torralba, A. (2005). Recognizing and learning object categories. http://people.csail.mit.edu/torralba/shortcourserloc/. In *ICCV Short Course*.

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *Int. Jour. of Computer Vision*, 61(2):55–79.

[Ferryman et al., 2006] Ferryman, J., Thirde, D., and Li., L. (2006). Overall evaluation of the pets2006 results. In Computational Vision Group, The University of Reading, U., editor, *IEEE Conference on Computer Vision and Pattern Recognition*.

[Fieguth and Terzopoulos, 1997] Fieguth, P. and Terzopoulos, D. (1997). Color-based tracking of heads and other mobile objects at video frame rates. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 21, Washington, DC, USA. IEEE Computer Society.

[Foresti, 1999] Foresti, G. L. (1999). Object recognition and tracking for remote video surveillance. *IEEE Transactions on Vehicular Technology*, pages 1045–1062.

[Foresti et al., 2000] Foresti, G. L., Mahonen, P., and Regazzoni, C. S. (2000). *Multimedia Video-based Surveillance Systems: Requirements, Issues and Solutions*. Kluwer Academic.

[Foresti et al., 2002] Foresti, G. L., Marcenaro, L., and Regazzoni, C. S. (2002). Automatic detection and indexing of video-event shots for surveillance applications. In *IEEE Transactions on Multimedia*, pages 459–471.

[French et al., 2007] French, A., Naeem, A., Dryden, I., and Pridmore, T. (2007). Using social effects to guide tracking in complex scenes. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'07)*, pages 212–217, London.

[Fryer, 1971] Fryer, C. M. (1971). Biomechanics of the lower extremity. *Instruct Course Lect*, 20:124–130.

[Gammeter et al., 2008] Gammeter, S., Jaeggli, T., Ess, A., Schindler, K., Leibe, B., and Gool, L. V. (2008). Articulated multi-body tracking under egomotion. In *Proc. European Conf. on Computer Vision (ECCV 2008)*.

[Gavrila and Davis, 1996] Gavrila, D. M. and Davis, L. S. (1996). 3-d model-based tracking of humans in action: a multi-view approach. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 73, Washington, DC, USA. IEEE Computer Society.

[Girija et al., 2000] Girija, G., Raol, J., Raj, R. A., and Kashyap, S. (2000). Tracking filter and multi-sensor data fusion. In *Sadhana - Academy Proceedings in Engineering Sciences*, pages 159–167.

[Gómez et al., 2008] Gómez, J., Guerrero, J., and Herrero, J. (2008). Visual tracking on the ground. In *Fifth International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 45–42, Funchal, Madeira.

[Gómez et al., 2006] Gómez, J., Herrero, J., Medrano, C., and Orrite, C. (2006). Multisensor system based on unscented kalman filter. In *IASTED International Conference on Visualization, Imaging, and Image Processing*, pages 59–66, Palma de Mallorca, Spain.

[Goodall, 1991] Goodall, C. (1991). Procrustes methods in the statistical análisis of shape. *Journal of the Royal Statistical Society*, 53(2):285–339.

[Grochow et al., 2004] Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. (2004). Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531.

[Gross and Shi, 2001] Gross, R. and Shi, J. (2001). The cmu motion of body (mobo) database. Technical report, Carnegie Mellon University, http://www.hid.ri.cmu.edu.

[Guo et al., 1994] Guo, Y., Xu, G., and Tsuji, S. (1994). Tracking human body motion based on a stick figure model. 5:1–9.

[Haritaoglu et al., 1999] Haritaoglu, I., Harwood, D., and Davis, L. (1999). Hydra: multiple people detection and tracking using silhouettes. In *Proc. IEEE workshop on Visual Surveillance*, pages 6–13.

[Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*, chapter chapter 7.8. Cambridge University press, second edition edition.

[Hayet et al., 2005] Hayet, J., Mathes, T., Czyz, J., Piater, J., Verly, J., and Macq, B. (2005). A modular multi-camera framework for team sports tracking. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance.*

[Heisele et al., 1997] Heisele, B., Kressel, U., and Ritter, W. (1997). Tracking non-rigid, moving objects based on color cluster flow. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 257, Washington, DC, USA. IEEE Computer Society.

[Herrero et al., 2003] Herrero, E., Orrite, C., and Senar, J. (2003). Detected motion classification with a double-background and a neighborhood-based difference. *Pattern Recognition Letters*, pages 2079–2092.

[Herrero-Jaraba, 2005] Herrero-Jaraba, J. E. (2005). *Análisis visual del movimiento humano.* PhD thesis, University of Zaragoza.

[Hogg, 1982] Hogg, D. (1982). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20.

[Hou et al., 2007] Hou, S., Galata, A., Caillette, F., Thacker, N., and Bromiley, P. (2007). Real-time body tracking using a gaussian process latent variable model. In *ICCV07*, pages 1–8.

[Howe, 2007a] Howe, N. R. (2007a). Evaluating lookup-based monocular human pose tracking on the humaneva test data. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2).*

[Howe, 2007b] Howe, N. R. (2007b). Recognition-based motion capture and the humaneva ii test data. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2).*

[Howell and McKenna, 2008]
Howell, J. and McKenna, S. (February 2008). The visually-mediated interaction home page. http://www.informatics.susx.ac.uk/research/groups/vision/iscanit/vmi2.html.

[Huber, 1996] Huber, E. (1996). 3-d real-time gesture recognition using proximity spaces. In *WACV '96: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, page 136, Washington, DC, USA. IEEE Computer Society.

[Husz et al., 2007] Husz, Z. L., Wallace, A. M., and Green, P. R. (2007). Evaluation of a hierarchical partitioned particle filter with action primitives. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2).*

[Hwang et al., 2003] Hwang, I., Balakrishnan, H., Roy, K., Shin, J., Guibas, L., and Tomlin, C. (2003). Multiple-target tracking and identity management algorithm for air traffic control. In *Proceedings of the Second IEEE International Conference on Sensors*, Toronto, Canada. IEEE Computer Society.

[Intille et al., 1997] Intille, S. S., Davis, J. W., and Bobick, A. F. (1997). Real-time closed-world tracking. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, page 697, Washington, DC, USA. IEEE Computer Society.

[Isard and Blake, 1996] Isard, M. and Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision*, volume 1.

[Isard and Blake, 1998a] Isard, M. and Blake, A. (1998a). Condensation—conditional density propagation forvisual tracking. *Int. J. Comput. Vision*, 29(1):5–28.

[Isard and Blake, 1998b] Isard, M. and Blake, A. (1998b). Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, volume 1, pages 893–908, London, UK. Springer-Verlag.

[Isard and MacCormick, 2001] Isard, M. and MacCormick, J. (2001). Bramble: A bayesian multiple-blob tracker. In *Eighth International Conference on Computer Vision (ICCV'01)*, volume 2, page 34.

[Iwasawa et al., 1997] Iwasawa, S., Ebihara, K., Ohya, J., and Morishima, S. (1997). Real-time estimation of human body posture from monocular thermal images. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 15, Washington, DC, USA. IEEE Computer Society.

[Jaeggli et al., 2006] Jaeggli, T., Koller-Meier, E., and Gool, L. V. (2006). Monocular tracking with a mixture of view-dependent learned models. In *AMDO*.

[Jain, 1997] Jain, E. A. H. P. H. K. R. C. (1997). Estimation of articulated motion using kinematically constrained mixture densities. In *NAM '97: Proceedings of the 1997 IEEE Workshop on Motion of Non-Rigid and Articulated Objects (NAM '97)*, page 10, Washington, DC, USA. IEEE Computer Society.

[Jang and CHoi, 2000] Jang, D. and CHoi, H. (2000). Active models for tracking people with a single video camera. *Pattern Recognition*, 33(7):1135–1146.

[Johansson, 1975] Johansson, G. (1975). Visual motion perception. *Scientific American*, 232:76–88.

[Ju et al., 1996] Ju, S. X., Black, M. J., and Yacoob, Y. (1996). Cardboard people: A parameterized model of articulated image motion. In *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 38, Washington, DC, USA. IEEE Computer Society.

[Julier and U., 1997] Julier, S. J. and U., J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. On Aerospace/Defence Sensing Simulation and Controls.*

[Kakadiaris and Metaxas, 1996] Kakadiaris, I. A. and Metaxas, D. (1996). Model-based estimation of 3d human motion with occlusion based on active multiviewpoint selection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pages 81–87, Los Alamitos, California, U.S.A. IEEE Computer Society.

[Karaulova et al., 2000] Karaulova, I., Hall, P., and Marshall, A. (2000). A hierarchical model of dynamics for tracking people with a single video camera. In *British Machine Vision Conference*, pages 352–361.

[Karlsson and Gustafsson, 2001] Karlsson, R. and Gustafsson, F. (2001). Monte carlo data association for multiple target tracking. In *IEE Target tracking: Algorithms and applications*, The Netherlands.

[Khan and Shah, 2006] Khan, S. M. and Shah, M. (2006). A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV (4)*, pages 133–146.

[Khan et al., 2004] Khan, Z., Balch, T., and Dellaert, F. (2004). Rao-blackwellized particle filter for eigentracking. In *CVPR '04*, volume 2, pages 980–986.

[Khan et al., 2005] Khan, Z., Balch, T., and Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819.

[Khan et al., 2006] Khan, Z., Balch, T., and Dellaert, F. (2006). Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(28):1960–1973.

[Kim and Ramakrishna, 2005] Kim, M. and Ramakrishna, R. S. (2005). New indices for cluster validity assessment. *Pattern Recogn. Lett.*, 26(15):2353–2363.

[Kitagawa, 1996] Kitagawa, G. (1996). Monte carlo filter and smoother for nongaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 2(1).

[Klette et al., 1998] Klette, R., Schlüns, K., and Koschan, A. (1998). *Computer Vision: Three-dimensional data from images.* Springer.

[Kogut and Trivedi, 2002] Kogut, G. and Trivedi, M. (2002). A wide area tracking system for vision sensor networks.

[Kohler, 2008] Kohler, M. (February 2008). Vision based hand gesture recognition systems. http://ls7-www.cs.uni-dortmund.de/research/gesture/.

[Koschan et al., 2003] Koschan, A., Kang, S., Paik, J., Abidi, B., and Abidi, M. (2003). Color active shape model for tracking non-rigid objects. *Image Vision Comput*, 24:1751–1765.

[Kuo et al., 2008] Kuo, P., Makris, D., Megherbi, N., and Nebel, J.-C. (2008). Integration of local image cues for probabilistic 2d pose recovery. In *4th International Symposium on Visual Computing (ISVC2008)*, Las Vegas, USA.

[Lanz, 2006] Lanz, O. (2006). Approximate bayesian multibody tracking. volume 28, pages 1436–1449.

[Lawrence, 2005] Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816.

[Lee and Elgammal, 2006] Lee, C.-S. and Elgammal, A. (2006). Body pose tracking from uncalibrated camera using supervised manifold learning. In *NIPS- Workshop on Evaluation of Articulated Human Motion and Pose Estimation. EHuM06.*

[Lee and Chen, 1984] Lee, H. and Chen, Z. (1984). Optimal search procedures for 3d human movement determination. In *CAIA84*, pages 389–394.

[Lee and Cohen, 2003] Lee, M. W. and Cohen, I. (2003). Human body tracking with auxiliary measurements. In *IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, pages 112–119.

[Leung and Yang, 1995] Leung, M. K. and Yang, Y.-H. (1995). First sight: A human body outline labeling system. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(4):359–377.

[Li and Chua, 2007] Li, J. and Chua, C. (2007). Transductive local exploration particle filter for object tracking. *Image Vision Computing*, 25(5):544–552.

[Li et al., 2006] Li, R., Yang, M., Sclaroff, S., and Tian, T. (2006). Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *ECCV06*, volume 2.

[Li and Zheng, 2004] Li, X. and Zheng, N. (2004). Adaptive target color model updating for visual tracking using particle filter. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 4, pages 3105–3109.

[Lindeberg, 1998] Lindeberg, T. (1998). Principles for automatic scale selection. Technical Report ISRN KTH NA/P–98/14–SE, Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), Stockholm, Sweden.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110.

[MacCormick and Blake, 2000] MacCormick, J. and Blake, A. (2000). A probabilistic exclusion principle for tracking multiple objects. *Int. J. Comput. Vision*, 39(1):57–71.

[Maggio and Cavallaro, 2005] Maggio, E. and Cavallaro, A. (2005). Hybrid particle filter and mean shift tracker with adaptive transition model. In *ICASSP*, pages 21–224.

[Martínez et al., 2006] Martínez, J., Herrero, J., Gómez, J., and Orrite, C. (2006). Automatic left luggage detection and tracking using multi-camera ukf. In *IEEE international Workshop on Performance Evaluation of Tracking and Surveillance (PETS 06)*, pages 59–66, New York, NY.

[Matsumoto et al., 2000] Matsumoto, K., Sudo, S., Saito, H., and Ozawa, S. (2000). Optimized camera viewpoint determination system for game soccer broadcasting. In *Proc. MVA2000, IAPR Workshop on Machine Vision Applications*, pages 115–118, Tokyo.

[Maurin et al., 2002] Maurin, B., Masoud, O., and Papanikolopoulos, N. (2002). Monitoring crowded traffic scenes. In *The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 19–24.

[Maybeck, 1982] Maybeck, P. (1982). *Stochastic models, estimation and control*, volume 2. Academic Press, NewYork.

[Mcallister et al., 2002] Mcallister, G., Mckenna, S. J., and Ricketts, I. W. (2002). Mlesac-based tracking with 2d revolute-prismatic articulated models. In *ICPR '02*.

[McKenna et al., 1999] McKenna, S., Raja, Y., and Gong, S. (1999). Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17:225–231.

[McKenna et al., 2000] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding: CVIU*, 80(1):42–56.

[McKenna et al., 1997] McKenna, S. J., Raja, Y., and Gong, S. (1997). Object tracking using adaptive color mixture models. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume I*, pages 615–622, London, UK. Springer-Verlag.

[Medrano et al., 2008] Medrano, C., Herrero, J., Martínez, J., and Orrite, C. (2008). Mean field approach for tracking similar objects.

[Meyer et al., 1998] Meyer, M., Ohmacht, T., Bosch, R., and Hotter, M. (1998). Video surveillance applications using multiple views of a scene. In *32nd Annual 1998 International Carnahan Conference on Security Technology Proceedings*, pages 216–219, Alexandria (VA, USA).

[Michalowski and Simmons, 2006] Michalowski, M. P. and Simmons, R. (2006). Multimodal person tracking and attention classification. In *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 349–350.

[Mikolajczyk and Schmid, 2001] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *roceedings. Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 525–531, Vancouver, BC, Canada.

[Mikolajczyk and Schmid, 2002] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 128–142, Copenhagen. Springer.

[Moeslund et al., 2006] Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(1):90–126.

[Montesinos et al., 1998] Montesinos, P., Gouet, V., and Deriche, R. (1998). Differential invariants for color images. In *Proceedings of 14 th International Conference on Pattern Recognition*.

[Nait-Charif and McKenna, 2003] Nait-Charif, H. and McKenna, S. J. (2003). Head tracking and action recognition in a smart meeting room. In *Proceedings PETS*.

[Needham and Boyle, 2001] Needham, C. J. and Boyle, R. D. (2001). Tracking multiple sports players through occlusion, congestion and scale. In *British Machine Vision Conference BMCV'01*, pages 93–102, Manchester, UK.

[Niyogi and Adelson, 1993] Niyogi, Sourabh, A. and Adelson, Edward, H. (1993). Analyzing and recognizing walking figures in XYT. Technical Report 223.

[Noriega and Bernier, 2007] Noriega, P. and Bernier, O. (2007). Multicues 2d articulated pose tracking using particle filtering and belief propagation on factor graphs. In *ICIP07*, volume 5.

[Nummiaro et al., 2002] Nummiaro, K., Koller-Meier, E., and Gool, L. V. (2002). A color-based particle filter. In *Symposium for pattern recognition of the DAGM*, volume 2449, pages 353–360.

[Nummiaro et al., 2003] Nummiaro, K., Koller-Meier, E. B., and Gool, L. V. (2003). An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110.

[Okawa and Hanatani, 1992] Okawa, Y. and Hanatani, S. (1992). Recognition of human body motions by robots. In *Proceedings of Internacional Conference On Intelligent Robots and Systems*, pages 2139–2146, Raleigh, NC.

[Okuma et al., 2003] Okuma, K., Little, J. J., and Lowe, D. (2003). Automatic acquisition of motion trajectories: Tracking hockey players. In *Proceedings of SPIE*, volume 5304.

[Okuma et al., 2004] Okuma, K., Taleghani, A., de Freitas, N., Little, J., and Lowe, D. (2004). A boosted particle filter: Multitarget detection and tracking. In *Proc. ECCV*, volume 3021 of LNCS, pages 28–39.

[Ormoneit et al., 2005] Ormoneit, D., Black, M. J., Hastie, T., and Kjellström, H. (2005). Representing cyclic human motion using functional analysis. *Image and Vision Computing*, 23(14):1264–1276.

[O'Rourke and Badler, 1980] O'Rourke, J. and Badler, N. (1980). Model-based image analysis of human motion using constraint propagation. *PAMI*, 2(6):522–536.

[Paragios and Deriche, 2000] Paragios, N. and Deriche, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(3):266–280.

[Peng et al., 2005] Peng, N., Yang, J., and Liu, Z. (2005). Mean shift blob tracking with kernel histogram filtering and hypothesis testing. *Pattern Recognition Letters*, 26:605–614.

[Perales and Torres, 1994] Perales, F. J. and Torres, J. (1994). A system for human motion matching between synthetic and real image based on a biomechanic graphical model. In *Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA.

[Pérez et al., 2002] Pérez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK. Springer-Verlag.

[Perez et al., 2004] Perez, P., Vermaak, J., and Blake, A. (2004). Data fusion for visual tracking with particles. In *Proceedings of IEEE (issue on State Estimation)*.

[Peterfreund, 1999] Peterfreund, N. (1999). Robust tracking of position and velocity with kalman snakes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(6):564–569.

[Pitt and Shephard, 1999] Pitt, M. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filter. *Journal of the American Statistical Association*, 94.

[Polana and Nelson, 1994] Polana, R. and Nelson, R. (1994). Low level recognition of human motion. In *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82.

[Polat et al., 2003] Polat, E., Yeasin, M., and Sharma, R. (2003). Robust tracking of human body parts for collaborative human computer interaction. *Comput. Vis. Image Underst.*, 89(1):44–69.

[Poppe, 2007] Poppe, R. (2007). Evaluating example-based pose estimation: Experiments on the humaneva sets. In *Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2)*.

[Porikli, 2005] Porikli, F. (2005). Integral histograms: A fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition*, volume 1, pages 829–836.

[Punska, 1999] Punska, O. (1999). Bayesian approaches to multi-sensor data fusion. Master's thesis, Cambridge University Engineering Department.

[Qu et al., 2005] Qu, W., Schonfeld, D., and Mohamed, M. (2005). Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model. 2:535–540.

[Rago et al., 1995] Rago, C., P.Willett, and R.Streit (1995). A comparison of the jpdaf and pmht tracking algorithms. In *IEEE Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 3571–3574.

[Rahimi et al., 2005] Rahimi, A., Recht, B., and Darrell, T. (2005). Learning appearance manifolds from video. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1.

[Ramanan et al., 2007] Ramanan, D., Forsyth, D., and Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81.

[Rashid, 1980] Rashid, R. F. (1980). Towards a system for the interpretation of moving light display. *IEEE Trans. PAMI*, 2(6):574–581.

[Regazzoni et al., 1999] Regazzoni, C., Fabri, G., and Vernazza, G. (1999). *Advanced Video-Based Surveillance Systems*. Kluwer Academic.

[Rehg et al., 2003] Rehg, J., Morris, D., and Kanade, T. (2003). Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. *The International Journal of Robotics Research*, 22(6):393–418.

[Rehg and Kanade, 1995] Rehg, J. M. and Kanade, T. (1995). Model-based tracking of self-occluding articulated objects. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pages 612–617, Washington, DC, USA. IEEE Computer Society.

[Ren et al., 2004] Ren, J., Orwell, J., Jones, G., and Xu, M. (2004). Real-time 3d soccer ball tracking from multiple cameras. In *British Machine Vision Conference*, pages 829–838.

[Ripley, 1987] Ripley, B. (1987). *Stochastic Simulation*. Wiley, New York.

[Rogez et al., 2006] Rogez, G., Orrite, C., Martínez, J., and Herrero, J. (2006). Probabilistic spatio-temporal 2d-model for pedestrian motion analysis in monocular sequences. In *AMDO*.

[Rohr, 1994] Rohr, K. (1994). Towards model-based recognition of human movements in image sequences. *CVGIP: Image Underst.*, 59(1):94–115.

[Rosin and Ellis, 1995] Rosin, P. L. and Ellis, T. (1995). Image difference threshold strategies and shadow detection. In *BMVC '95: Proceedings of the 1995 British conference on Machine vision*, volume 1.

[Rossi and Bozzoli, 1994] Rossi, M. and Bozzoli, A. (1994). Tracking and counting people. In *1st International Conference on Image Processing*, pages 212–216, Austin, Texas, USA.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

[Roweiss, 1999] Roweiss, S. (1999). Gaussian identities. Technical report, University of Toronto, http://www.cs.toronto.edu/∼roweis/notes/gaussid.pdf.

[Rui and Chen, 2001] Rui, Y. and Chen, Y. (2001). Better proposal distributions: Object tracking using unscented kalman filter. In *CVPR '01*, pages 786–793.

[S. Wu, 2004] S. Wu, T. C. (2004). Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern Recognition*, 37:175–188.

[Sacchi and Regazzoni, 2000] Sacchi, C. and Regazzoni, C. S. (2000). A distributed surveillance system for detection of abandoned objects in unmanned railway environments. *IEEE Transactions on Vehicular Technology*, pages 2013–2026.

[Safonova et al., 2004] Safonova, A., Hodgins, J. K., and Pollard, N. S. (2004). Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.*, 23(3):514–521.

[Sato et al., 1998] Sato, K., Maeda, T., Kato, H., and Inokuchi, S. (1998). Cad-based object tracking with distributed monocular camera forsecurity monitoring. In *Proceedings of the Second CAD-Based Vision Workshop*, pages 291–297, Champion, PA, USA.

[Segen and Pingali, 1996] Segen, J. and Pingali, S. G. (1996). A camera-based system for tracking people in real time. In *ICPR '96: Proceedings of the International Conference on Pattern Recognition (ICPR '96)*, volume 3, page 63, Washington, DC, USA. IEEE Computer Society.

[Senior, 2002] Senior, A. (2002). Tracking people with probabilistic appearance models. pages 48–55.

[Sidenbladh et al., 2000] Sidenbladh, H., Black, M. J., and Fleet, D. J. (2000). Stochastic tracking of 3d human figures using 2d image motion. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 702–718, London, UK. Springer-Verlag.

[Siebel and Maybank, 2004] Siebel, N. and Maybank, S. (2004). The advisor visual surveillance system. In *ECCV Workshop on Applications of ComputerVision*, pages 103–1111.

[Siebel, 2003] Siebel, N. T. (2003). *Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications.* PhD thesis, University of Reading, Computational VIsion Group, Deparment of Computer Science.

[Sigal et al., 2007] Sigal, L., Balan, A., and Black, M. J. (2007). Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems, NIPS-2007*.

[Sigal and Black, 2006] Sigal, L. and Black, M. J. (2006). Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University.

[Sigal et al., 2000] Sigal, L., Sclaroff, S., and Athitsos, V. (2000). Estimation and prediction of evolving color distributions for skinsegmentation under varying illumination. In *Computer Vision and Pattern Recognition*, volume 2, pages 152–159.

[Silverman, 1986] Silverman, B. (1986). *Density Estimation.* Chapman and Hall, London.

[Sitbon and Passerieux, 1995] Sitbon, S. and Passerieux, J. (1995). New efficient target tracking based upon hidden markov model and probabilistic data association. In *29th Asilomar Conference on Signals, Systems and Computers*, volume 2, page 849.

[Sminchisescu and Jepson, 2004] Sminchisescu, C. and Jepson, A. (2004). Generative modeling for continuous non-linearly embedded visual inference. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 96, New York, NY, USA. ACM.

[Sminchisescu and Triggs, 2001] Sminchisescu, C. and Triggs, B. (2001). Covariance scaled sampling for monocular 3d body tracking. In *CVPR01*, volume 1, pages 447–454.

[Spengler and Schiele, 2003] Spengler, M. and Schiele, B. (2003). Automatic detection and tracking of abandoned objects. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France.

[Stauffer and Grimson, 2000] Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757.

[Stenger et al., 2006] Stenger, B., Thayananthan, A., Torr, P. H., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384.

[Stringa and Regazzoni, 2000] Stringa, E. and Regazzoni, C. S. (2000). Real-time video-shot detection for scene surveillance applications. *IEEE Transactions on Image Processing*, pages 69–79.

[Sudderth et al., 2004] Sudderth, E., Mandel, M., Freeman, W., and Willsky, A. (2004). Visual hand tracking using nonparametric belief propagation. In *Computer Vision and Pattern Recognition Workshop*, volume 12, page 189.

[Sullivan and Carlsson, 2006] Sullivan, J. and Carlsson, S. (2006). Tracking and labelling of interacting multiple targets. In *Proc. 9th European Conf. on Computer Vision (ECCV 2006)*.

[Teh and Roweis, 2003] Teh, Y. and Roweis, S. (2003). Automatic alignment of local representations. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, USA. MIT Press.

[Tenenbaum, 1998] Tenenbaum, J. B. (1998). Mapping a manifold of perceptual observations. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 682–688, Cambridge, MA, USA. MIT Press.

[Thirde et al., 2006] Thirde, D., Borg, M., Ferryman, J., J.Aguilera, Kampel, M., and Fernandez, G. (2006). Multi-camera tracking for visual surveillance applications. In *11th Computer Vision Winter Workshop*, Czech Republic.

[Tian et al., 2005] Tian, T.-P., Li, R., and Sclaroff, S. (2005). Articulated pose estimation in a learned smooth space of feasible solutions. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 50, Washington, DC, USA. IEEE Computer Society.

[Torma and Szepesvari, 2004] Torma, P. and Szepesvari, C. (2004). Enhancing particle filters using local likelihood sampling. In *Proc. ECCV*, volume 3021 of LNCS, pages 16–27.

[Tsai, 1987] Tsai, R. (1987). Metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344.

[Urtasun et al., 2005a] Urtasun, R., Fleet, D. J., and Fua, P. (2005a). Monocular 3-d tracking of the golf swing. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 932–938, Washington, DC, USA. IEEE Computer Society.

[Urtasun et al., 2006] Urtasun, R., Fleet, D. J., and Fua, P. (2006). 3d people tracking with gaussian process dynamical models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, Washington, DC, USA. IEEE Computer Society.

[Urtasun et al., 2005b] Urtasun, R., Fleet, D. J., Hertzmann, A., and Fua, P. (2005b). Priors for people tracking from small training sets. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 403–410, Washington, DC, USA. IEEE Computer Society.

[Utsumi et al., 1998] Utsumi, A., Mori, H., Ohya, J., and Yachida, M. (1998). Multiple-view-based tracking of multiple humans. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, page 597, Washington, DC, USA. IEEE Computer Society.

[Varshney, 2008] Varshney, P. K. (2008). Multisensor data fusion and applications. In *IEEE SIU*.

[Vermaak et al., 2003a] Vermaak, J., Doucet, A., and Perez, P. (2003a). Maintaining multimodality through mixture tracking. In *International Conference on Computer Vision*, volume 2, page 1110, Nice.

[Vermaak et al., 2003b] Vermaak, J., Lawrence, N. D., and Pérez, P. (2003b). Variational inference for visual tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:773.

[Vermaak et al., 2002] Vermaak, J., Pérez, P., Gangnet, M., and Blake, A. (2002). Towards improved observation models for visual tracking: selective adaptation. In *Proc. Europ. Conf. Computer Vision*, Copenhagen, Denmark.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Wachter and Nagel, 1999] Wachter, S. and Nagel, H.-H. (1999). Tracking persons in monocular image sequences. *Comput. Vis. Image Underst.*, 74(3):174–192.

[Wagner, 2007] Wagner, M. (2007). *Tracking with Multiple Sensors.* PhD thesis.

[Wallner et al., 1995] Wallner, F., Graf, R., and Dillmann, R. (1995). Real-time map refinement by fusing sonar and active stereo-vision. volume 3, pages 2968–2973.

[Wan and v. d. M., 2001] Wan, E. A. and v. d. M., R. (2001). *The Unscented Kalman Filter*, chapter 7.

[Wang et al., 2003] Wang, Q., Xu, G., and Ai, H. (2003). Learning object intrinsic structure for robust visual tracking. In *Proc. CVPR*, volume 2, page 227.

[Welch and Bishop, 1995] Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical report, University of North Carolina, Chapel Hill, NC, USA.

[Wren et al., 1997] Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785.

[Wu et al., 2003] Wu, Y., Hua, G., and Yu, T. (2003). Tracking articulated body by dynamic markov network. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1094, Washington, DC, USA. IEEE Computer Society.

[Xu et al., 2005] Xu, M., Lowey, L., Orwell, J., and Thirde, D. (2005). Architecture and algorithms for tracking football players with multiple cameras. *IEEE Proceedings-Vision, Image and Signal Processing.*

[Xu et al., 2004] Xu, M., Orwell, J., and Jones, G. A. (2004). Tracking football players with multiple cameras. In *International Conference on Image Processing*, volume 5, pages 2909–2912.

[Yamada et al., 2002] Yamada, A., Shirai, Y., and Miura, J. (2002). Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02)*, volume 1, page 10303.

[Yang and Wu, 2006] Yang, M.-S. and Wu, K.-L. (2006). Unsupervised possibilistic clustering. *Pattern Recogn.*, 39(1):5–21.

[Yu et al., 2006] Yu, Q., Cohen, I., Medioni, G., and Wu, B. (2006). Boosted markov chain monte carlo data association for multiple target detection and tracking. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 675–678, Washington, DC, USA. IEEE Computer Society.

[Yu and Wu, 2004] Yu, T. and Wu, Y. (2004). Collaborative tracking of multiple targets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, DC, USA. IEEE Computer Society.

[Zhang et al., 1996] Zhang, Y., Leung, H., Lo, T., and J.Litva (1996). Distributed sequential nearest neighbour multitarget tracking algorithm. In *IEE Proc.-Radar, Sonar Navig.*, volume 143, pages 255–260.

Index